Universiteit Leiden
The Netherlands

Data Science and Artificial Intelligence

Evaluating Adversarial Robustness in Time-Series Classification: A Comparative Study on Self-Supervised Learning Models

Jesse Kroll s3666778

Supervisors:
Wadie Skaf, M.Sc.[1] & Dr. Mitra Baratchi

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

01/07/2025

[1] Chair for Artificial Intelligence Methodology (AIM), RWTH Aachen University

Contents

Nomenclature **1**

1 Introduction **1**

2 Related Work **4**

3 Methods **7**

3.1 Adversarial Attacks 7

3.1.1 Individual White-Box Attack 9

3.1.2 Universal Black-Box Attack 12

3.1.3 Individual Gray-box attack 14

3.2 Self-Supervised Learning models 14

3.3 Evaluation Metrics 15

4 Experiment **17**

5 Results **19**

6 Conclusion **28**

6.1 Limitations and Further Research 28

7 Bibliography **32**

8 Appendix **33**

Nomenclature

$\arg_{\max}$ Operator that returns the index of the maximum value

ΔA Absolute accuracy drop: $\Delta A = A_{\text{clean}} - A_{\text{adv}}$

$\Delta_R A$ Relative accuracy drop: $\Delta_R A = \frac{A_{\text{clean}} - A_{\text{adv}}}{A_{\text{clean}}}$

$\delta_{\mathbf{x}} \in \mathbb{R}^T$ Adversarial perturbation, same length as $\mathbf{x}$

$\mathbf{x}_{\text{adv}}$ Adversarial counterpart of $\mathbf{x}$: $\mathbf{x}_{\text{adv}} = \mathbf{x} + \delta_{\mathbf{x}}$ and $\mathbf{x}_{\text{adv}}, \delta_{\mathbf{x}} \in \mathbb{R}^T$

$\mathbf{x} \in \mathbb{R}^T$ Clean time-series sample of length T

$\mathbf{z}$ Latent representation or feature vector

$\varepsilon \in \mathbb{R}$ Perturbation budget: upper bound on the ℓ_p-norm of any allowed perturbation, i.e., $\|\delta\|_p \leq \varepsilon_{\max}$

A_{adv} Classification accuracy on the adversarial (perturbed) test set

A_{clean} Classification accuracy on the clean test set

$f(\mathbf{x})$ Feature extractor or encoder mapping input $\mathbf{x}$ to $\mathbf{z}$

$f_{\text{surrogate}}$ Surrogate model: the model used to craft adversarial examples in transfer-based attacks

f_{target} Target model: the model under attack in transfer-based adversarial scenarios

$h : \mathbb{R}^d \to [0, K-1]$ Classifier mapping from d-dimensional feature space to class labels

$h_k(\mathbf{z})$ Classifier output for class k given feature vector $\mathbf{z}$

$X \in \mathbb{R}^{[N,T]}$ Dataset with N samples and length T

$y' \in [0, K-1]$ Class label of $\mathbf{x}_{\text{adv}}$, K is the total number of classes

$y \in [0, K-1]$ Ground-truth class label of $\mathbf{x}$, where K is the total number of classes

Abstract

Time-series classification plays a crucial role in healthcare, finance, and industrial control. Adversarial perturbations in time-series classification pose significant risks to model reliability. Despite recent advances in unsupervised representation learning methods for time-series data, their robustness to adversarial attacks remains underexplored. This study evaluates three state-of-the-art self-supervised learning models, namely Series2Vec, "Time-Series to Vector" (TS2Vec), and Time-Series Representation Learning via Temporal and Contextual Contrasting (TS-TCC), across five attack scenarios including gradient-based (Fast Gradient-Sign Method (FGSM), Basic Iterative Method (BIM), Projected Gradient Descent (PGD)), and architecture-agnostic (Differential Evolution (DE) and Transfer Projected Gradient Descent (T-PGD)) attacks. On 128 univariate UCR datasets, results show that TS2Vec is most robust with a degradation of only 8% at most thanks to the hierarchical temporal-level contrasting. In contrast, Series2Vec demonstrated vulnerability especially to universal attacks, while TS-TCC demonstrated moderate robustness. Overall, the results reveal significant differences in the adversarial robustness of unsupervised representation learning models, emphasizing the need for future research into more resilient model architectures.

Chapter 1

Introduction

Time-series data are sequences of samples ordered by time, where each observation corresponds to a specific time point. The rise of large-scale sensing technologies such as satellites has led to time-series becoming the backbone of modern decision making in domains including healthcare, finance, and industrial control [Baratchi et al., 2022]. In such safety-critical environments, a single misclassification can lead to a cascade of costly production downtime, erroneous medical interventions and major losses in financial resources [Ismail Fawaz et al., 2019]. While Deep Neural Networks (DNNs) trained in a fully supervised manner demonstrate promising results for classification [Karim et al., 2018], DNNs can only achieve high accuracies when ample labelled data is available. Annotating large amounts of time series data is expensive and often infeasible in real-world settings, which poses a challenge for large-scale application of DNNs. In recent years, the field of unsupervised representation learning has demonstrated that useful features can be learned without labels; in particular self-supervised learning (SSL) methods in vision and speech domains have demonstrated strong performance [Baevski et al., 2020, Chen et al., 2020]. Generally, unsupervised representation learning realizes a mapping from raw sequences to feature vectors without using any class labels, while SSL is viewed as a specialised subset of utilising pretext tasks that automatically extract meaningful representations from the data itself. Furthermore, unsupervised representation learning models can achieve better results when transferring across different domains and capture data invariances more accurately [Eldele et al., 2021, Yue et al., 2022].

Recently, these ideas migrated to time-series and SSL models like TS2Vec have topped existing benchmarks, while utilising 95% fewer labels than are required for DNNs [Chen et al., 2020]. Thanks to these recent breakthroughs, SSL is the current state-of-the-art approach for time-series classification by outperforming fully supervised DNNs across most datasets and tasks [Foumani et al., 2024]. Additionally, recent studies have revealed that small, human-imperceptible perturbations significantly reduce the accuracy of classical DNNs [Fawaz et al., 2019]. For example, a simple attack called Fast Gradient-Sign Method (FGSM), that perturbs a time-series in the direction of the gradient of the loss with respect to the input, can reduce the classification accuracy from 96% to 4% on electrocardiogram (ECG), presenting severe shortcomings of classical DNNs. These perturbations are called adversarial attacks. In real-world applications, robustness to such adversarial attacks constitutes a prerequisite to trusting time-series classification models in high-stakes, risk-laden environments. While substantial progress in natural-language processing and computer-vision

domains has been achieved, the time-series classification domain lacks research depth on adversarial attacks. Although it has been shown that SSL models achieve high classification accuracies under sterile data conditions, the adversarial robustness in SSL models for time-series classification has not yet been comprehensively studied. Moreover, the evaluation of multiple models on architectural robustnes—using matched hyperparameters on identical adversarial attack protocols—has been partly investigated for DNNs by Li et al. [2024], but not on SSL models. In fact, such a systematic comparison of robustness in unsupervised representation learning models, and especially SSL models, is still absent. Hence, this study addresses this literature gap by examining the alleged inherent robustness of state-of-the-art SSL models [Eldele et al., 2024].

Robustness to adversarial attacks is a critical requirement for deploying time-series SSL models in safety-sensitive domains. Recent advances have significantly improved classification performance, but little is known about how robust these models are when exposed to adversarial perturbations. In particular, no systematic benchmark currently exists that assesses the adversarial robustness of pre-trained encoders. Hence, this thesis provides the first systematic benchmark of adversarial robustness for state-of-the-art SSL models in time-series classification. The thesis evaluates three state-of-the-art SSL models, which represent their respective subclasses of SSL methods within the unsupervised representation learning taxonomy provided by Meng et al. [2023] and as shown in Figure 2.1. The models are representing temporal-level and instance-level contrastive and predictive methods. The models representing the subclasses are TS2Vec, Series2Vec, and TS-TCC. The 128 UCR datasets serve as a comparable benchmark for evaluating the robustness of the models based on the classification accuracy under adversarial attacks. A unified attack-generation pipeline across models ensures a fair and consistent basis for comparison. A novel architecture-agnostic attack type named Differential Evolution (DE) is introduced that addresses the gap in the literature for full black-box adversarial attacks. DE optimizes adversarial perturbations beyond the attack-vector of gradient loss offering a complementary and less biased perspective on model robustness. DE, transfer attacks and three classical gradient-based attack types are utilised for broad evaluation. Accordingly, the study examines which intrinsic properties of SSL model families result in greater adversarial robustness. The following research questions motivate the study:

- **How robust are state-of-the-art unsupervised representation learning models for time-series classification against adversarial attacks, as measured by their impact on classification accuracy?**
- **Which SSL model architectures are especially prone to certain types of adversarial attacks, and what characteristics explain their robustness or vulnerability?**
- **How do the embedding spaces of Series2Vec, TS2Vec, and TS-TCC influence the transferability and robustness of adversarial examples across models?**

The thesis delivers the following contributions:

- **Adversarial Robustness Benchmarking Pipeline and Novel Attack Space Taxonomy** — Developed a systematic, unbiased, and reproducible experimental pipeline publicly available

for evaluating the robustness of SSL models (Series2Vec, TS2Vec, TS-TCC) across diverse adversarial attack scenarios (FGSM, BIM, PGD, DE and T-PGD) on the UCR datasets.

- **Novel Attack Space Taxonomy** — Proposed a structured, three-dimensional taxonomy (with the axis: adversarial knowledge, perturbation scope, attack goal) to classify and analyse adversarial attacks enabling consistent classification for novel gray-box, semi-targeted, and group-level attacks.
- **Novel Differential Evolution Attack** — Designed and implemented a universal black-box adversarial attack based on DE tailored explicitly for time-series classification revealing vulnerabilities overlooked by gradient-based methods and broadening the scope of black-box attacks available in literature.
- **Comparative Robustness Analysis of SSL Models** — Provided comparative insights into embedding transferability and robustness among SSL architectures, showcasing the inherent robustness and susceptibility of TS2Vec and Series2Vec respectively. Indicating superiority of temporal-level contrastive over instance-level methods, while all models transfer embedding vulnerabilities symmetrically

Giving an overview on the structure of the thesis, this chapter introduces the topic, motivation and objective; Chapter 2 discusses related work, which forms the foundation of the thesis; Chapter 3 details the experimental design, including datasets, attack types, attack parameters, and models. Chapter 4 describes the experimental setup; Chapter 5 presents empirical results and visualizations, while discussing bottlenecks of the experimental setup; Chapter 6 summarizes the findings and outlines directions for future research.

Chapter 2

Related Work

The idea of adversarial examples on time-series was formulated by Goodfellow et al. [2015] first, who showed that models using gradient-based optimization can be vulnerable to small, carefully crafted perturbations in the input data that are imperceptible to humans but cause the model to misclassify with high confidence. This phenomenon, known as adversarial vulnerability, exposed a fundamental weakness in DNNs and spurred a growing field of research into both generating and defending against such adversarial examples. Moreover, the first gradient-based attack called FGSM was proposed by Goodfellow et al. [2015], which leverages knowledge of model weights. The FGSM generates adversarial examples by adding a small perturbation to the input in the direction of the gradient of the loss, effectively maximizing the model's prediction error. While FGSM lacks attack strength, it is still utilised as an effective baseline with low computational cost.

Fawaz et al. [2019] built on Goodfellow et al. [2015] findings and adapted them to the time-series classification domain and showed that a single–step FGSM can significantly reduce accuracy on UCR datasets. Additionally, they showed that adversarial attacks transfer and hence can generalize across models. Siddiqui et al. [2020] extended FGSM to PGD and the BIM, but the study was restricted to three classical architectures and was conducted exclusively under white-box settings. It laid the foundation for showing that BIM and PGD are strong attacking methods that are relevant, while being more computationally expensive due to the iterative design. Two surveys have been essential for drawing a comprehensive picture of the SSL landscape. Eldele et al. [2024] developed a novel taxonomy for label-efficient time series representation learning and Zhang et al. [2024] explicitly highlighted self-supervised learning for time series adversarial robustness as an open frontier that currently lacks research depth. Lastly, Zhang et al. [2024] remarked that adversarial attacks during the pre-training phase have not yet been explored. Eldele et al. specifically highlighted augmentation and contrastive learning as powerhouses for robustness. Hence, these two papers provide direct justification to investigate the properties of SSL models with respect to adversarial robustness.
While Zhang et al. [2024] and Eldele et al. [2024] provided a fine-grained taxonomy, Meng et al. [2023] gave a broader overview on the SSL landscape as shown in Figure 2.1. This taxonomy serves as a cornerstone for exploring differences in robustness across current SSL architectures and allows for comparing such SSL categories in a structured manner. There are three major SSL methods: contrastive, adversarial, and predictive as described by Meng et al. [2023] as in the following . While this thesis investigates contrastive methods at the instance- and temporal-level, and predictive methods, all of the SSL methods are shown in Figure 2.1 for completeness. All SSL models differ in

their pretext task that guides each model towards creating meaningful representations of time-series. When applying contrastive methods, such representations get crafted through embeddings by self-discrimination. Typically, samples are augmented to discern similarities and discover contextualized underlying factors of variation for feature extraction. Such augmented views are contrasted to match samples as positive or negative pairs with respect to similarity. Contrasting can be performed on three different levels: prototype-, temporal-, and instance-level. Prototype-level methods target the implicit semantics shared by samples within the cluster. Capturing scale-invariant representations at each individual timestamp, known as temporal-level contrasting, aims to understand time-dependency. Instance-level methods contrast augmentations of samples individually treating different samples as negative pairs. Adversarial methods generate synthetic samples and train a generator and a discriminator in a min-max player fashion to improve the model's ability to distinguish between samples. Predictive methods, as the name suggests, predict masked slices to gain future missing or contextual information.

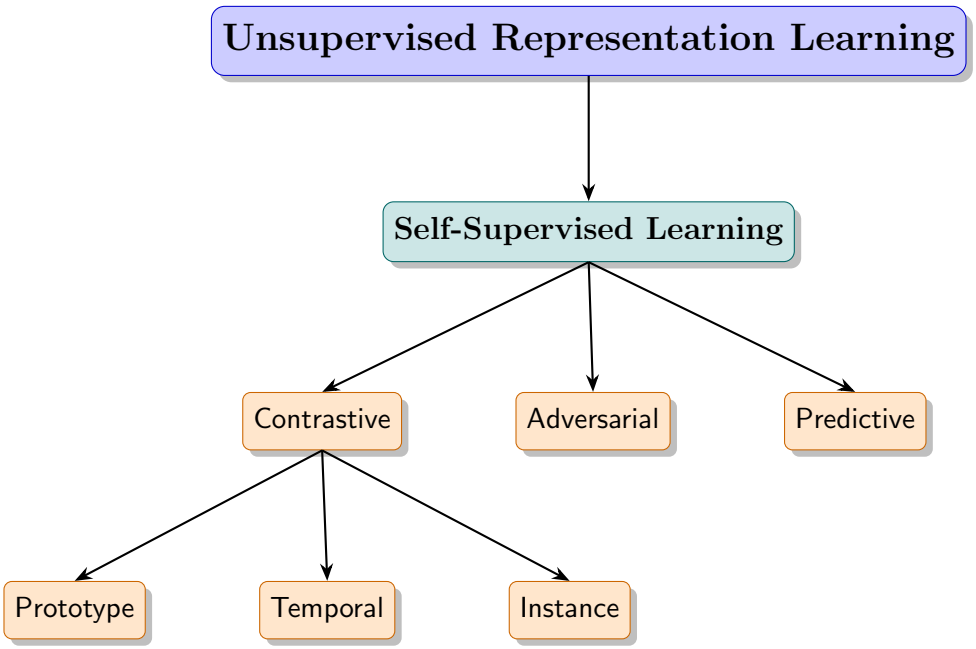


Figure 2.1: Hierarchical decomposition of unsupervised representation learning into SSL and its methods as proposed by Meng et al. [2023] for time-series models.

Talking about existing limitations of existing benchmarks, while Trirat et al. [2023] and Zhang et al. [2024] provided useful overviews on time-series taxonomies and the SSL model landscape, direct comparisons of models on the same datasets are not summarized in one paper, but mostly scattered across the papers proposing novel models [Eldele et al., 2021, Foumani et al., 2024, Guen and Thome, 2023, Malhotra et al., 2016, Yue et al., 2022]. Li et al. [2024] included an adversarial robustness comparison on classification including TS2Vec and four DNNs until here and still overlooks cross-comparing adversarial robustness in self-supervised learners. Current adversarial robustness evaluation protocols vary in perturbation budgets, defined as ε with respect to the ℓ_p norm used. Generally, ℓ_p-norm is defined for a sample time-series x as $\|x\|_p = \left(\sum_{t=1}^{T} |x_t|^p\right)^{1/p}$ where $(1 \leq p < \infty)$. Based on that definition, the perturbation vector δ_x is bounded by $\|\delta_x\|_\infty =$

$\max_{t=1,\ldots,T} \|\delta_{x,t}\| \leq \varepsilon$. For example, studies use $\varepsilon = 0.1$ [Fawaz et al., 2019] vs. $0.05 \leq \varepsilon \leq 0.3$ [Siddiqui et al., 2020] and ℓ_∞ vs. ℓ_2 norm respectively. Likewise, robustness metrics vary between accuracy drop and attack-success rate (ASR). For comparing different models on robustness, accuracy drop is the preferred choice to infer total impact and overall robustness. While models (e.g., TS2Vec) have been explored [Li et al., 2024] across different adversarial attacks and results are consistent per paper, different models can be hardly compared due to the inconsistency of parameter and attack choice in the literature. This study leverages the full 128-dataset UCR archive with unified budgets, providing a standardized benchmark and the first cross-taxonomy robustness comparison on SSL models. Most studies [Fawaz et al., 2019, Karim et al., 2021, Pialla et al., 2025] focus on untargeted, individual, white-box attacks, while the rest of the attack taxonomy space is underexplored, especially black-box attacks. Fawaz et al. [2019] explored transferability across DNNs investigating that adversarial examples can be transferred across architectures. In fact, transfer attacks among SSL time-series classification models have not yet been explored, despite offering direct insights into robustness comparisons between models. Furthermore, representation transferability across unsupervised representation learning encoders is unmeasured. Unlike prior work, this study offers the first systematic comparison of adversarial robustness in SSL architectures. This study benchmarks three SSL encoders the UCR datasets, fixing a single perturbation budget (ℓ_∞, $\varepsilon = 0.1$) to guarantee parameter parity. Each model is set to matching hyperparameters i.e. embedding dimension, batch-size and attacked by five methods that span the white-, gray-, and black-box spectrum. Robustness is measured via accuracy-drop metrics (ΔA, $\Delta_R A$). Friedman followed by Nemenyi tests rank both models and attacks, thus yielding a statistically rigorous, cross-taxonomy robustness benchmark for SSL time-series models.

Chapter 3

Methods

Let $\mathbf{x} \in \mathbb{R}^T$ denote a univariate time-series sample of length T with class label $y \in [0, K-1]$, where K is the number of classes. A SSL encoder $f : \mathbb{R}^T \to \mathbb{R}^d$ maps $\mathbf{x}$ to a latent representation $\mathbf{z} = f(\mathbf{x})$, optimized by an intermediate loss $\mathcal{L}_{SSL}$ on unlabeled data. For downstream classification, a classifier $h : \mathbb{R}^d \to [0, K-1]$ predicts

$$y = \arg_{\max} h_k(f(\mathbf{x})).$$

Adversarial attacks in time-series classification involve deliberately applying small perturbations (ε) to an input time series $\mathbf{x} \in \mathbb{R}^T$ to generate a similar but subtly altered version $\mathbf{x}_{adv} = \mathbf{x} + \delta_{\mathbf{x}}$ where $\mathbf{x}_{adv}, \delta_{\mathbf{x}} \in \mathbb{R}^T$ [Li et al., 2024] . The intention of the attack is to change the predicted label of the model. This process is formally described by the following:

$$\arg_{\max} f(\mathbf{x}) \neq \arg_{\max} f(\mathbf{x} + \delta_{\mathbf{x}}), \quad \text{s.t.} \quad \|\delta_{\mathbf{x}}\| \ll \|\mathbf{x}\|. \tag{3.1}$$

A successful adversarial attack, where ε controls the magnitude of the attack, occurs when the predicted label on the perturbed input differs from the original true label.

$$y \;=\; \arg_{\max} h_k(f_k(\mathbf{x})) \quad \text{and} \quad y' \;=\; \arg_{\max} h_k(f_k(\mathbf{x} + \delta_{\mathbf{x}}))$$

Trivially, it is required that:

$$y \neq \; y' \quad \text{while} \quad \|\delta_{\mathbf{x}}\|_p \leq \varepsilon$$

where $p = \infty$ in this study.

3.1 Adversarial Attacks

Rathore et al. [2020] defined an adversarial attack taxonomy among three dimensions being adversary knowledge, perturbation scope, and attack goal. The following bulleted list describes the two extremes of each dimension.

- ***Adversary Knowledge:***
  - **White-box attacks** assume the adversary has full access to the architecture and weights of the model, enabling exploitation of model gradients.

  - **Black-box attacks** operate without access to the internal model parameters and instead rely solely on observing the model's outputs in response to input queries.

- ***Attack Goal:***
  - **Targeted attacks** aim to mislead the model into predicting a specific label y' chosen by the adversary.
  - **Untargeted attacks** are satisfied with any incorrect label as long as it is not the true label.

- ***Perturbation Scope:***
  - **Individual attacks** generate a unique perturbation δ_x for each input x.
  - **Universal attacks** learn a single perturbation δ_U that can be applied to any input to cause misclassification across multiple samples.

Figure 3.1 shows the attack taxonomy space[1] with the three dimensions and corresponding extremes on the axes. This novel attack space taxonomy can be built on for future work and allows for clear classification of novel attacks. Moreover, the taxonomy provides a foundation for easily classifying gray-box attacks, where the adversary has partial access to the model internals. Gray-box attacks like T-PGD can be defined according to the amount of knowledge access along all three continuous dimensions. Similarly, semi-targeted, and group-level attacks can be categorized using the taxonomy. Semi-targeted attack types neither aim to push the model towards a single class label nor to any erroneous class label, but towards a certain cluster of class labels to misclassify. Group-level attacks sit between universal and individual attacks, where a certain group or batch is attacked per perturbation vector.
To yield attack results that map directly to the risk factor in the industry, it is effective to explore untargeted attacks as a comparative baseline because in critical domains any misclassification is harmful. The reduced attack taxonomy space is visualized in Figure 3.2. The attack space relevant to the study is marked with a blue-red sub-cube. As the relevant attack taxonomy space can be represented along two dimensions by collapsing the goal axis, resulting in the attack space being projected onto the fully untargeted plane marked red in Figure 3.2. All the attacks applied in the paper inhabit this plane as visualized in Figure 3.3.
In the following, all construction pipelines of attacks are detailed. New variables are listed with a description once, while being omitted thereafter for clarity. Note that ε, which is set to 0.1 by default throughout the experiments, is the attack radius that determines the magnitude of the adversarial perturbations. The ℓ_∞ norm of a time–series is defined as $\|\mathbf{x}\|_\infty = \max_t |x_t|$ and is used for the attack to evaluate on the worst case scenario. Also note that an additional query set is always split from the UCR train set, while maintaining separate train and test sets. The query set serves two purposes: it provides data for training the DE and the surrogate model in T-PGD; and it simulates an adversarial setting where the attacker has access to a separate pool of unlabelled samples from the same data domain, but not to the target model's train/test data.

[1]The latex code for recycling the attack taxonomy space can be found at: tsc_ar

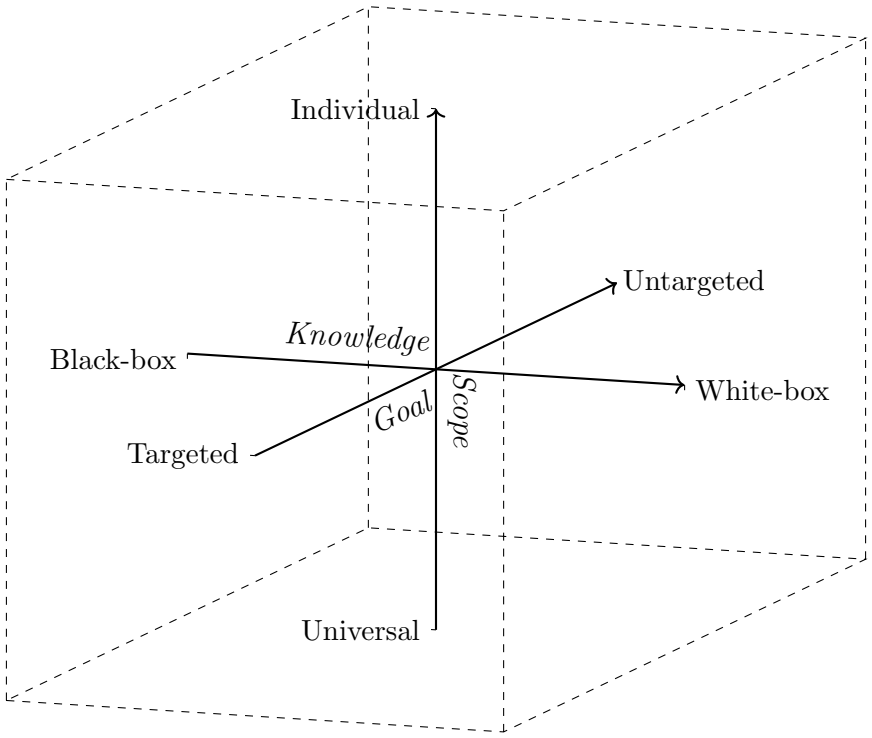


Figure 3.1: The figure shows the adversarial attack taxonomy space. The dashed cube shows the entire attack space along the three dimensions: Adversarial knowledge (x-axis), and perturbation scope (y-axis), and attack goal (z-axis). For clarity all axis names are written cursive. The labels at the ends of the axis show the different extremes of the dimensions.

3.1.1 Individual White-Box Attack

The FGSM method as applied by Fawaz et al. [2019] is the simplest method allowing for computationally cheap attacks by leveraging the model loss and weight access. It provides a useful baseline for white-box attacks that perturb each sample individually as shown in Algorithm 1. The $[\min_d, \max_d]$ values represent the highest and lowest values of the test set. Note that the CLAMP-function clips values to the range $[\min_d, \max_d]$, such that values below $\min_d$ are set to $\min_d$, and values above $\max_d$ are set to $\max_d$.

Algorithm 1 Fast Gradient Sign Method (FGSM)

Input: clean sample x, true label y, model $f(\cdot;\theta)$, loss $J(\theta, x, y)$, ε
Output: adversarial sample x_{adv}

1: $g \leftarrow \nabla_x J(\theta, x, y)$
2: $\delta \leftarrow \varepsilon \cdot \text{sign}(g)$
3: $x_{\text{adv}} \leftarrow \text{CLAMP}(x + \delta, \min_d, \max_d)$
4: **return** x_{adv}

BIM as described by Fawaz et al. [2019] and detailed in Algorithm 2, also known as Iterative FGSM, extends FGSM by applying multiple small-step perturbations instead of a single large step. At each iteration, BIM computes the gradient of the loss with respect to the input and updates

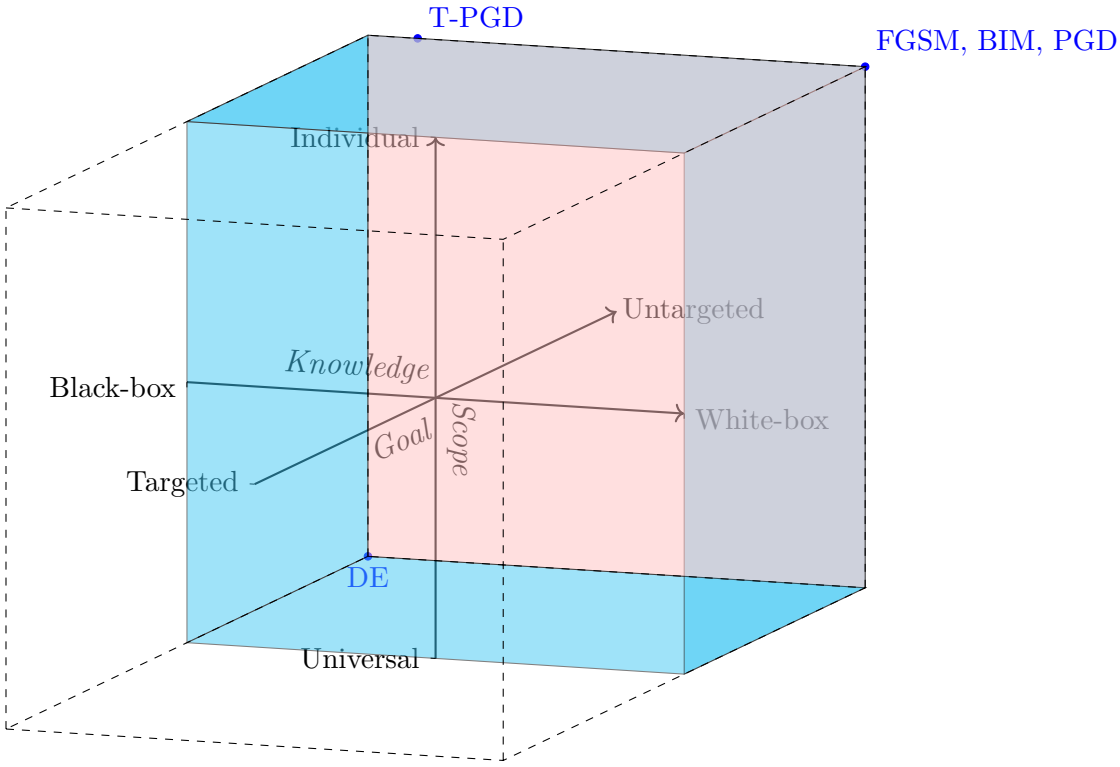


Figure 3.2: Extending Figure 3.1 with the blue and red cube showing the attack space that is investigated in the study. Note that the light blue cube is just a slice of the full attack space as it is constructed by splitting the cube in half on the goal-axis.

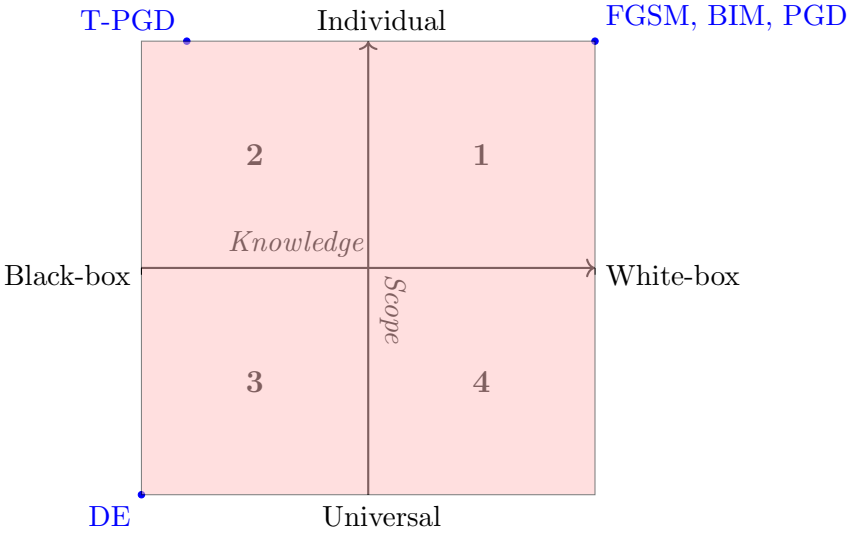


Figure 3.3: A 2-D projection of the untargeted attack taxonomy as marked red in Figure 3.2. The perturbation goal dimension as shown in Figure 3.2 has been collapsed and omitted to show the attack space relevant to this study only. The subcategories as represented by the squares are enumerated for referencing purposes.

the adversarial example by a small step α in the direction of the sign of the gradient. The total perturbation does not exceed the specified budget ε as the updated adversarial example is projected back onto the ℓ_∞-ball of radius ε centered at the original input x (with the PROJECT-function). After each update, the result is also clamped to the valid data range (e.g., $[0, 1]$ for normalized inputs) to ensure that the adversarial sample remains a valid time series.

Algorithm 2 Basic Iterative Method (BIM)

Input: $x, y, \theta, J, \varepsilon, \alpha$ (step size), T (iterations)
Output: x_{adv}

$x_{\text{adv}} \leftarrow x$
for $t = 1$ to T **do**
  $g \leftarrow \nabla_{x_{\text{adv}}} J(\theta, x_{\text{adv}}, y)$
  $x_{\text{adv}} \leftarrow x_{\text{adv}} + \alpha \cdot \text{sign}(g)$
  $x_{\text{adv}} \leftarrow \text{PROJECT}(x_{\text{adv}}, \text{center} = x, \text{radius} = \varepsilon)$
  $x_{\text{adv}} \leftarrow \text{CLAMP}(x + \delta, \min_{\text{d}}, \max_{\text{d}})$
end for
return x_{adv}

Ultimately, the PGD method as shown in Algorithm 3 utilised by Siddiqui et al. [2020] also iteratively adjusts the attack in accordance with the model weights and extends BIM by randomly perturbing the input within the allowed ϵ-ball before iterating (with the UNIFORM-function) to maximize the per-sample misclassification.

Algorithm 3 Projected Gradient Descent (PGD)

Input: $x, y, \theta, J, \varepsilon, \alpha, T$
Output: x_{adv}

$x_{\text{adv}} \leftarrow x + \text{UNIFORM}(-\varepsilon, +\varepsilon)$
$x_{\text{adv}} \leftarrow \text{CLAMP}(x_{\text{adv}}, \text{data\_min}, \text{data\_max})$
for $t = 1$ to T **do**
  $g \leftarrow \nabla_{x_{\text{adv}}} J(\theta, x_{\text{adv}}, y)$
  $x_{\text{adv}} \leftarrow x_{\text{adv}} + \alpha \cdot \text{sign}(g)$
  $x_{\text{adv}} \leftarrow \text{PROJECT}(x_{\text{adv}}, \text{center} = x, \text{radius} = \varepsilon)$
  $x_{\text{adv}} \leftarrow \text{CLAMP}(x + \delta, \min_{\text{d}}, \max_{\text{d}})$
end for
return x_{adv}

3.1.2 Universal Black-Box Attack

Effective universal white-box attacks were introduced by Rathore et al. [2020]. However, no universal black-box attacks utilising natural computing and in specific Differential Evolution (DE) have been proposed yet. Hence, to broaden the robustness comparison, this study newly introduces this attack type. The attack does not have access to the test or train set of the target model, but leverages a small query set, which is from the same domain, disjoint from the data seen by the target model. In this study this is handled by splitting a 15% portion of the original UCR train set for each dataset. A DE universal black-box attack consists of leveraging an original DE implementation by writing a custom fitness function that tries to minimize the average label classification match between the original queries and the perturbed queries. Note that DE is chosen as it performs well on global, gradient-free, continuous problem space [Storn and Price, 1997]. Moreover, time-series inherit meaningful subsequences essential for robust classification, which offers attack surface to meaningfully perturb such subsequences by cross-over, selection and mutation. Algorithm 5 details how the DE with the "best1bin" method works. First, a population of random vectors are created with the length bounds of the time-series in the query set. These vectors represent candidate solutions to produce δ_{opt}, which will be added to the test set of the target model to universally perturb the time-series with the same adversarial vector. Then for multiple generations mutation, crossover, and selection are performed to create a parent and child sample, and after each generation the best candidate vector is updated. Mutation adds a weighted difference of two other samples in the population to the parent sample, then the crossover rate determines the probability with which each value of index j is passed on to the child sample. Selection determines whether the child sample will replace the parent sample based on their corresponding fitness. Algorithm 4 predicts the original labels of the query batch by querying the target model in l. 2. and defines the bounds in respect to the shape and values of the query samples. Algorithm 4 leverages a fitness function that is passed to the selection process in the DE. This fitness function smooths the δ_{flat} to camouflage the attack and to minimize high frequency noise, clips it to the valid bounds and predicts the class labels of all samples perturbed with δ_{flat}. The metric that the function is minimizing for is the average label mismatch between all clean and perturbed queries. After deploying Algorithm 5, the input is smoothed and clipped once again before universally adding the adversarial vector δ_{flat} to the test set to ensure validity of each test sample.

Algorithm 4 DE Black-Box Attack

Input: test set X, query set X_{query}, model $\rightarrow$ predict_fn, data_range, attack bound ε, popsize, max_iter, smoothing parameter σ
Output: $X_{\text{adv}},\ \delta_{\text{opt}}$

$X_{\text{eval}} \leftarrow X$
$\text{preds}_{\text{orig}} \leftarrow \text{predict\_fn}(X_{\text{query}})$
$\text{shape} \leftarrow \text{shape}(X_{\text{query}})[1:]$
$D \leftarrow \prod(\text{shape})$
$\text{bounds} \leftarrow [(-\varepsilon, \varepsilon)]^D$
function FITNESS(δ_{flat})
    $\delta \leftarrow \text{SMOOTH}(\delta_{\text{flat}}.\text{reshape}(\text{shape}), \sigma)$
    $\delta \leftarrow \text{CLIP}(\delta, -\varepsilon, \varepsilon)$
    $X_{\text{adv}} \leftarrow \text{CLIP}(X_{\text{query}} + \delta, \text{data\_range})$
    $\text{preds} \leftarrow \text{predict\_fn}(X_{\text{adv}})$
    return $\text{MEAN}(\text{preds} == \text{preds}_{\text{orig}})$
end function
result $\leftarrow$ DIFFERENTIAL_EVOLUTION(
FITNESS,
bounds,
popsize = popsize,
maxiter = max_iter,
mutation = m,
recombination = r,
maxiter = max_iter)
$\delta_{\text{opt}} \leftarrow \text{SMOOTH}(\text{result.x.reshape}(\text{shape}), \sigma)$
$X_{\text{adv}} \leftarrow \text{CLIP}(X_{\text{eval}} + \delta_{\text{opt}}, \text{data\_range})$
return $X_{\text{adv}},\ \delta_{\text{opt}}$

Algorithm 5 Differential Evolution (best1bin)

Input: Fitness $f : \mathbb{R}^D \to \mathbb{R}$, Search domain $[l, u]^D$, Population Size N_P, Mutation Factor F, Crossover Rate CR, generations G
Output: solution vector x_{best}

Initialize: $\{x_i^0\}_{i=1}^{N_P} \sim \mathcal{U}([l, u]^D)$, $f_i^0 = f(x_i^0)$, $x_{\text{best}}^0 = \arg\min_i f_i^0$
for $t = 1, \ldots, G$ **do**
    for $i = 1, \ldots, N_P$ **do**
        Mutation
        Pick distinct $r_1, r_2 \neq i$
        $v = x_{\text{best}}^{t-1} + F \cdot (x_{r_1}^{t-1} - x_{r_2}^{t-1})$
        Clip $v \in [l, u]^D$
        Crossover
        Choose $j_{\text{rand}} \in \{1, \ldots, D\}$
        $u_j = \begin{cases} v_j & \text{if } \text{rand}_j \leq \text{CR or } j = j_{\text{rand}} \\ x_{i,j}^{t-1} & \text{otherwise} \end{cases}$
        Selection
        if $f(u) < f(x_i^{t-1})$ **then**
            $x_i^t = u,\ f_i^t = f(u)$
        else
            $x_i^t = x_i^{t-1},\ f_i^t = f_i^{t-1}$
        end if
    end for
    Update best: $x_{\text{best}}^t = \arg\min_i f_i^t$
end for
return x_{best}^G

3.1.3 Individual Gray-box attack

To create a transfer attack similar to Fawaz et al. [2019] and Karim et al. [2021], a surrogate model θ_s is employed that acts as a proxy for the true (target) model. The surrogate model is trained from scratch on a dedicated query set, which is a dataset partitioned to be disjoint from both the train and test splits of the target model analogous to Section 3.1.2. This guarantees that the surrogate model is not exposed to any data used by the target model, thus maintaining the limited adversarial knowledge, while imitating representations of the target model. While Zhang et al. [2020] explored FGSM, and Fawaz et al. [2019] investigated FGSM and BIM transfer attacks, this study leverages a Transfer Projected Gradient Descent (T-PGD), which is novel in the literature. The purpose of this surrogate model is to merely supply the gradients for the PGD attack on the test set of the target model. While the adversarial has no access to the model internals or the train set, the test set of the target model is accessed identical to the original PGD attack as in Section 3.1.1. So, when taxonomized precisely, the T-PGD it is not a full black-box attack, but a gray-box attack as shown by Figure 3.3. Note that the surrogate model and target model are trained the same way, where the only difference is they are being trained on the query set and train set, respectively. The full T-PGD attack as shown in Algorithm 6 is crafted almost identical to Algorithm 3, but the gradients for the T-PGD are derived from the surrogate model (θ_s) as indicated in l. 4.

Algorithm 6 Transfer Projected Gradient Descent (TPGD)

Input: x, y, surrogate model θ_s, J, ε, α, T
Output: x_{adv}

1: $x_{adv} \leftarrow x + \text{UNIFORM}(-\varepsilon, +\varepsilon)$
2: $x_{adv} \leftarrow \text{CLAMP}(x + \delta, \min_d, \max_d)$
3: **for** $t = 1$ to T **do**
4: $\quad g \leftarrow \nabla_{x_{adv}} J(\theta_s, x_{adv}, y)$ $\quad \rightarrow$ *surrogate gradient*
5: $\quad x_{adv} \leftarrow x_{adv} + \alpha \cdot \text{sign}(g)$
6: $\quad x_{adv} \leftarrow \text{PROJECT}(x_{adv}, \text{ center} = x, \text{ radius} = \varepsilon)$
7: $\quad x_{adv} \leftarrow \text{CLAMP}(x + \delta, \min_d, \max_d)$
8: **end for**
9: **return** x_{adv} $\quad \rightarrow$ *for target model*

3.2 Self-Supervised Learning models

All models are implemented as presented in the original published papers that introduced the models. The model data pipeline has been adapted for each model such that the train-test splits from the original UCR archive can be used directly for classification. Note that the models correspond to the SSL classes in Figure 2.1. The architectural choices of SSL models, such as augmentation strategy, loss function design, directly influence their susceptibility to adversarial perturbations. The learned representation space and the transferability of adversarial gradients are affected by such choices. In the following, architectural components are detailed of the models, highlighting design factors relevant for adversarial robustness analysis.

1. **Series2Vec** [Foumani et al., 2024] learns representations by directly predicting the similarity between pairs of time series in both temporal and spectral domains, unlike most existing contrastive approaches that rely on hand-crafted data augmentations. It uses time-series-specific similarity measures as self-supervised targets, such as Soft-DTW for the temporal domain and Euclidean distance for the frequency domain, instead of an InfoNCE loss. While training, the model encodes each time series into both temporal and frequency domain representation to enhance similarity preservation among similar samples. The loss directs the learned representations towards respecting the true similarities among time series, without inducing augmentation noise. So, the model belongs to the instance-level contrastive methods in Figure 2.1.

2. **TS2Vec**[Yue et al., 2022] is a contrastive method on the temporal-level (Figure 2.1) and has shown to achieve state-of-the-art accuracy on the UCR datasets. TS2Vec fosters contextual consistency by creating representations of the same timestamp from different augmented views with masking and cropping. The training objective combines a hierarchical temporal contrastive loss and an instance-wise contrastive loss that distinguishes different series at each timestamp with a temporal CNN using residual blocks. TS2Vec has been the first SSL model to showcase successful integration of a hierarchical temporal loss. Its projection head, which is a simple multi-layer-perceptron (MLP), maps each time step embedding into a contrastive space via InfoNCE losses applied across temporal hierarchies.

3. **TS-TCC** [Eldele et al., 2022] combines two SSL classes, namely, temporal-level contrastive and predictive models. It is predictive because one of its core pretext tasks is forecasting future latent representations and training the model to predict and distinguish the next K steps of a weakly augmented series from other negative steps in the batch, enforcing temporal continuity. Additionally, it is contrastive because it applies an InfoNCE-style loss on paired context vectors (c_t^w, c_t^s), minimizing distance of corresponding time-step embeddings across weak (c_t^w) and strong (c_t^s) augmentations, while maximizing distance of all other contexts, corresponding to the temporal-level contrastive methods. This is implemented with the shared 1D-CNN encoder with dual heads for contrastive and predictive learning. Overall, TS-TCC is a hybrid model that both predicts future latent features and contrasts context embeddings across views, and is therefore classified under both predictive and contrastive methods.

3.3 Evaluation Metrics

Following Yue et al. [2022], accuracy was used to assess the classification results for each model performance on the different UCR datasets. Model robustness to adversarial attacks is measured by the difference between the classification accuracy of the clean and perturbed test set. A significant drop in the accuracy of the perturbed test set compared to the clean test set is considered a successful attack, indicating a lack of model robustness.

- A_{clean} denote the classification accuracy on the clean test set,
- A_{adv} denote the classification accuracy on the adversarially perturbed test set,

- ΔA and $\Delta_R A$ represents the robustness degradation termed accuracy drop and relative accuracy drop, where:

$$\Delta A = A_{\text{clean}} - A_{\text{adv}} \tag{3.2}$$

$$\Delta_R A = \frac{A_{\text{clean}} - A_{\text{adv}}}{A_{\text{clean}}} \tag{3.3}$$

In addition to reporting accuracies, attack results are visualized by t-SNE plots that show the clean and perturbed time-series as data instances in a graph. Moreover, t-SNE allows visualization of perturbation strength of the time-series by computing high-dimensional representation that is mapped to the low-dimensional counterpart based on a distance-based probability that one instance is the neighbour of another instance. Moreover, to define the property of transferability, one can utilise ΔA and $\Delta_R A$.
Let f_A and f_B be two encoders trained on the same task but using different architectures or learning strategies. Let $\mathcal{A}$ be an adversarial attack algorithm that generates adversarial examples $\mathbf{x}_{\text{adv}}$ based on a surrogate model $f_{surrogate} = f_s$. Define the relative accuracy drop ($\Delta_R A$) from clean to adversarial inputs as:

$$\Delta_R A(f_t \mid f_s) = \frac{A_{clean}(f_t, \mathbf{x}_{\text{clean}}) - A_{adv}(f_t, \mathbf{x}_{\text{adv}})}{A_{clean}(f_t, \mathbf{x}_{\text{clean}})} \tag{3.4}$$

where:

- $f_{target} = f_t$ is the target model,
- $\mathbf{x}_{\text{adv}} = \mathcal{A}(f_s)$ is adversarial samples generated using f_s on $\mathcal{A}$
- $A_{clean}(f, \mathbf{x})$ denotes the classification accuracy of model f on input $\mathbf{x}$.

Thus, symmetric transferability holds between f_A and f_B under attack $\mathcal{A}$ if:

$$\Delta_R A(f_A \mid f_B) \approx \Delta_R A(f_B \mid f_A) \tag{3.5}$$

Meaning both models suffer a comparable relative accuracy degradation when attacked using adversarial examples generated from the other. This definition can be equivalently stated using the absolute accuracy drop (ΔA) and omitting the "R"-subscript.

Chapter 4

Experiment

The aim of the thesis is to provide a fair evaluation of multiple attack types as proposed in Section 3.1 across TS2Vec, TS-TCC, and Series2Vec. In particular, the study aims to answer the following questions:

Research Questions

- **How robust are state-of-the-art unsupervised representation learning models for time-series classification against adversarial attacks, as measured by their impact on classification accuracy?**
- **Which SSL model architectures are especially prone to certain types of adversarial attacks, and what characteristics explain their robustness or vulnerability?**
- **How do the embedding spaces of Series2Vec, TS2Vec, and TS-TCC influence the transferability and robustness of adversarial examples across models?**

The UCR dataset provided by Dau et al. [2018] serves as the baseline for all evaluations because it consists of 128 datasets from different domains extracted from a multitude of real-world examples. Moreover, the original UCR test and train sets are z-normalized and nan-values are substituted with zeros following Yue et al. [2022] to allow reproducibility.
The experiment pipeline, including the adversarial perturbations, is visualized; each model was initially trained and evaluated on clean data using the original UCR train-test splits to establish a baseline accuracy (A_{clean}) as represented in blue in Figure 4.1. Subsequently, each model was subjected to each adversarial attack, which crafts a perturbed test set, specifically FGSM, BIM, PGD (white-box individual attacks), DE (black-box universal attack), and T-PGD (gray-box individual attack) as shown in red in Figure 4.1. For the T-PGD, each model also acted as a surrogate model to generate adversarial perturbations applied to the other models, providing insight into model transferability and cross-model robustness.
For each attack scenario, adversarial perturbations were constrained by predefined attack budgets using ℓ_∞-norm to comprehensively evaluate model vulnerability. Generally, the attack magnitude ϵ was set to 0.1 throughout experiments. The robustness metrics ΔA and $\Delta_R A$ as defined in Section 3.3 were computed individually for each combination of model, attack method, and dataset. Note that each attack-model-dataset tuple was run once for the same seed.

Finally, the same Friedman test and Nemenyi post hoc tests were used to enable fair statistical comparisons. These tests were used twice: once to compare all models across attack types and the second time to compare attack types per model. Statistical significance of accuracy differences was established at a p-value < 0.05 for all statistical tests throughout the study. The first Friedman test was applied to each model, where the resulting accuracies of the attacks (including the clean accuracies) were ranked from 1 to 8 for each of the 128 datasets and shown in a Critical Difference (CD) diagram. From these rankings, the Friedman test calculated whether there are statistically significant differences in model performance across the different attack scenarios. The second type of Friedman test was applied for all attack and model pairs across all 128 datasets on the relative accuracy drops ($\Delta_R A$). As the three models had different average accuracies, the results could only be evaluated in a statistically sound way by comparing the accuracy drop relative to the clean accuracy. The significant differences were also visualized with a Critical Difference (CD) graph. This ensures statistically sound evidence on robustness ranking among SSL models under adversarial conditions. Reproducibility is fostered by documenting experimental parameters, including hyperparameter configurations, attack parameters, statistical analysis and experiment scripts. All code, and experimental details are publicly accessible in the provided GitHub repository: tsc_ar[1].

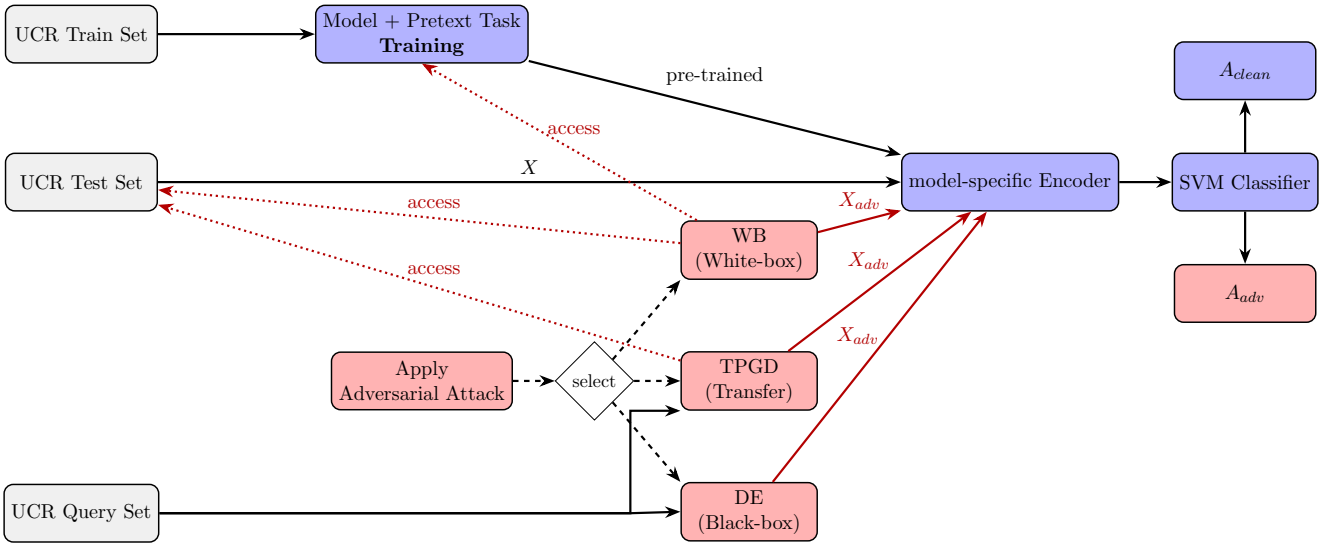


Figure 4.1: Experimental pipeline showing the different adversarial attack creation methods (individual white-box , DE, and transfer attacks). The UCR query set is used for T-PGD and DE attacks. Each set of adversarial samples per attack are processed independently by the trained encoder and Support-Vector-Machine (SVM) classifier to yield the specific A_{adv}. Also, A_{clean} is evaluated on the unperturbed test set X. Note that the red dotted arrows indicate adversarial access.

[1]https://github.com/JesseK18/tsc_ar

Chapter 5

Results

This chapter presents the core findings of the study and answers the main research question in detail by analysing robustness comparisons, specific attack susceptibility, novel attack evaluations, embedding transferability, and design implications for novel robust SSL models.

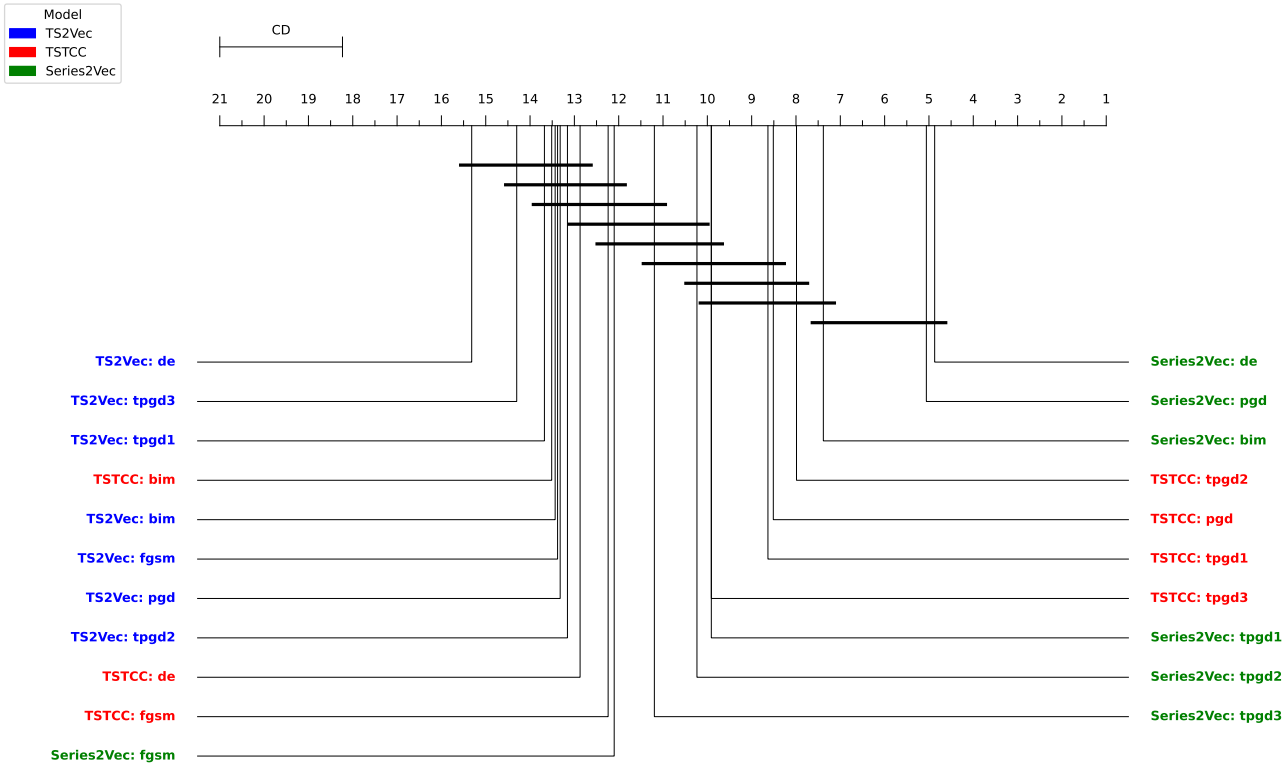


Figure 5.1: A Critical Difference (CD) diagram visualizing the statistically significant differences in $\Delta_R A$ across all attacks and model pairs. Note that the exact numerical values can be found in Table 1. Note that TPGD1, TPGD2, TPGD3 use TS2Vec, TS-TCC, Series2Vec as surrogate model respectively. In the CD diagram, stronger attacks (higher $\Delta_R A$) are positioned on the right, while weaker ones appear on the left. A low rank means that the attack has a high $\Delta_R A$ and, hence, degrades the classification accuracy strongly.

How robust are state-of-the-art unsupervised representation learning models for time-series classification against adversarial attacks, as measured by their impact on classification accuracy? To address this, the absolute and relative classification accuracy drops (ΔA and $\Delta_R A$) were compared under attack for each model, aggregating average accuracies per attack scenario. Statistically significant results are shown in a Critical Difference (CD) diagram Figure 5.1 across all tested UCR datasets. The attack scenarios per model shown in the box-plots in Figures 5.2, 5.4, and 5.6 corresponding to Series2Vec, TS2Vec, and TS-TCC, respectively, demonstrate that differences in robustness across SSL models exist. Moreover, the critical difference graph in Figure 5.1 confirms the significant differences across models. TS2Vec yielded the lowest ΔA and $\Delta_R A$ of all three models and consistently achieved the highest ranks as shown in Figure 5.1. Furthermore, the weakest attacks of TS-TCC and Series2Vec, being BIM and FGSM with a $\Delta_R A$ of 8-10% respectively, resulted in similar $\Delta_R A$ as the strongest attacks on TS2Vec underpinning its robustness. Generally, the resilience of TS2Vec indicates that hierarchical temporal contrasting is superior to the other model architectures with respect to adversarial robustness.

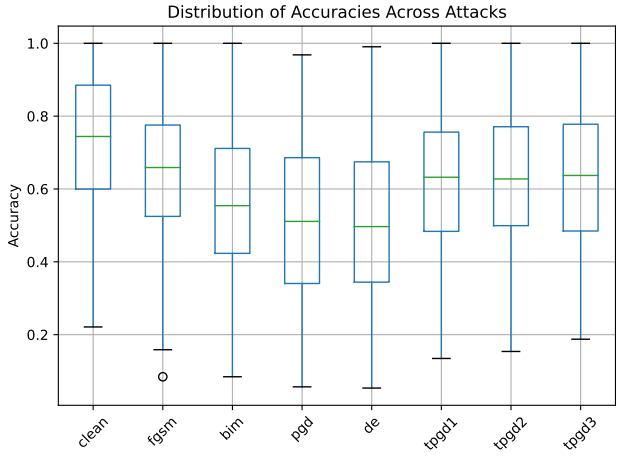


Figure 5.2: A box plot visualizing the results of Table 5.1 for **Series2Vec**.

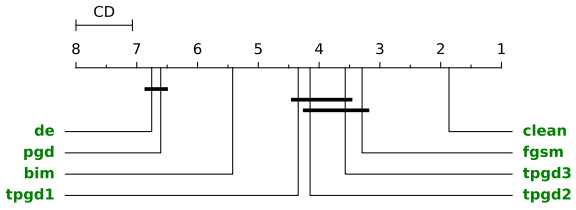


Figure 5.3: A Critical Difference (CD) diagram visualizing statistically significant differences in ΔA across attacks from Figure 5.2 for **Series2Vec**.

Table 5.1: Average absolute, relative accuracy drop and rank per adversarial attack for **Series2Vec**. Note that TPGD1, TPGD2, TPGD3 use TS2Vec, TS-TCC, Series2Vec as surrogate model respectively.

Rank	Attack	Avg. Drop	Avg. R. Drop (%)
1	DE	0.216 ± 0.298	29.11
2	PGD	0.205 ± 0.189	27.74
3	BIM	0.159 ± 0.182	21.34
4	TPGD1	0.109 ± 0.154	14.85
5	TPGD2	0.107 ± 0.150	14.46
6	TPGD3	0.091 ± 0.134	12.11
7	FGSM	0.081 ± 0.140	10.82

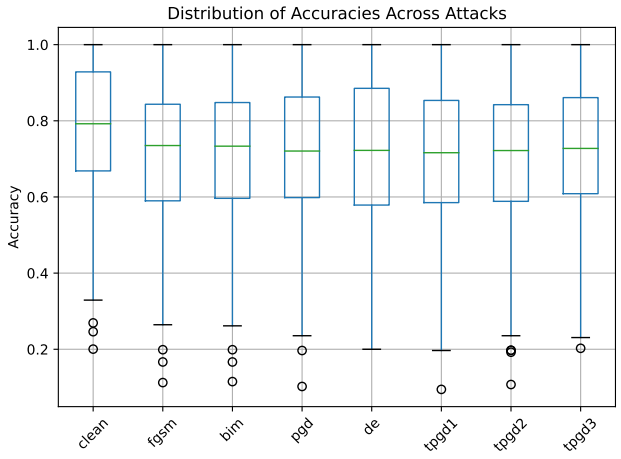


Figure 5.4: A box plot visualizing the results of Table 5.2 for **TS2Vec**.

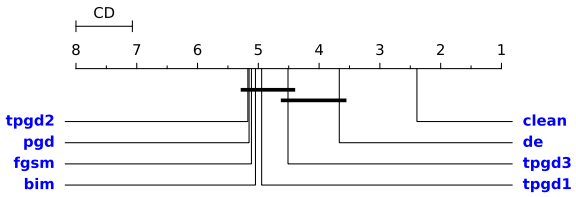


Figure 5.5: A Critical Difference (CD) diagram visualizing the statistically significant differences in ΔA across attacks from Figure 5.4 for **TS2Vec**.

Table 5.2: Average absolute, relative accuracy drop and rank per adversarial attack for **TS2Vec**. Note that TPGD1, TPGD2, TPGD3 use TS2Vec, TS-TCC, Series2Vec as surrogate model respectively.

Rank	**Attack**	**Avg. Drop**	**Avg. R. Drop (%)**
1	TPGD2	0.061 ± 0.105	8.04
2	BIM	0.057 ± 0.106	7.43
3	FGSM	0.057 ± 0.106	7.42
4	TPGD1	0.057 ± 0.103	7.38
5	PGD	0.055 ± 0.101	7.31
6	DE	0.055 ± 0.125	6.92
7	TPGD3	0.051 ± 0.087	6.63

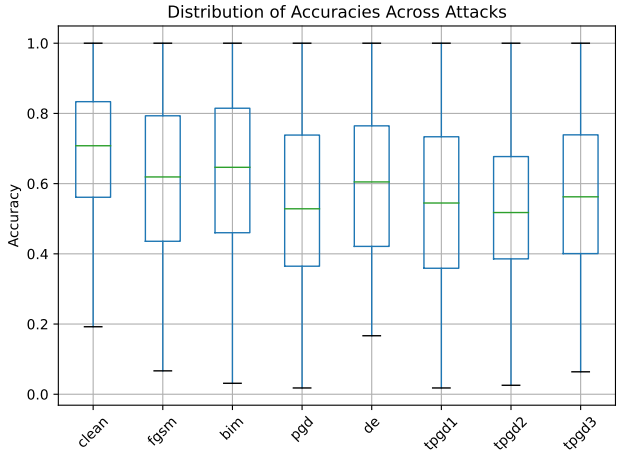


Figure 5.6: A box plot visualizing the results of Table 5.3 for **TS-TCC**.

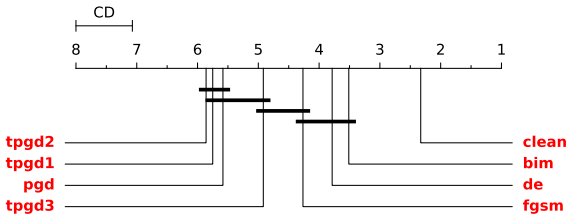


Figure 5.7: A Critical Difference (CD) diagram visualizing the statistically significant differences in ΔA across attacks from Figure 5.6 for **TS-TCC**.

Table 5.3: Average absolute, relative accuracy drop and rank per adversarial attack for **TS-TCC**. Note that TPGD1, TPGD2, TPGD3 use TS2Vec, TS-TCC, Series2Vec as surrogate model respectively.

Rank	Attack	Avg. Drop	Avg. R. Drop (%)
1	TPGD2	0.141 ± 0.156	20.65
2	PGD	0.135 ± 0.162	20.08
3	TPGD1	0.134 ± 0.165	19.87
4	TPGD3	0.112 ± 0.149	16.67
5	FGSM	0.078 ± 0.147	12.02
6	DE	0.073 ± 0.139	10.57
7	BIM	0.054 ± 0.120	8.57

Which SSL model architectures are especially prone to certain types of adversarial attacks, and what characteristics explain their robustness or vulnerability? To yield a per attack comparison for each model, Friedman and Nemenyi post hoc tests were conducted across datasets. Results are shown with box-diagrams in Figures 5.2, 5.4, and 5.6 accompanied with the corresponding CD diagrams in Figures 5.3, 5.5, and 5.7, and the numerical values can be found in Tables 3, 4, and 2. Analysing all single CD diagrams per model one can see that the strength of the white-box attacks are as expected and aligned with the findings of Siddiqui et al. [2020] and Pialla et al. [2025]. For all three SSL models, the attack strength strictly follows: $FGSM \leq BIM \leq PGD$. Series2Vec experienced the largest accuracy degradation under DE and PGD ($\Delta_R A$=29% & $\Delta_R A$=27%), whereas other attacks caused only moderate degradation. The attacks aim to maximize misclassification by targeting instance-specific features and augmentation invariances. When applying DE, adversarial noise is tailored to exploit instance-specific subsequence features with noise that is unknown to Series2Vec. Moreover, Series2Vec has not learned invariances in the data via augmentation, but its features are tightly aligned with the structure of the data as measured by meaningful distances across the spectral and temporal domains. While Series2Vec is equally robust to TS-TCC on T-PGD attacks, TS-TCC is more robust on BIM and DE as shown in Figure 5.1. TS-TCC also achieved similar robustness to TS2Vec under the simplest white-box attack scenarios being BIM and FGSM, as well as under the DE attack. Interestingly, TS2Vec is indifferent to almost all attack types except DE as shown in the CD diagram in Figure 5.5, which yielded the lowest $\Delta_R A$. This can be understood by the fact that the hierarchical temporal contrasting inherently processes and trains on augmented views fostering feature-space invariances. Moreover, as there are almost no significant differences across attack types on TS2Vec and overall $\Delta_R A$ is low with 6-8%, the results consolidate the claim of Yue et al. [2022] that TS2Vec creates universal time-series representations. Similarly to TS2Vec, TS-TCC is most robust to DE attacks ($\Delta_R A$=10%) as augmentation is used for the contrasting pretext task. TS-TCC is more vulnerable to cross-model transfer compared to other attacks as shown in Figure 5.6. As shown in Figure 5.1, the FGSM attack, which is the simplest attack, only minimally perturbed the data without significant cross-model differences. This finding highlights the overall adversarial robustness of utilising embeddings produced by SSL models. Also, looking at the standard deviations for the ΔA across all models in Tables 5.1, 5.2, and 5.3, the standard deviation ranged from 0.08 to 0.20, which is high. Thus, the robustness of models fluctuates across datasets, where datasets with easily distinguishable classes do almost not drop in classification

accuracy, while datasets that contain classes, which are difficult to distinguish and create a narrow decision boundary, get perturbed significantly stronger. This shows that the robustness of SSL models is strongly dependent on the dataset complexity.

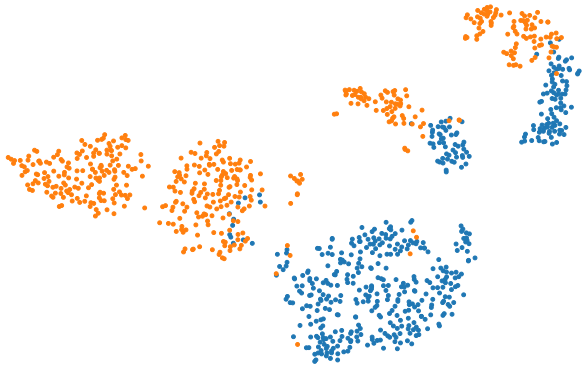
As shown in Figure 5.8[1], the embeddings for the "ItalyPowerDemand" dataset —a dataset that is easy to classify and most simple DNNs classify it correctlyFawaz et al. [2019]— formed similar clusters across models and were very robust against adversarial perturbations from adversarial perturbations from a PGD attack, as clusters did not change. Generally, rotations or mirroring as in Figure 5.8 c) to d) for TS2Vec are meaningless in t-SNE plots Wattenberg et al. [2016], if all other structures remain intact. Accuracy drops for TS2Vec, TS-TCC and Series2Vec are 0%, 7% and 17% respectively. The 2D embedding visualization does not clearly underpin this general trend for TS2Vec being the most robust and Series2Vec being the most vulnerable model, as seen in the t-SNE plots. When looking at the 3D plots, one can see that there are slight differences across the models in the formed clusters. However, it is difficult to attribute the differences in the embeddings to the corresponding accuracy drops as the labels do not show a difference in spatial overlap. Nevertheless, the robustness order is the following: TS2Vec $\leq$ TS-TCC $\leq$ Series2Vec. One possible explanation is that the decision boundaries learned in unison with the SVM-classifier are not visible in the t-SNE plots; thus, it could be that the classifier is better fine-tuned for the embeddings of TS2Vec and less optimal for the embeddings of Series2Vec.
As shown in Figure 5.9[1], all three models produce embeddings with similar ring-like structures in the t-SNE plots, suggesting that their learned representation spaces share a comparable global geometry. The "ChlorineConcentration" dataset is quite complex and easier to perturb than other datasets as its classes are adjacent and differences among classes are more subtle. TS-TCC appeared to have a dense and tightly clustered representation of embeddings. This indicates that its decision boundaries are narrow and highly localized, possibly explaining its relatively lower clean accuracy (e.g., 0.56) on the "ChlorineConcentration" dataset. However, under PGD attack the embeddings are robust as the accuracy drops only minimally to 0.54. This leads to the assumption that TS-TCC gradients are sharp and non-smooth demonstrated by the narrow clustering. Moreover, this sharpness may explain why the T-PGD models were relatively effective compared to the other white-box attacks. When using the exact same sharp gradients of the TS-TCC model for creating the white-box attack, the gradient direction cannot be easily exploited as the steps are not effective due to getting stuck in local minima or flat areas masking gradients [Foret et al., 2021]. However, when creating a model that is trained on disjoint data but with the same architecture as with the T-PGD attacks, the local minima can be escaped. This phenomenon of pseudo-robustness to iterative methods is known as obfuscated gradients [Athalye et al., 2018]. In contrast, Series2Vec yielded a higher clean accuracy of 0.74, but was subject to a high accuracy drop to 0.25. The embeddings remained visually similar before and after perturbation, suggesting that the model lacks ability to adaptat or compensate for the adversarial shift. Hence, this confirms the results from Figure 5.1 that the representation learning of Series2Vec is shallow. TS2Vec balances clean performance and robustness as its accuracy dropped from 0.75 to 0.43 under the PGD attack. The embeddings showed some drift, but retained a coherent structure. This aligns with the observation that TS2Vec learns meaningful and robust features.

[1]Interactive and 3D versions of the t-SNE plots are available at the following website: https://jessek18.github.io/tsc_ar_docs/.

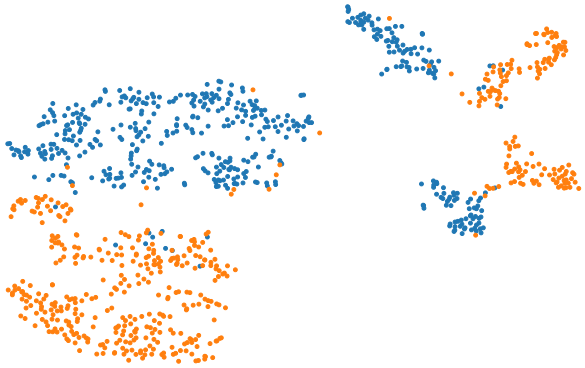
(a) Clean - **Series2Vec** (Accuracy: 0.93)

(b) PGD - **Series2Vec** (Accuracy: 0.76)

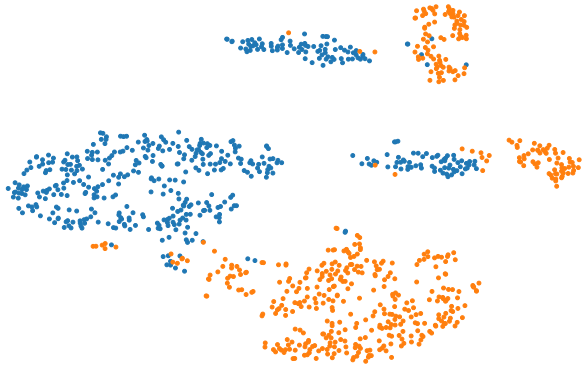
(c) Clean - **TS2Vec** (Accuracy: 0.95)



(d) PGD - **TS2Vec** (Accuracy: 0.95)



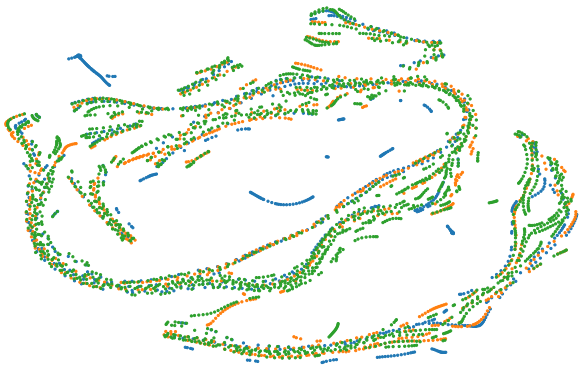
(e) Clean - **TS-TCC** (Accuracy: 0.93)



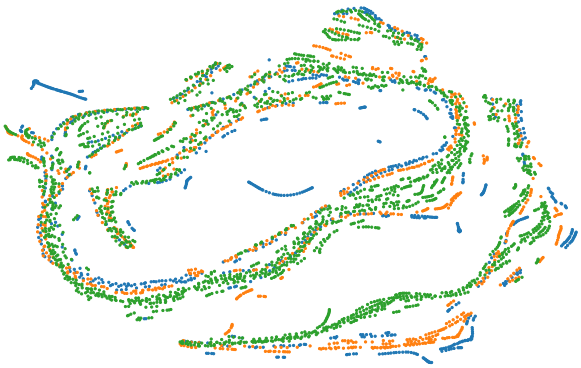
(f) PGD - **TS-TCC** (Accuracy: 0.86)

Figure 5.8: t-SNE embeddings of clean and PGD-perturbed test samples of the "ItalyPowerDemand" dataset (2 classes) for Series2Vec, TS2Vec, and TS-TCC. Left column: clean test embeddings; right column: PGD-perturbed counterparts.

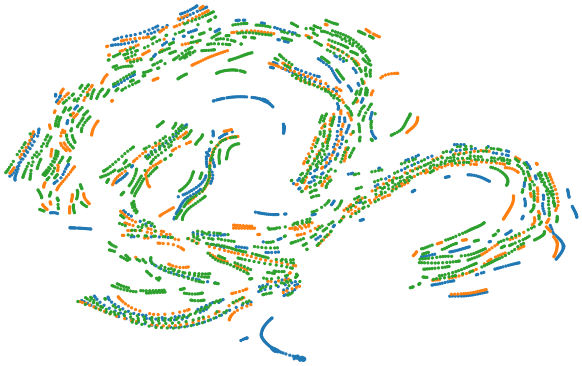
Note that the T-PGD and PGD attacks were all equally strong at the $\Delta_R A$ between 16-20% when attacking TS-TCC. The surrogate models for the T-PGD were always trained on disjoint samples that the target model was not trained on. So, the loss surface of TS-TCC predictive model part may exhibit sharp, locally robust, but globally susceptible regions that can be exploited by models



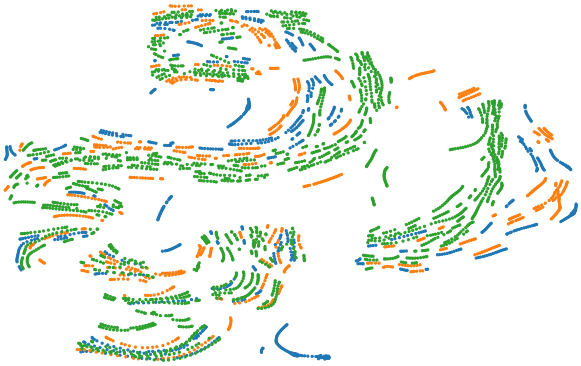
(a) Clean - **Series2Vec** (Accuracy: 0.74)



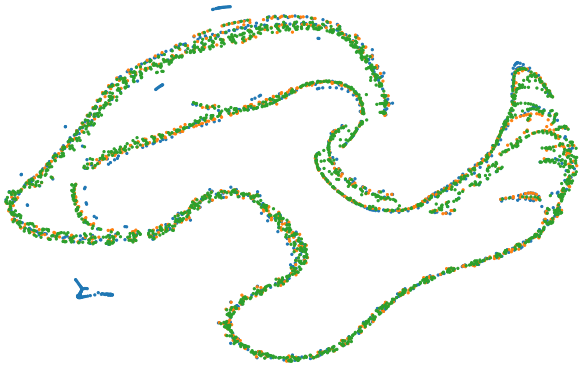
(b) PGD - **Series2Vec** (Accuracy: 0.26)



(c) Clean - **TS2Vec** (Accuracy: 0.75)



(d) PGD - **TS2Vec** (Accuracy: 0.43)



(e) Clean - **TS-TCC** (Accuracy: 0.56)

(f) PGD - **TS-TCC** (Accuracy: 0.52)

Figure 5.9: t-SNE embeddings of clean and PGD-perturbed test samples of the "ChlorineConcentration" dataset (3 classes) for Series2Vec, TS2Vec, and TS-TCC. Left column: clean test embeddings; right column: PGD-perturbed counterparts. The same t-SNE plots with interactivity and in 3D can be found under the following link: https://jessek18.github.io/tsc_ar_docs/.

trained on disjoint data. Furthermore, the surrogate model can explore alternative adversarial loss patterns that escape local minima or gradient masking possibly embedded in the predictive model part of TS-TCC. Generally, the strength of the transfer-based gray-box attacks underpins

the importance of considering surrogate-based adversarial evaluations to assess the vulnerabilities that cannot be exposed by pure individual white-box attacks. Especially, when cross-comparing SSL model types, transfer attacks reveal differences between contrastive and predictive models. While DE was not the strongest attack overall, it is a useful benchmark that strongly sets apart robustness between temporal-level and instance-level contrasting based models. At the same time, DE is a functional black-box attack which is rare in current literature and offers opportunities to simulate the more realistic attack scenarios, where the adversarial has no knowledge about the target model available.

How do the embedding spaces of Series2Vec, TS2Vec, and TS-TCC influence the transferability and robustness of adversarial examples across models? TS-TCC exhibited greater susceptibility when subjected to transfer-based attacks compared to the other attacks as shown in Figure 5.6. The comparative analysis of T-PGD attacks demonstrated that TS2Vec is significantly more robust as a target model than TS-TCC and Series2Vec as target models for all surrogate models including itself as shown in Figure 5.1. However, as shown in Figures 5.3, 5.5, and 5.7, the differences in robustness between Series2Vec, TS2Vec, and TS-TCC as surrogate models were not significant under any transfer scenario across target models. This means that symmetric transferability holds ($\Delta_R A(f_A \mid f_B) \approx \Delta_R A(f_B \mid f_A)$), as defined in Equation 3.3, where each model can be substituted for A or B. This means that each model is equally susceptible to any surrogate model. Hence, the representations of all models inherently create similar representation spaces for time-series features. This indicates that, while TS2Vec consistently exhibited superior robustness to both transfer-based and white-box adversarial attacks, Series2Vec , TS2Vec, and TS-TCC still exhibited largely symmetric transferability. The individual relative model differences in accuracy drops for T-PGD are worth investigating, especially that T-PGD attacks were the strongest attacks for TS-TCC, while being weak attacks for Series2Vec and TS2Vec. As TS-TCC is a hybrid model, inference about the different model types is not trivial. However, there could be two reasons why TS-TCC is relatively more susceptible to T-PGD attacks. The interplay of the predictive and temporal-contrastive loss is especially vulnerable or the loss landscape of TS-TCC, particularly in its predictive component, may exhibit sharp gradients and localized minima as already indicated by the t-SNE plots in Figure 5.9, making it susceptible to perturbations derived from surrogate models trained on disjoint data. Also, as TS2Vec outperforms TS-TCC, and both use temporal contrasting, but TS-TCC uses predictive pretext tasks in hybrid, one can derive a twofold of hypotheses, which could both be equally true. Firstly, temporal-level, instance-level contrasting and predictive models share the same or similar embedding spaces leading to the symmetrical transferability. Secondly, the temporal-level part of TS-TCC drives the robustness up, while the predictive part derogates robustness, resulting in a zero-sum such that there is no significant difference. As TS-TCC performs worse on the T-PGD attacks compared to the other attacks, TS2Vec performs the best overall, and as both TS-TCC (one part of the hybrid) and TS2Vec belong to the temporal-contrastive models, it seems that the second hypothesis is favourable. So, following that logic, predictive methods seem to be less robust in a hybrid model.

To answer what model architecture is inherently most robust, it is evident that temporal-level contrastive models are most robust as TS2Vec achieved the lowest accuracy drops. The emphasis is on the synthesis of hierarchical temporal encoding and an InfoNCE loss. The hierarchical encoding ensures that information is not concentrated at a single temporal scale, while the InfoNCE loss

enforces alignment only for true temporal positives and prevents attacks to target spurious or highly variable aspects of the data. Series2Vec shows that instance-level contrasting was strictly less robust than temporal-level contrasting. Generally, from the results of TS-TCC hybrid models did not show stronger robustness, but might even be susceptible to a broader attack surface due to leveraging multiple distinct pretext tasks. However, other combinations of SSL classes beyond TS-TCC might form more robust hybrid models.

Chapter 6

Conclusion

This study systematically evaluated the adversarial robustness of leading SSL models for classification, revealing significant differences between SSL architectures. TS2Vec demonstrated the strongest adversarial robustness across all attack benchmarks by showing minimal relative accuracy drops of around 8%. The robustness of TS2Vec can be attributed to the hierarchical temporal-level contrastive architecture that augments data and thereby regularizes the embedding space against adversarial perturbations. In contrast, Series2Vec, which is an instance-level contrasting method that does not rely on augmentation, but encodes instance features along the temporal and spectral dimensions, proved the most vulnerable, especially to universal attacks, with average relative accuracy drops of 29%. TS-TCC, representing a hybrid model of temporal-level contrasting, and predictive models, showed moderate robustness, but was more vulnerable to T-PGD, hinting at obfuscated gradients and the vulnerable interplay with the predictive pretext objective. DE was shown to maximally perturb Series2Vec and minimally perturb TS-TCC and TS2Vec, indicating that it is a valuable attack alternative for comparing robustness across model types. Generally, while more unsupervised representation learning models need to be analysed for an exhaustive comparison, the results indicate that SSL models can symmetrically transfer embeddings, sharing robust architecture characteristics. Thus, SSL models effectively distil robust, semantically meaningful representations that are closely aligned with inherent data characteristics.

6.1 Limitations and Further Research

This study focuses exclusively on untargeted attacks instead of targeted attacks due to the comparison and implementation complexity and to ensure baseline comparability in real-world applications. Targeted attacks generate perturbations that foster misclassification towards a specific class or set of classes. This is challenging to implement and interpret across different contrastive models that may possess varying class-specific embedding spaces. Hence, a meaningful comparison of targeted attacks exceeds the ambit of the study. Also, the study evaluates the accuracy across all models, attacks and datasets with the same SVM-classifier, following the protocol of [Yue et al., 2022] to maximize the comparability and the accuracy of the extracted embeddings. The observed symmetry in transferability may be partially affected by the use of a fixed SVM classifier. Future work could strengthen this conclusion by replicating the results across multiple alternative classifiers, thereby measuring the impact of the embedding-classifier dependency under adversarial conditions.

Also, the statistically significant differences in clean average accuracies on the UCR datasets across models pose a bottleneck for comparison. While employing relative accuracy drops helps to address this issue, it does not fully eliminate limitations on the strength of the inferences. Besides, the query set size for black-box attacks is relatively small (0.15% of the original UCR train split), which limits the scope and generalization of the DE and T-PGD. While, the sparse query samples simulate real-world scenarios appropriately, the comparison fairness could be improved by utilising query and train sets of the same sample size. The study can be extended by implementing defensive strategies, such as training on augmented or adversarially perturbed train and test sets using, for example, white noise, Gaussian filtering, or adversarial training with PGD attacks. Additional attack types can be implemented to broaden the generalization of the study, such as frequency domain attacks i.e. Fast Fourier Transformation attacks and adapted variants of TS-Fool [Wang et al., 2024]. Also, comparing ℓ_2 vs ℓ_{inf} as conducted by Siddiqui et al. [2020] norm perturbations could yield valuable insights into difference in embedding representations, ultimately resulting in different $\Delta_r A$. Lastly, the most promising research direction is to implement multiple SSL models to gain a deeper understanding of inherent characteristics of the different SSL architectures. In particular, implementing SSL prototype level contrasting, stand-alone predictive, and adversarial SSL models will enhance the understanding of inherently robust SSL architectures.

Acknowledgements

This thesis was conducted at the Leiden Institute of Advanced Computer Science (LIACS) and built upon foundational ideas and guidance provided by Wadie Skaf and Mitra Baratchi, whose insights significantly shaped the direction of this study. The support of LIACS, through access to the GPU cluster "GRACE" was crucial for carrying out the experiments presented in this study. The creators and maintainers of the UCR Time Series Classification Archive [Dau et al., 2018] are gratefully acknowledged for providing access to a comprehensive and well-curated dataset collection, which was essential for the experimental evaluation in this study.

Bibliography

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, 2018. URL https://arxiv.org/abs/1802.00420.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, 2020.

Mitra Baratchi, Can Wang, Thomas Bäck, Holger H. Hoos, Steffen Limmer, and Markus Olhofer. Towards time-series feature engineering in automated machine learning for multi-step-ahead forecasting. *Engineering Proceedings*, 18(1), 2022. ISSN 2673-4591. doi: 10.3390/engproc2022018017. URL https://www.mdpi.com/2673-4591/18/1/17.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. The ucr time series classification archive, October 2018. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan. Time-series representation learning via temporal and contextual contrasting. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 2352–2359, 2021.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Self-supervised contrastive representation learning for semi-supervised time-series classification. *arXiv preprint arXiv:2208.06616*, 2022.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, and Xiaoli Li. Label-efficient time series representation learning: A review. *IEEE Transactions on Artificial Intelligence*, 5(12):6027–6042, December 2024. ISSN 2691-4581. doi: 10.1109/tai.2024.3430236. URL http://dx.doi.org/10.1109/TAI.2024.3430236.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Adversarial attacks on deep neural networks for time series classification. *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019. doi: 10.1109/IJCNN.2019.8851936.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization, 2021. URL https://arxiv.org/abs/2010.01412.

Navid Mohammadi Foumani, Chang Wei Tan, Geoffrey I Webb, Hamid Rezatofighi, and Mahsa Salehi. Series2vec: similarity-based self-supervised representation learning for time series classification. *Data Mining and Knowledge Discovery*, 38(4):2520–2544, 2024.

Goodfellow, Shlens, and Szegedy. Explaining and harnessing adversarial examples. 2015. URL https://arxiv.org/abs/1412.6572.

Benjamin Le Guen and Nicolas Thome. Timeclr: A self-supervised contrastive learning framework for univariate time series representation. *arXiv preprint arXiv:2302.01528*, 2023. URL https://arxiv.org/abs/2302.01528.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019. doi: 10.1007/s10618-019-00619-1.

Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. Lstm fully convolutional networks for time series classification. *IEEE Access*, 6:1662–1669, 2018. ISSN 2169-3536. doi: 10.1109/access.2017.2779939. URL http://dx.doi.org/10.1109/ACCESS.2017.2779939.

Fazle Karim, Somshubra Majumdar, and Houshang Darabi. Adversarial attacks on time series. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3309–3320, 2021. doi: 10.1109/TPAMI.2020.3022817.

Zhengyang Li, Wenhao Liang, Chang Dong, Weitong Chen, and Dong Huang. Correlation analysis of adversarial attack in time series classification, 2024. URL https://arxiv.org/abs/2408.11264.

Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. In *Proceedings of the 33rd International Conference on Machine Learning (ICML) Anomaly Detection Workshop*, 2016. URL https://arxiv.org/abs/1607.00148.

Qianwen Meng, Hangwei Qian, Yong Liu, Yonghui Xu, Zhiqi Shen, and Lizhen Cui. Unsupervised representation learning for time series: A review. *arXiv preprint*, 2023. URL https://arxiv.org/abs/2308.01578.

Gautier Pialla, Hassan Ismail Fawaz, Maxime Devanne, Jonathan Weber, Lhassane Idoumghar, Pierre-Alain Muller, Christoph Bergmeir, Daniel F. Schmidt, Geoffrey I. Webb, and Germain Forestier. Time series adversarial attacks: an investigation of smooth perturbations and defense approaches. *International Journal of Data Science and Analytics*, 19(1):129–139, 2025. doi: 10.1007/s41060-023-00438-0.

Pradeep Rathore, Arghya Basak, Sri Harsha Nistala, and Venkataramana Runkana. Untargeted, targeted and universal adversarial attacks and defenses on time series. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, July 2020. doi: 10.1109/ijcnn48605.2020.9207272. URL http://dx.doi.org/10.1109/IJCNN48605.2020.9207272.

Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. Benchmarking adversarial attacks and defenses for time-series data. *Neural Information Processing*, pages 544–554, 2020. doi: 10.1007/978-3-030-63836-8_45.

Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997. doi: 10.1023/A:1008202821328.

Patara Trirat, Yooju Shin, Junhyeok Kang, Youngeun Nam, Jihye Na, Minyoung Bae, Joeun Kim, Byunghyun Kim, and Jae-Gil Lee. Universal time-series representation learning: A survey. *arXiv preprint arXiv:2310.01517*, 2023. URL https://arxiv.org/abs/2310.01517.

Yanyun Wang, Dehui Du, Haibo Hu, Zi Liang, and Yuanhao Liu. TSFool: Crafting Highly-Imperceptible Adversarial Time Series through Multi-Objective Attack. *arXiv preprint arXiv:2209.06388*, 2024. URL https://arxiv.org/abs/2209.06388. Code: https://github.com/FlaAI/TSFool.

Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-sne effectively. *Distill*, 2016. doi: 10.23915/distill.00002. URL https://archive.org/details/distill-00002.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series, 2022. URL https://arxiv.org/abs/2106.10466.

Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Y. Zhang, Yuxuan Liang, Guansong Pang, Dongjin Song, and Shirui Pan. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10):6775–6793, 2024. doi: 10.1109/TPAMI.2024.3387317. URL https://doi.org/10.1109/TPAMI.2024.3387317.

Yinghua Zhang, Yangqiu Song, Jian Liang, Kun Bai, and Qiang Yang. Two sides of the same coin: White-box and black-box attacks for transfer learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '20, page 2989–2997. ACM, August 2020. doi: 10.1145/3394486.3403349. URL http://dx.doi.org/10.1145/3394486.3403349.

Appendix

Note that TPGD1, TPGD2, TPGD3 use TS2Vec, TS-TCC, Series2Vec as surrogate model respectively.

Table 1: Friedman Test Statistics and Algorithm Ranks

Statistic	**Value**
Friedman	1275.75
p-value	8.27×10^{-189}
CD	2.77

Algorithm	**Rank**
Series2Vec: de	4.87
Series2Vec: pgd	5.06
Series2Vec: bim	7.38
TSTCC: tpgd2	7.99
TSTCC: pgd	8.51
TSTCC: tpgd1	8.63
Series2Vec: tpgd1	9.91
TSTCC: tpgd3	9.91
Series2Vec: tpgd2	10.23
Series2Vec: tpgd3	11.20
Series2Vec: fgsm	12.10
TSTCC: fgsm	12.24
TSTCC: de	12.87
TS2Vec: tpgd2	13.16
TS2Vec: pgd	13.32
TS2Vec: fgsm	13.38
TS2Vec: bim	13.43
TSTCC: bim	13.51
TS2Vec: tpgd1	13.68
TS2Vec: tpgd3	14.30
TS2Vec: de	15.32

Table 2: CD for **Series2Vec**

Algorithm	**Average Rank**
clean	1.86
fgsm	3.29
tpgd3	3.57
tpgd2	4.15
tpgd1	4.34
bim	5.42
pgd	6.61
de	6.75

Statistics = 853.38, p = 5.5e-108, CD = 0.928

Table 3: CD for **TS2Vec**

Algorithm	**Average Rank**
clean	2.39
de	3.67
tpgd3	4.51
tpgd1	4.95
bim	5.05
fgsm	5.11
pgd	5.15
tpgd2	5.17

Statistics = 956.16, p = 3.1e-127, CD = 0.928

Table 4: CD for **TS-TCC**

Algorithm	**Average Rank**
clean	2.33
bim	3.51
de	3.79
fgsm	4.27
tpgd3	4.92
pgd	5.58
tpgd1	5.75
tpgd2	5.86

Statistics = 883.78, p = 1.2e-113, CD = 0.928

Table 5: Summary statistics (mean, median, standard deviation) of accuracy for each attack for all models. Note that TPGD1, TPGD2, TPGD3 use TS2Vec, TS-TCC, Series2Vec as surrogate model respectively.

Attack	**Series2Vec**			**TS2Vec**			**TSTCC**		
	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
Clean	0.724	0.744	0.190	0.767	0.792	0.183	0.686	0.708	0.194
FGSM	0.643	0.659	0.212	0.710	0.735	0.205	0.607	0.619	0.235
BIM	0.565	0.554	0.221	0.709	0.733	0.205	0.632	0.646	0.232
PGD	0.519	0.511	0.218	0.711	0.721	0.204	0.550	0.528	0.234
DE	0.507	0.497	0.220	0.711	0.722	0.210	0.612	0.605	0.225
TPGD1	0.614	0.632	0.212	0.710	0.716	0.202	0.551	0.545	0.232
TPGD2	0.616	0.628	0.210	0.706	0.722	0.203	0.545	0.518	0.225
TPGD3	0.633	0.637	0.207	0.716	0.727	0.197	0.571	0.562	0.223