Security and Robustness of Time-Series Machine Learning

Jana Doppa
(Joint work w/ Taha Belkhouja and Yan Yan)

Huie-Rogers Endowed Chair Associate Professor of CS

Berry Distinguished Professor of Engineering

School of Electrical Engineering & Computer Science

WASHINGTON STATE UNIVERSITY

Outline

I. Introduction

II. Deep Learning for Time-Series Data through Adversarial Lens

   i. Vulnerability against adversarial attacks

   ii. Novel deep learning adversarial frameworks for time-series data

   iii. Application of adversarial frameworks in wearable sensors

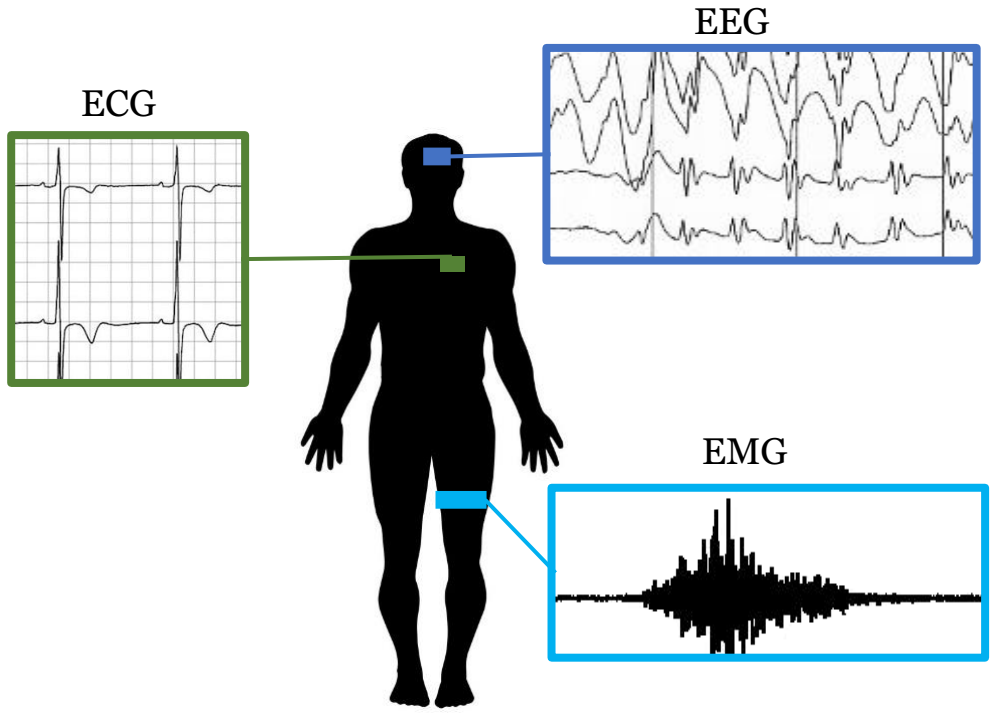
   iv. Robust adversarial training for time-series data

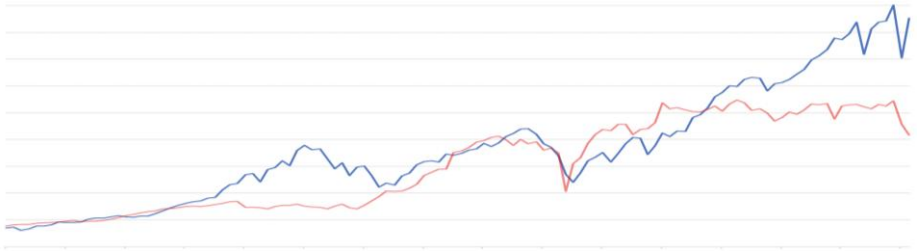
III. Conclusion

Introduction

Time-series Data is Ubiquitous

- IoTs
  - Smart Homes
  - Smart Health
  - Wearables
- Finance
  - Sales/Stocks
  - Customer demand
- Monitoring systems
  - Smart grids





ML Application of Time-Series

- Classification
  - Human Activity Recognition, Medical diagnosis
- Forecasting
  - Weather prediction, Stock prediction
- Imputation
  - Medical data collection
- Outlier detection
  - Monitoring systems

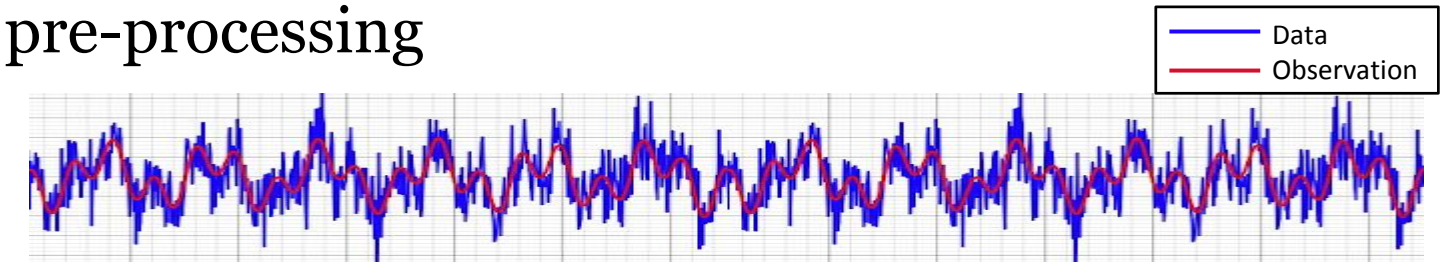
ML Models of Time-Series

- Classical
  - ARIMA
  - VARMA
  - Linear regression
  - Regression random forest

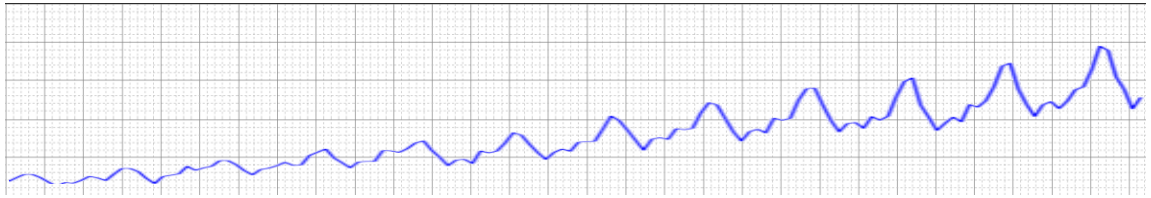
- Deep learning
  - LSTM
  - Inception time-series
  - GAN time series
  - Transformers

Time-Series Challenges

- Noise
- Complexity in pre-processing



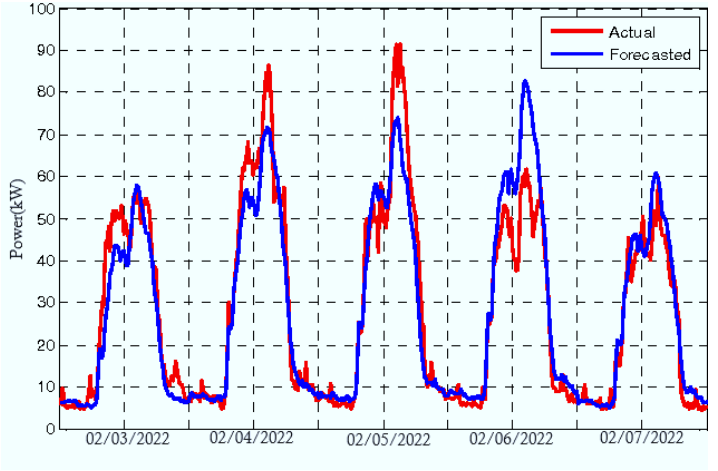
- Modeling temporal behavior
- Periodicity / Stationarity



- Outliers

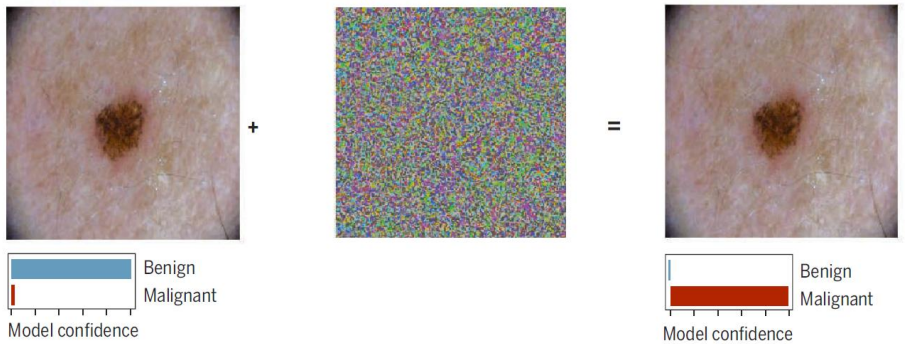
Time-Series Challenges

- The need for robust ML models for time-series data:
  - The data is vulnerable to several corruption threats
  - Importance for safety-critical application
  - Avoid catastrophic scenario
    - Fall detection, medical diagnosis, prediction stability of smart grid

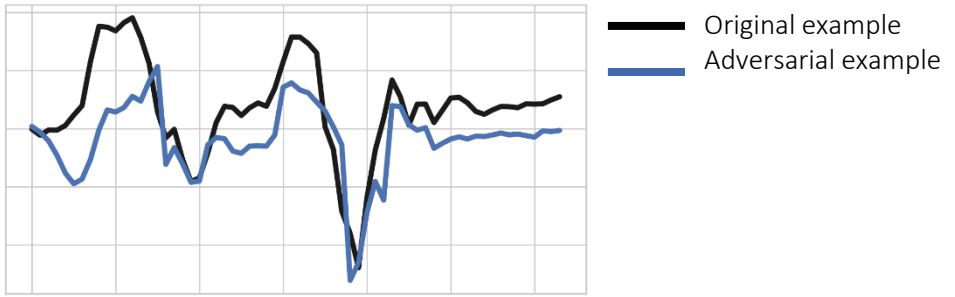


Robustness challenges: Adversarial perturbations

- Medical case example:



Adversarial misrepresentation of the data in image-based AI systems.[1]

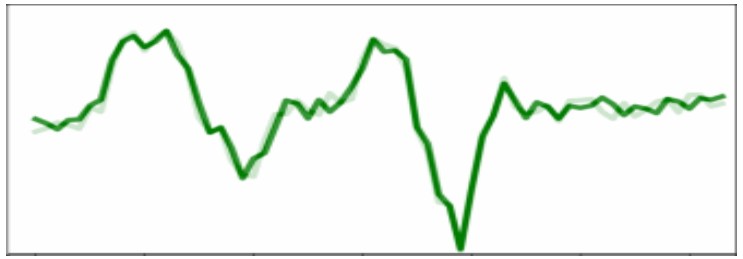


Adversarial misrepresentation of the temporal clinical data in sensor-based AI systems.

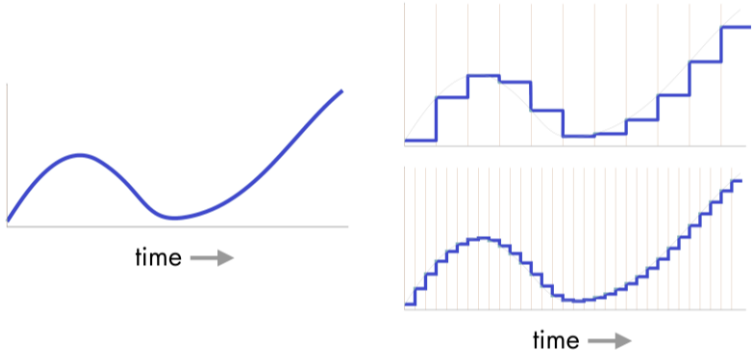
[1]: Finlayson, Samuel G., et al. "Adversarial attacks on medical machine learning." *Science* 363.6433 (2019): 1287-1289.

Robustness challenges: Natural perturbations

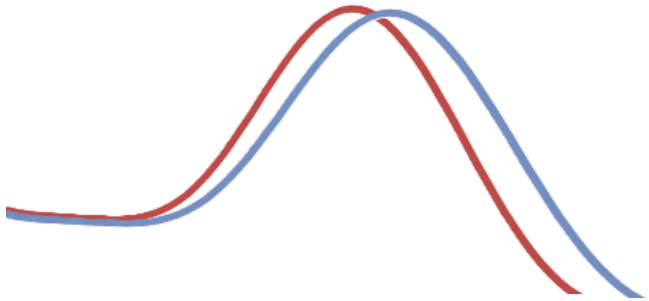
- Random Noise



- Deployment of IoTs in real-world settings:
  - Sensor mis-calibration
  - Sensor's change in orientation
  - Record sampling mismatch



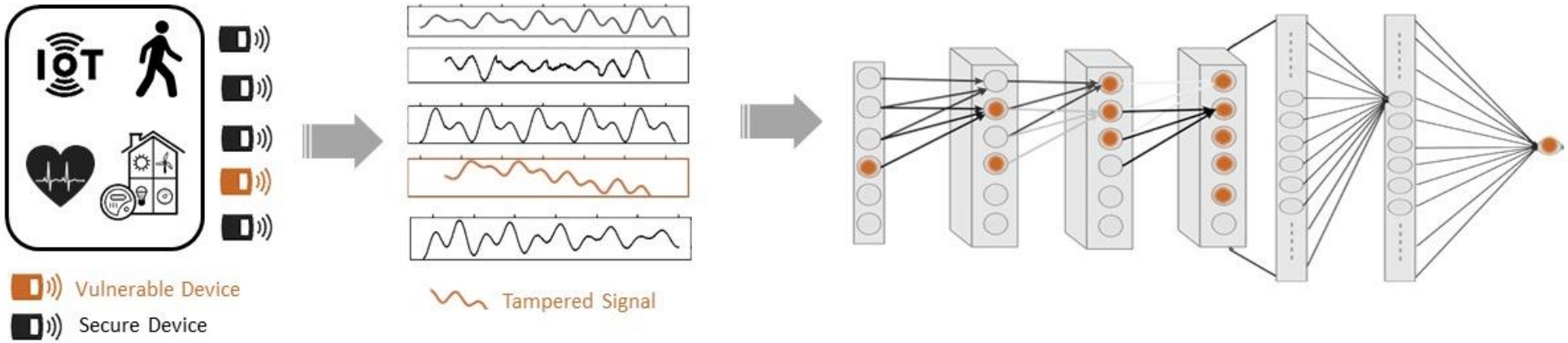
  - Temporal delays



Deep Learning for Time-Series Data through Adversarial Lens

Classification Application for Time-Series

- Human activity recognition
- IoT Monitoring systems
- Medical diagnosis

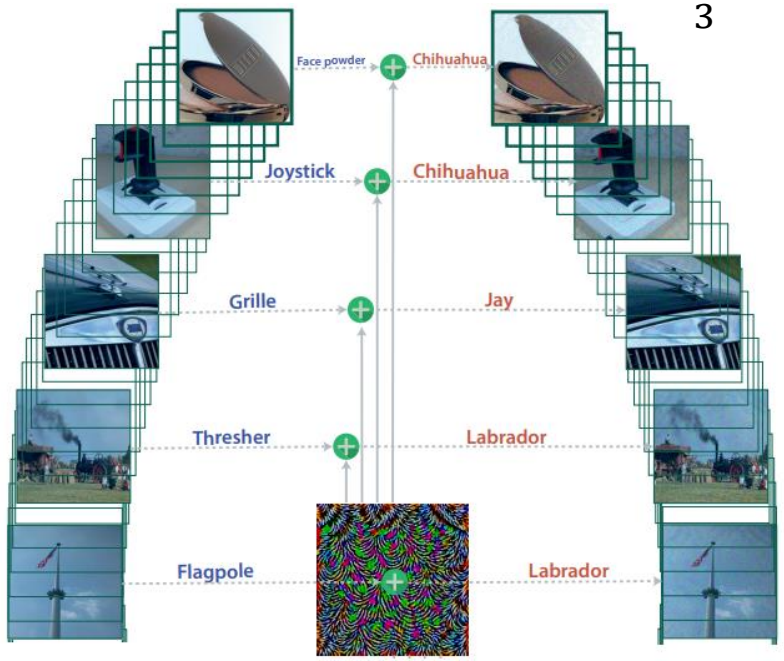


Motivation for Adversarial Robustness

- Prediction reliability against natural and hand-crafted perturbation
- Investigate worst-case scenario within the threat vector
- Improves over standard data augmentation techniques
- Resilient against the over-confidence phenomenon of deep models

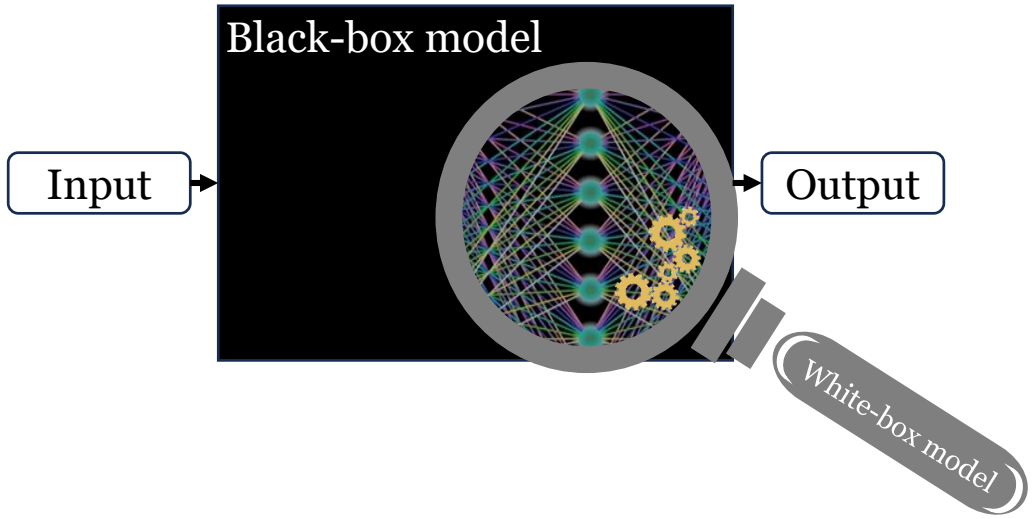
Types of Adversarial Attacks

Single instance vs Universal attack



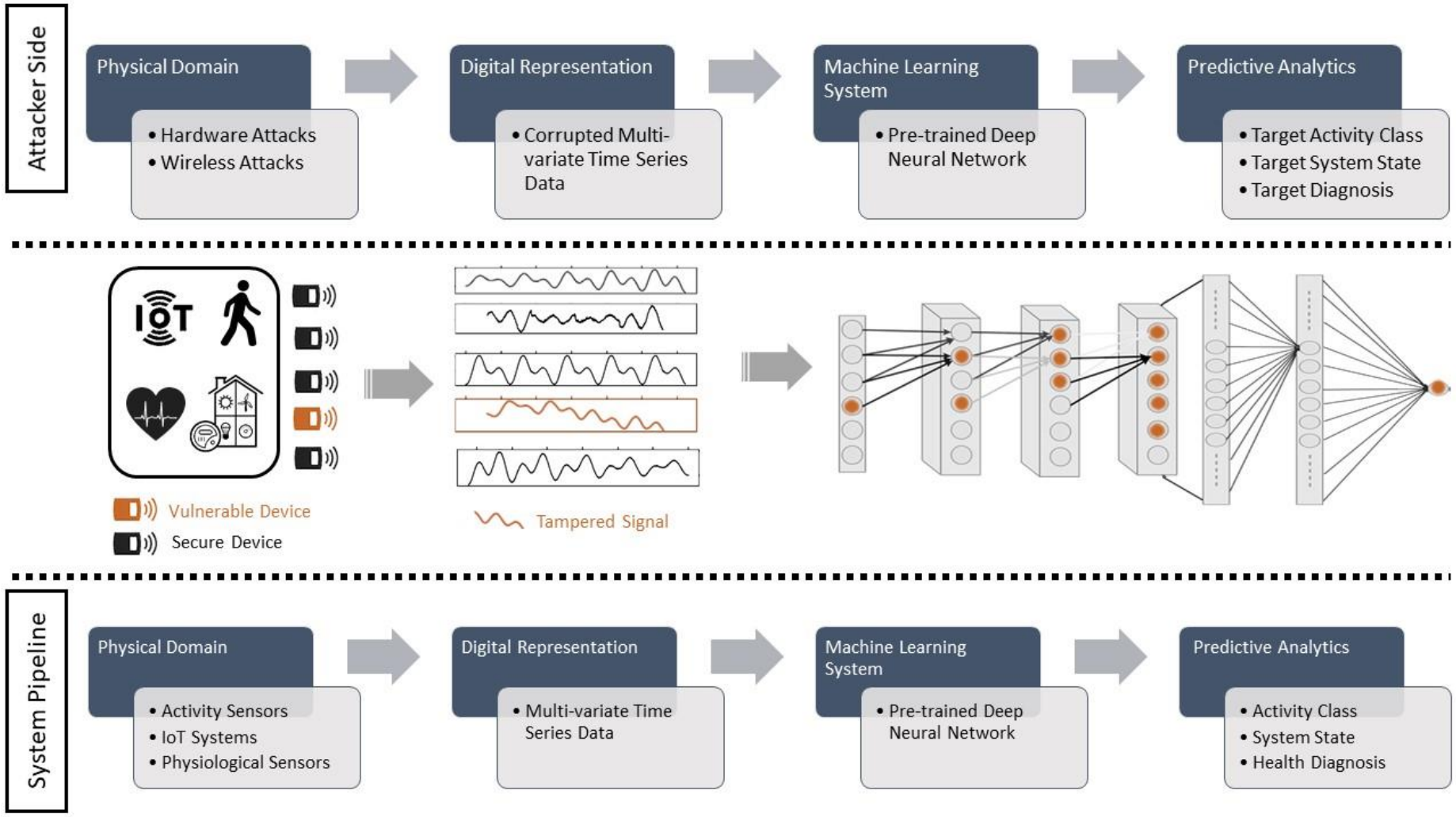
[3]

Black-box vs White box



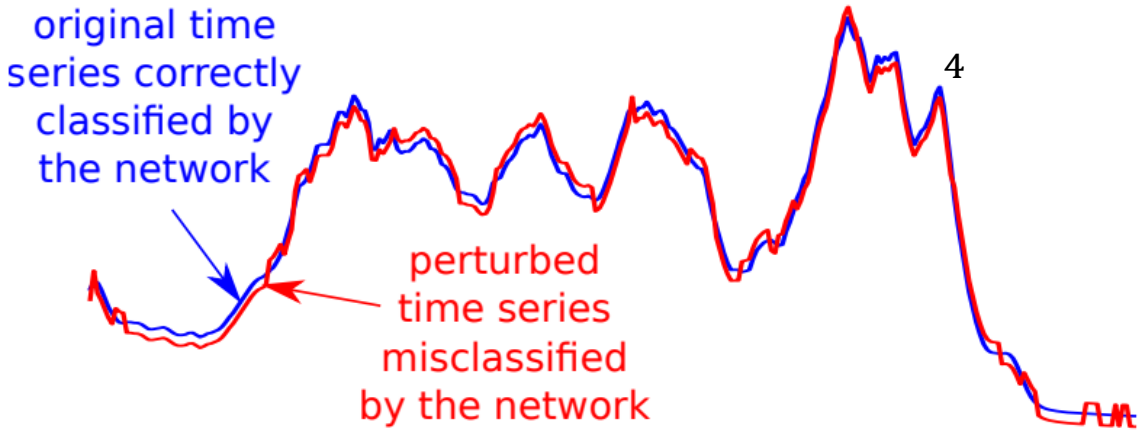
[3]:Moosavi–Dezfooli, et al. "Universal adversarial perturbations." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

Vulnerability of Time-series Deep Learning Systems



Inefficiency of Existing Algorithms

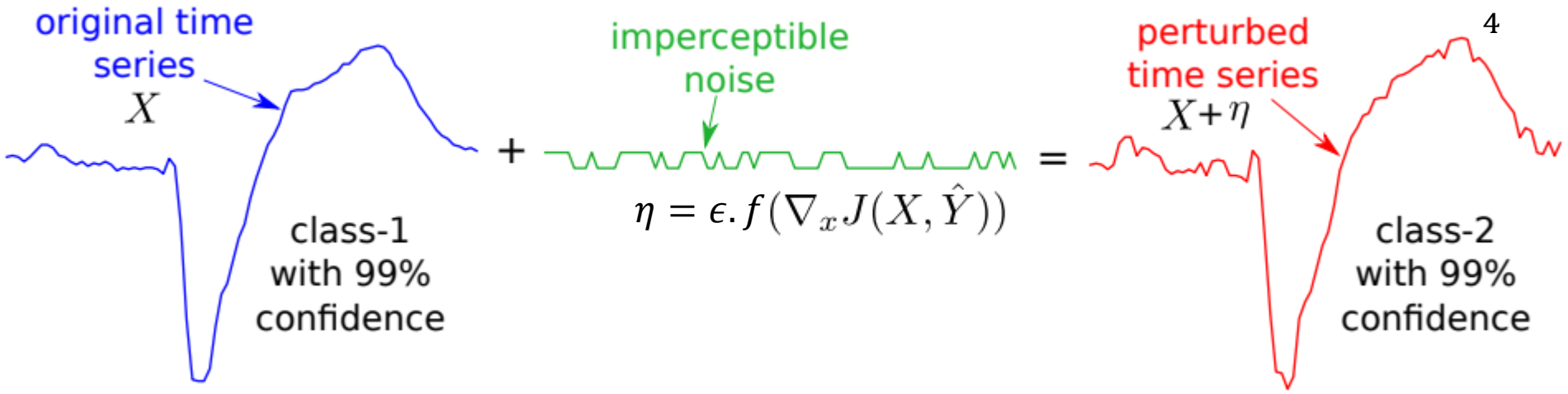
- Characteristics of time-series (e.g., fast-pace oscillations, sharp peaks) are different from images
- Most existing adversarial algorithms are not efficient or not applicable to time-series data.



4:Fawaz, Hassan Ismail, et al. "Adversarial attacks on deep neural networks for time series classification." 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019.

Inefficiency of Existing Algorithms

- Standard adversarial algorithms that are transferrable to time-series domain



- Imperceptibility via l_p norms

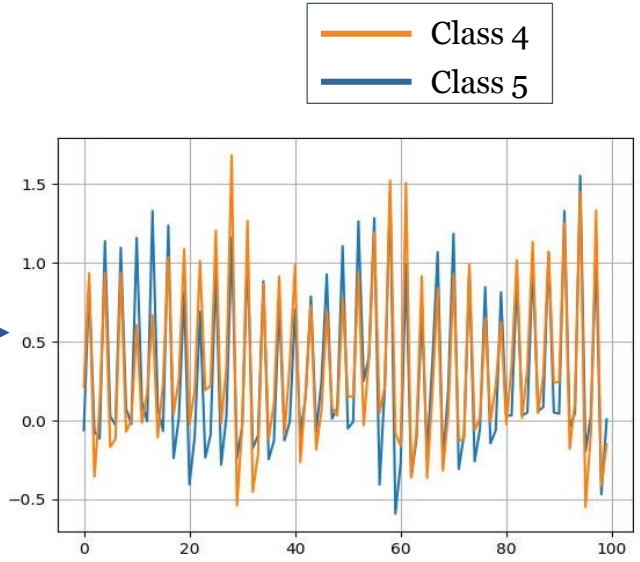
4:Fawaz, Hassan Ismail, et al. "Adversarial attacks on deep neural networks for time series classification." 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019.

Limitations of standard l_p norms

- Standard adversarial algorithm rely on l_p-norm to constraint adversarial attacks
- Case Study: WISDM dataset

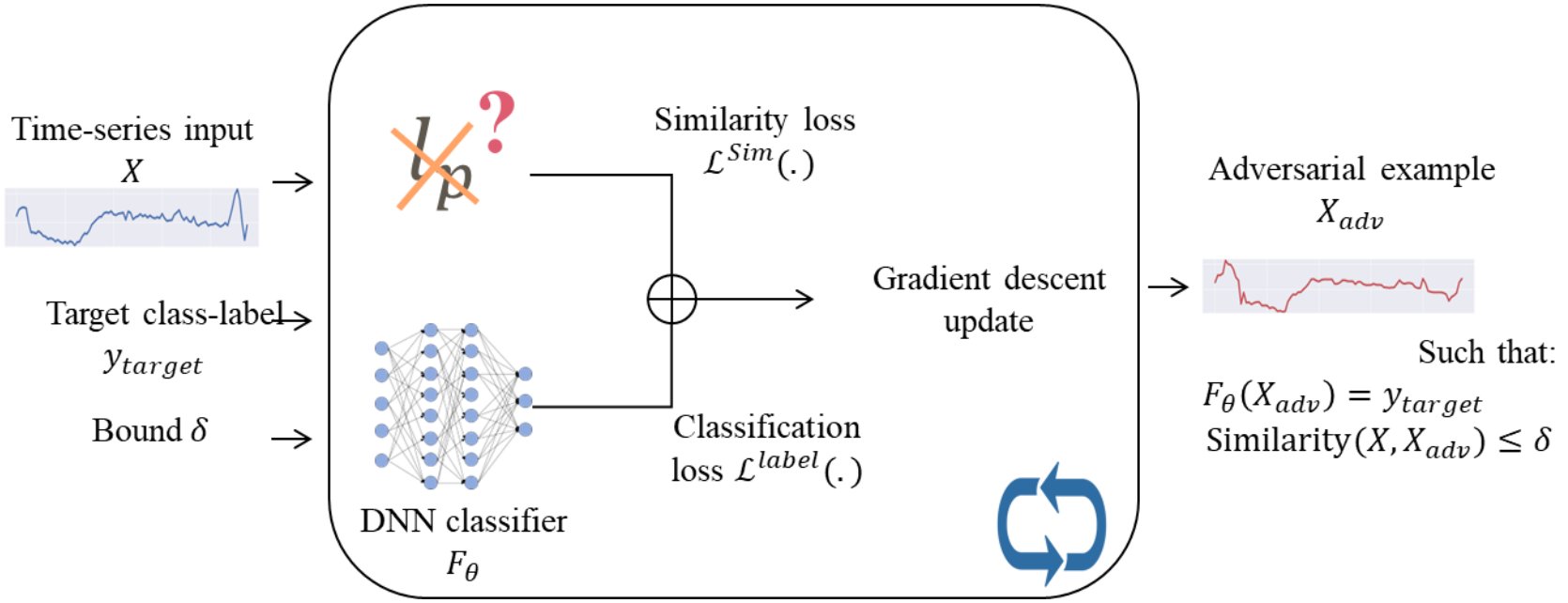
Class 2	0.015				
Class 3	0.009	0.014			
Class 4	0.011	0.012	0.011		
Class 5	0.009	0.014	0.008	0.008	
Class 6	0.007	0.012	0.007	0.008	0.003
	Class 1	Class 2	Class 3	Class 4	Class 5

Minimum normalized l_2 distance between examples from different classes



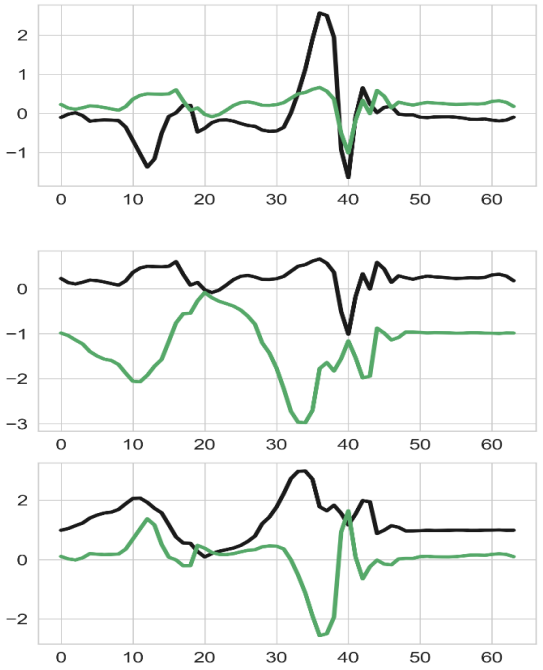
→ Perturbations based on Euclidean distance can result in adversarial time-series signals which semantically belong to a different class-label.

Novel Frameworks for Adversarial Time-Series Data



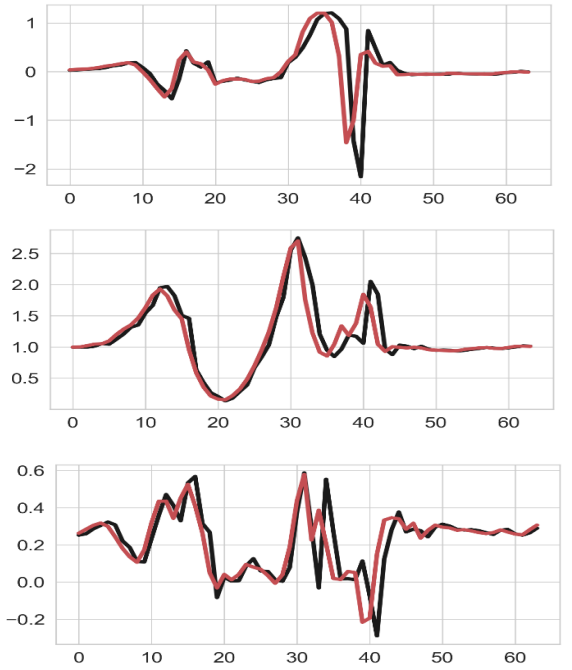
→ How to address the similarity loss?

Similarity Challenges



Approach 1:

- Extract statistical features that better represents the semantics of each class



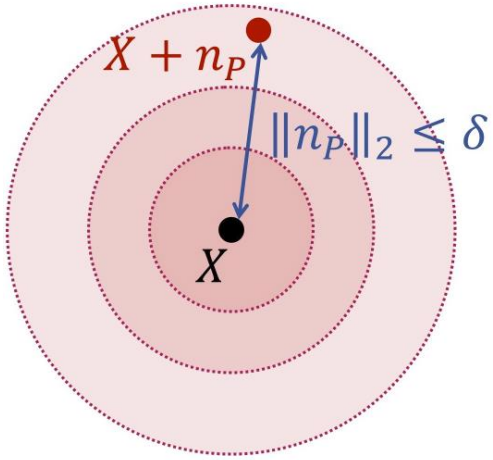
Approach 2:

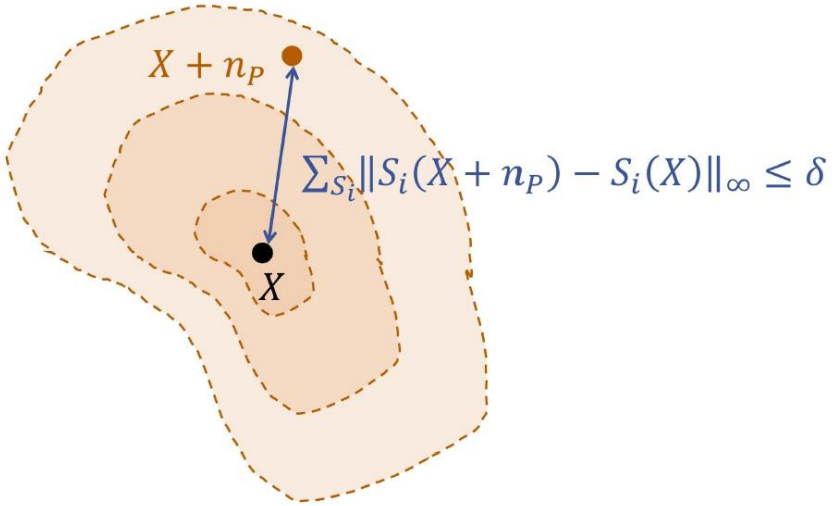
- Improve over l_p-norm-based distance by considering the temporal shifts

Approach 1: Adversarial Framework with Certified Robustness for Time-Series Domain via Statistical Features

- We investigate statistical features as similarity features

- Time-series data are comprehensible using multiple statistical tools
  - Mean, Standard Deviation, Skewness…





Adversarial transformation hypothesis

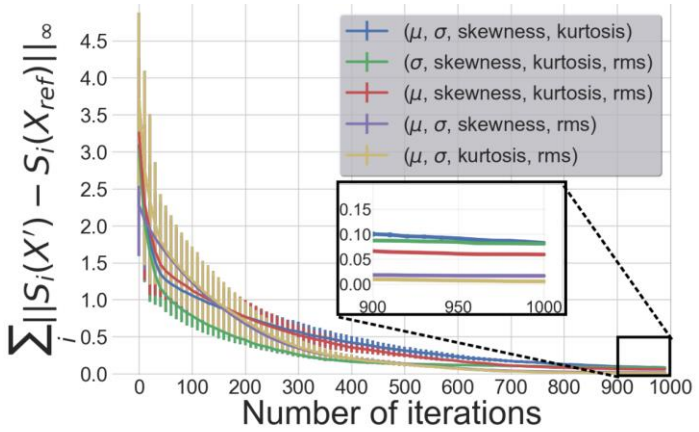
- Polynomial transformation: $X_{adv} = PT(X)$ vs. $X_{adv} = X + \delta$
  - Inspired by power series, we approximate adversarial transformation using a polynomial representation with a chosen degree d: $PT(X) = \sum_{k=0}^{d} a_k X^k + O(X^{d+1})$
  - Reduces search complexity by generalizing to a universal perturbation

Theorem. Using the statistical space, a larger set of possible adversarial attacks can be generated by polynomial transformations than standard additive perturbations.

$$\left\{ X_{adv} = \mathcal{PT}(X),\ \forall a_k \right\} \supsetneq \left\{ X_{adv} = X + \delta,\ \forall \delta \right\}$$

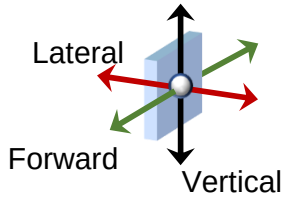
Which features to choose?

- Domain-agnostic

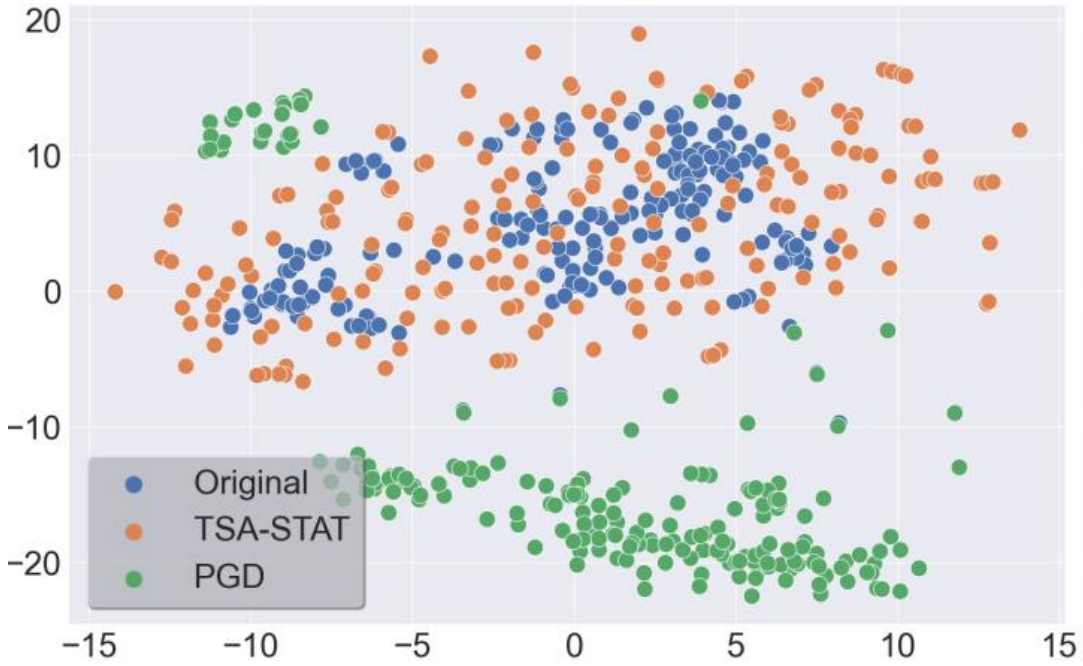


Convergence of the statistical loss using different statistical constraint sets

- Domain-specific
  - Accelerometers: Use of body acceleration computed from all accelerometer axes.



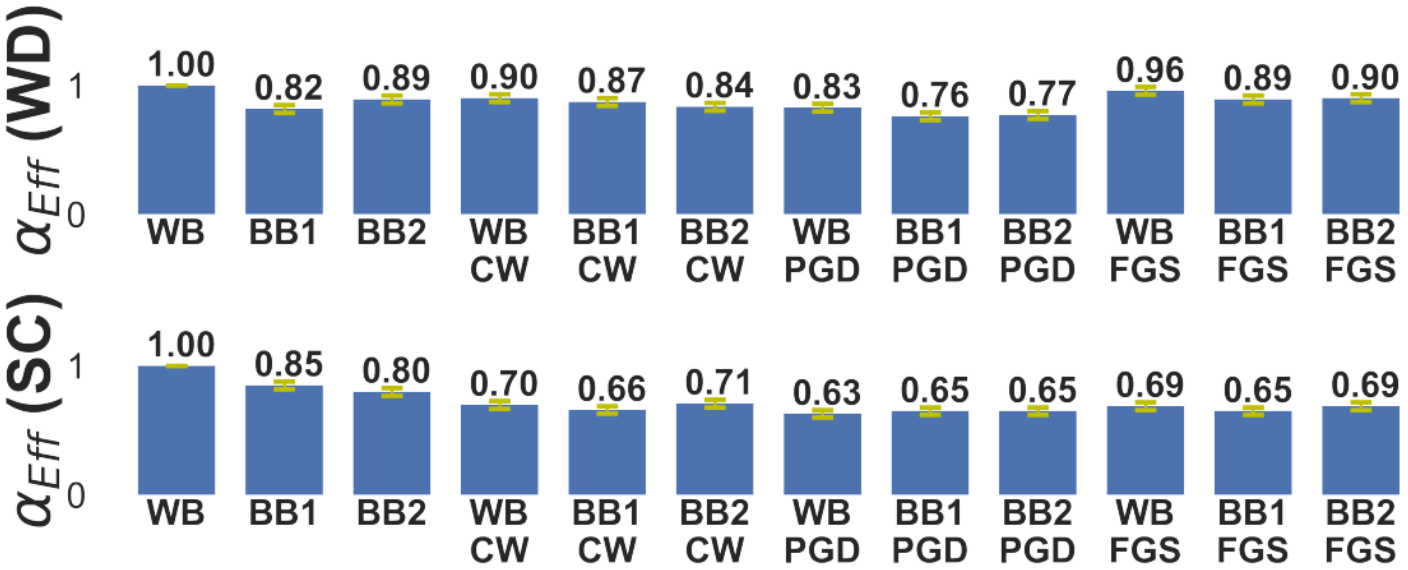
Qualitative assessment for TSA-STAT



t-Distributed Stochastic Neighbor Embedding showing the distribution of natural and adversarial examples from TSA-STAT and PGD.

General results: Attack performance

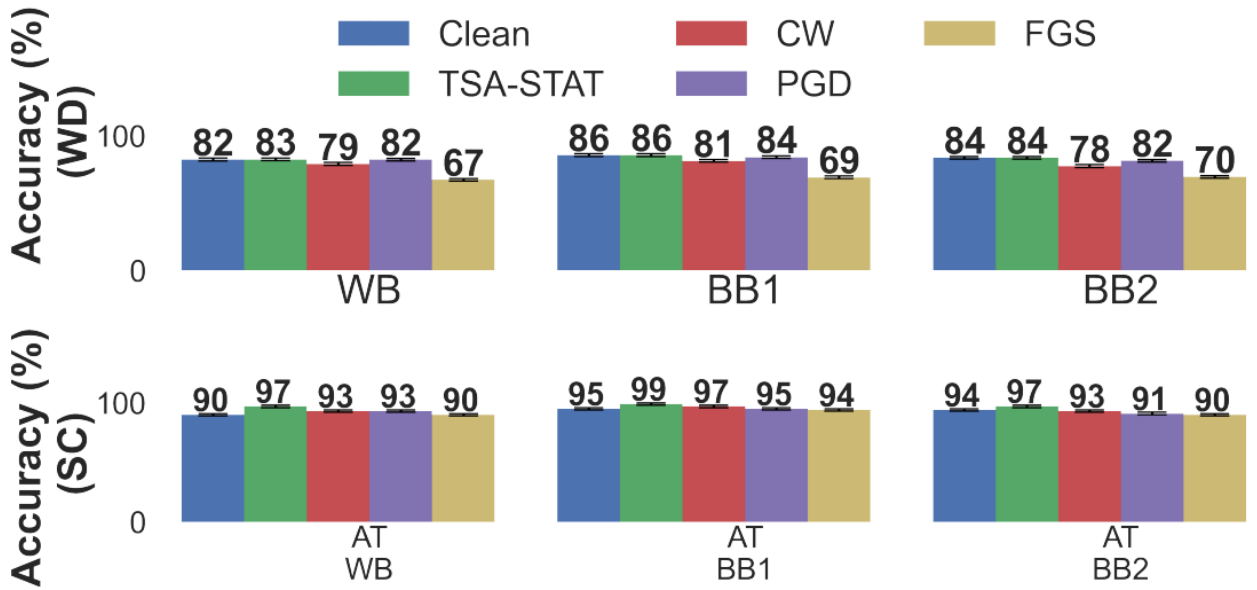
$$\alpha_{Eff} = \frac{\#\text{ successful adversarial examples}}{\text{Total \# of adversarial examples}}$$



Results of different adversarial algorithms attack performance under white-box and black-box settings on different deep models.

General results: Robustness performance

- Adversarial Training: Data augmentation technique using adversarial examples from training data to train the classifier



Results for adversarial training using adversarial examples from different adversarial algorithms for different deep models.

Adversarial Certification

- We derive theoretically guaranteed upper bounds for disturbances
- The bound ensures that classifier will provide *reliable classifications* if disturbances are within the bounds
- The certification δ provides the following guarantee for a given X and F_θ:

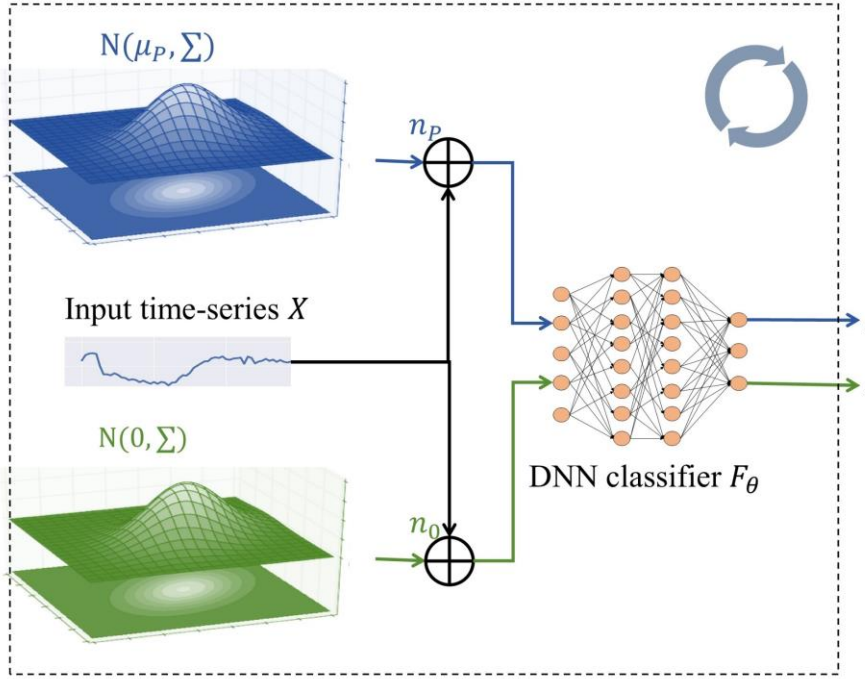
$$\forall\, \epsilon \leq \delta: \quad F_\theta(X+\epsilon) = F_\theta(X)$$

- Our results rely on

$$D_\alpha(\mathcal{N}(\mu_1, \sum_1)\|\mathcal{N}(\mu_2, \sum_2)) = \frac{\alpha}{2}(\mu_1-\mu_2)^T \sum_\alpha (\mu_1-\mu_2) - \frac{1}{2(\alpha-1)}\, ln \frac{|\sum_\alpha|}{|\sum_1|^{1-\alpha}|\sum_2|^\alpha}$$

$$, \textit{ where } \sum_\alpha = \alpha\sum_1 + (1-\alpha)\sum_2.$$

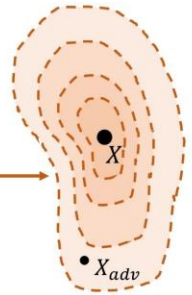
Adversarial Certification Algorithm



$$EP = \{p_j\}_{j\in Y}$$
$$E0 = \{p_j^0\}_{j\in Y}$$

If $\underset{j\in Y}{\operatorname{argmax}}\, p_j \neq \underset{j\in Y}{\operatorname{argmax}}\, p_j^0$ then Certification declined

Else compute bound δ



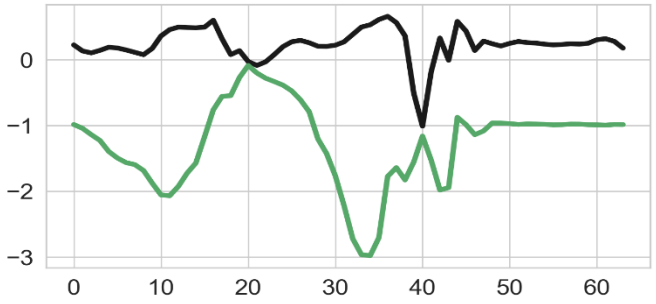
Robust region

$$\|\mu(X_{adv} = X + n_P) - \mu(X) \in \mathbb{R}^n\|_\infty \leq \delta$$

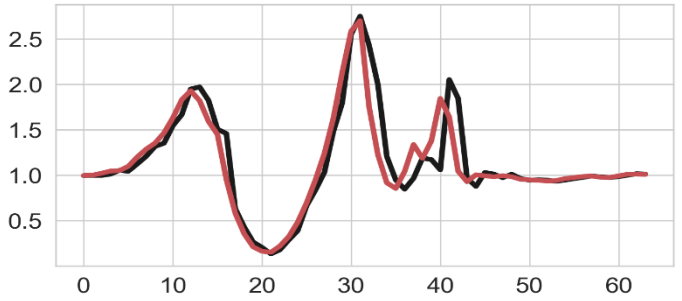
$$\|\mu_P\|_\infty^2 \leq \max_{\alpha\neq 1} \frac{2}{\alpha \cdot \sum^{(S)}} \cdot \left(-ln\left(1 - p_{(1)} - p_{(2)} + 2\left(\frac{1}{2}\left(p_{(1)}^{1-\alpha} + p_{(2)}^{1-\alpha}\right)\right)^{\frac{1}{1-\alpha}}\right)\right)$$

Lemma. If a certified bound has been generated for the mean of input time-series signal and classifier, then certified bounds for other statistical/temporal features can be derived consequently.

Similarity challenges for time-series



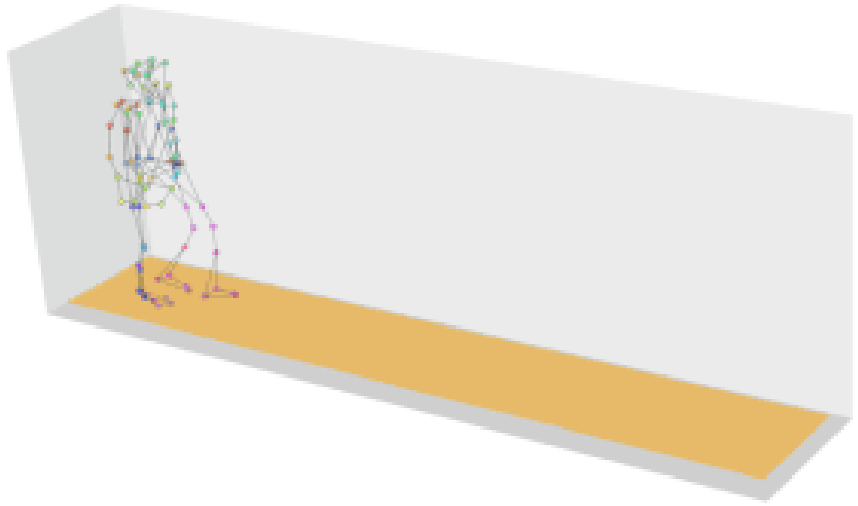
TSA-STAT can capture successfully this similarity case



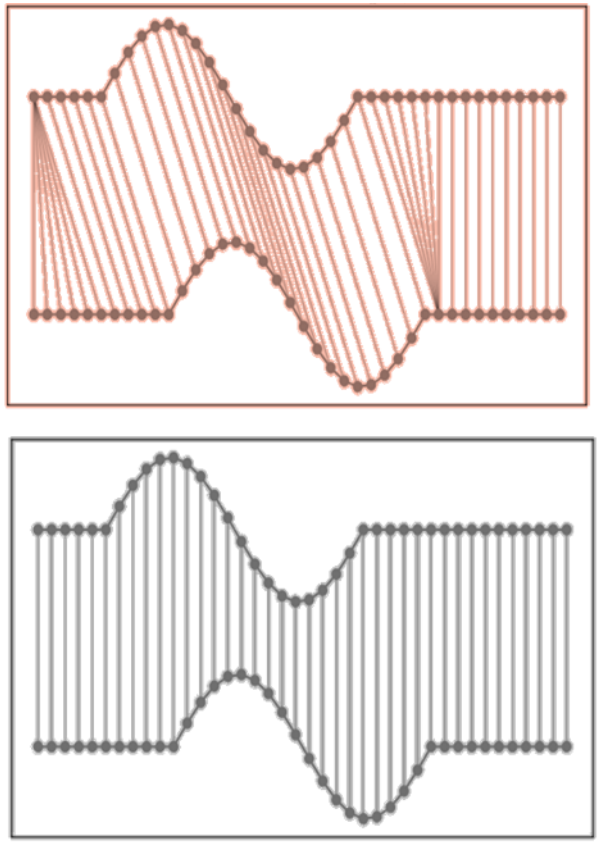
TSA-STAT is too complex for this perturbation case

Approach 2: Dynamic Time Warping based Adversarial Framework for Time-Series Domain

- We investigate a similarity measure that accounts for shifts along temporal axis, scaling and frequency change

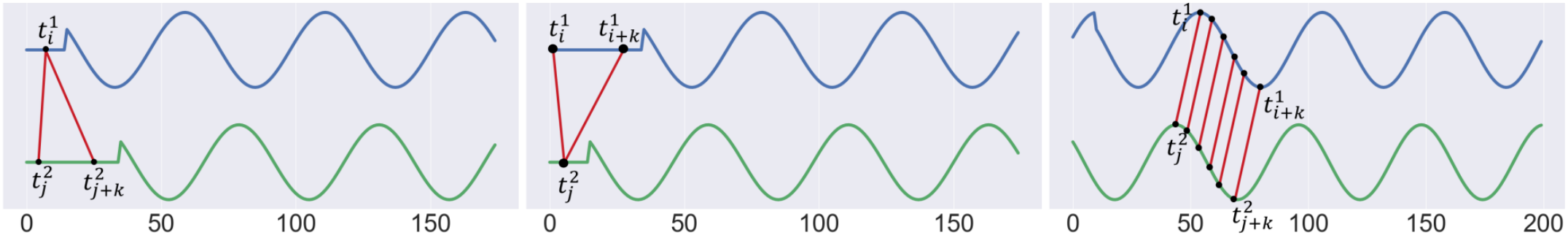


Two repetitions of the same walking sequence were recorded using a motion-capture system.[5]



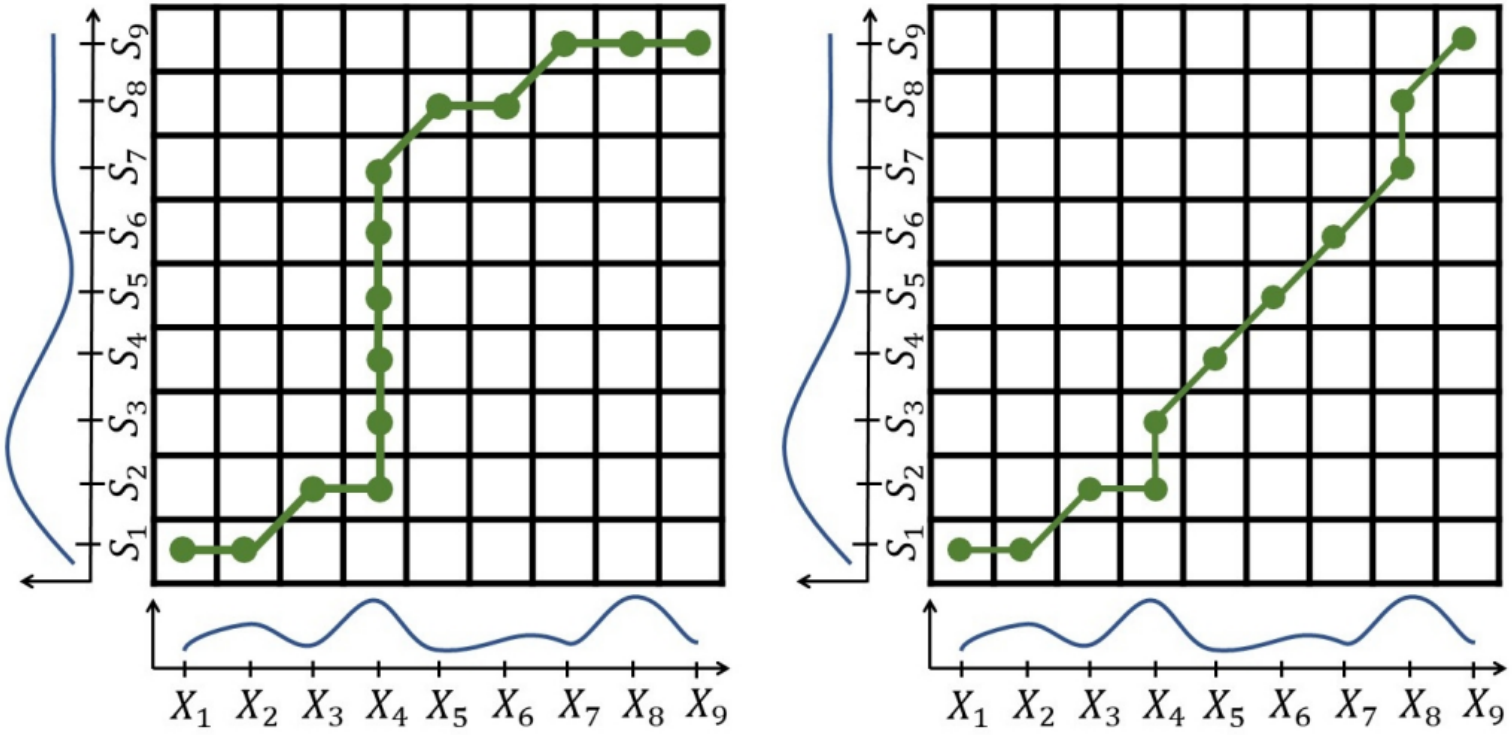
5: Olsen, Niels Lundtorp *et al*. "Simultaneous inference for misaligned multivariate functional data." Journal of the Royal Statistical Society Series C: Applied Statistics 67.5 (2018): 1147-1176.

Dynamic Time Warping



- We investigate Dynamic Time Warping approach for adversarial attacks.
- Dynamic Time Warping seeks for the optimal temporal alignment.
- A temporal alignment is a matching between time indexes t_i of the two time-series t^1 and t^2.

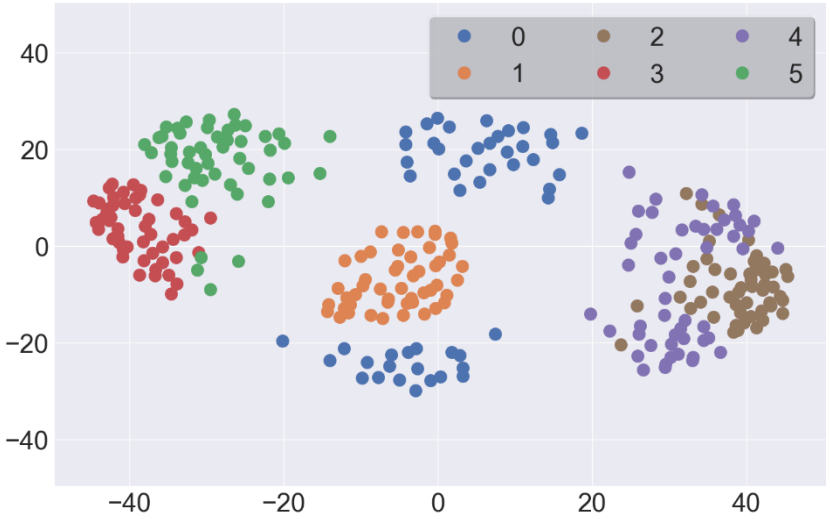
Dynamic Programming for DTW



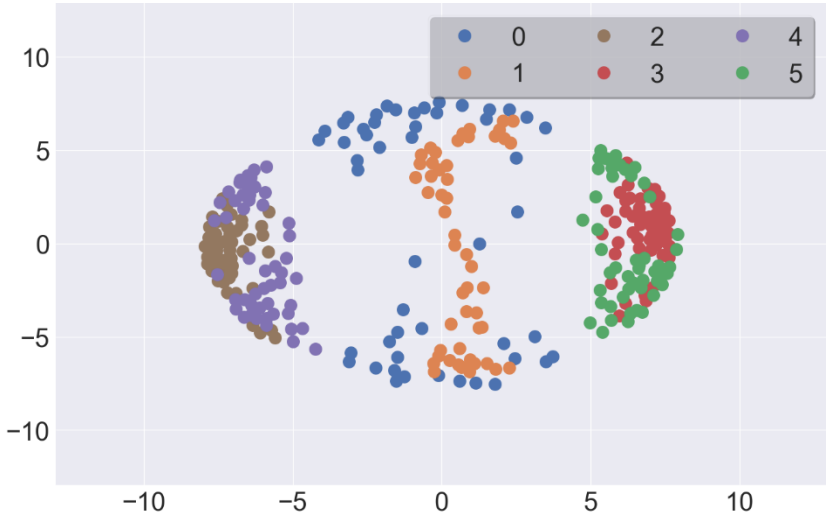
- $DTW(X,S) = \min_{\pi_k} \|X_i - S_i\|$
- Iterative – Quadratic complexity – Slow!!

DTW vs. Euclidean Space: Similarity Comparison

DTW Space



Euclidean Space



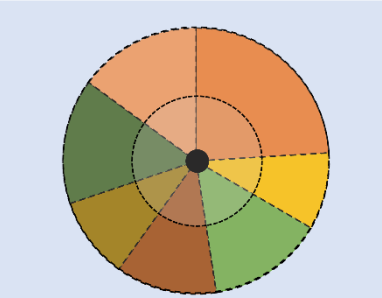
t-Distributed Stochastic Neighbor Embedding showing the empirical class distribution of real-world data examples and their similarity using DTW measure and Euclidean distance.

- DTW space exhibits better clustering for same-class data than Euclidean space

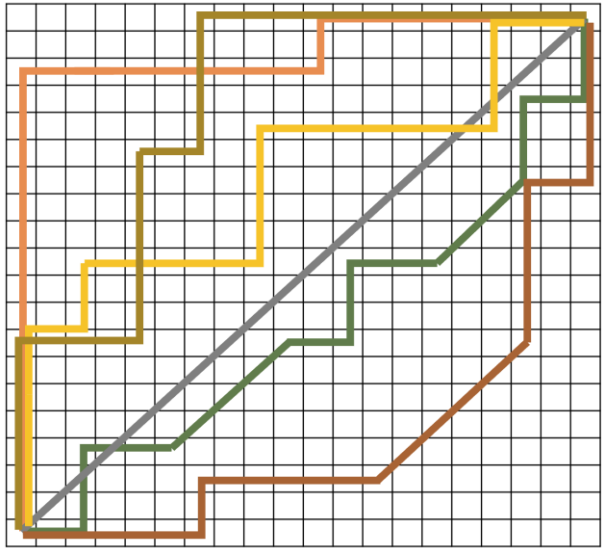
→ How to overcome the challenges of implementing DTW in the adversarial setting?

Naïve DTW Approach

- At each iteration:
  1. Compute the optimal DTW alignment
  2. Use the alignment in the gradient and create an adversarial example



DTW space



- High computational cost
- Single adversarial example

Key Insight: Random Alignment Path Approximation

- We formalize the following theorem to devise an effective and efficient algorithm

Theorem. For a time-series X and a random alignment path P, the resulting adversarial example from DTW-AR is equivalent to using standard DTW computation (tight approximation).

Key Insight: Random Alignment Path Approximation

Algorithm 1 DTW-AR based Adversarial Algorithm

Input: time-series X; DNN classifier F_θ; target class-label y_{target}; learning rate η; maximum iterations MAX

Output: adversarial example X_{adv}

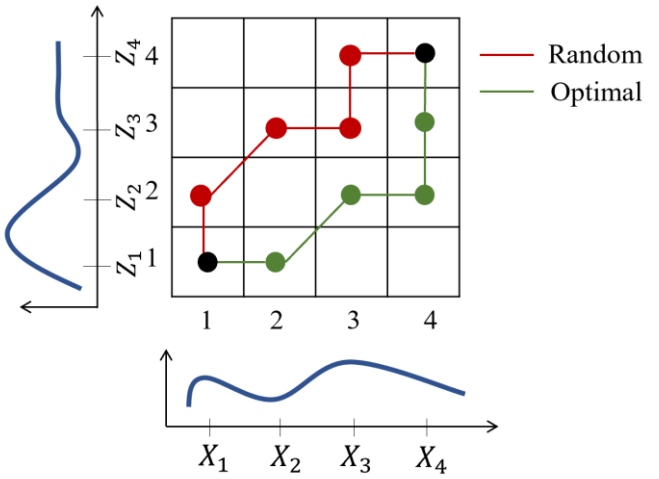
1: $P_{rand} \leftarrow$ random alignment path
2: Initialization: $X_{adv} \leftarrow X$
3: **for** i=1 to MAX **do**
4: $\quad \mathcal{L}(X_{adv}) \leftarrow \mathcal{L}^{label}(X_{adv}) + \mathcal{L}^{DTW}(X_{adv}, P_{rand})$
5: $\quad$ Compute gradient $\nabla_{X_{adv}} \mathcal{L}(X_{adv})$
6: $\quad$ Perform gradient descent step:
$\quad X_{adv} \leftarrow X_{adv} - \eta \times \nabla_{X_{adv}} \mathcal{L}(X_{adv})$
7: **end for**
8: **return** optimized adversarial example X_{adv}

Theorem. For a time-series X and a random alignment path P, the resulting adversarial example from DTW-AR is equivalent to using standard DTW computation (tight approximation).

How?

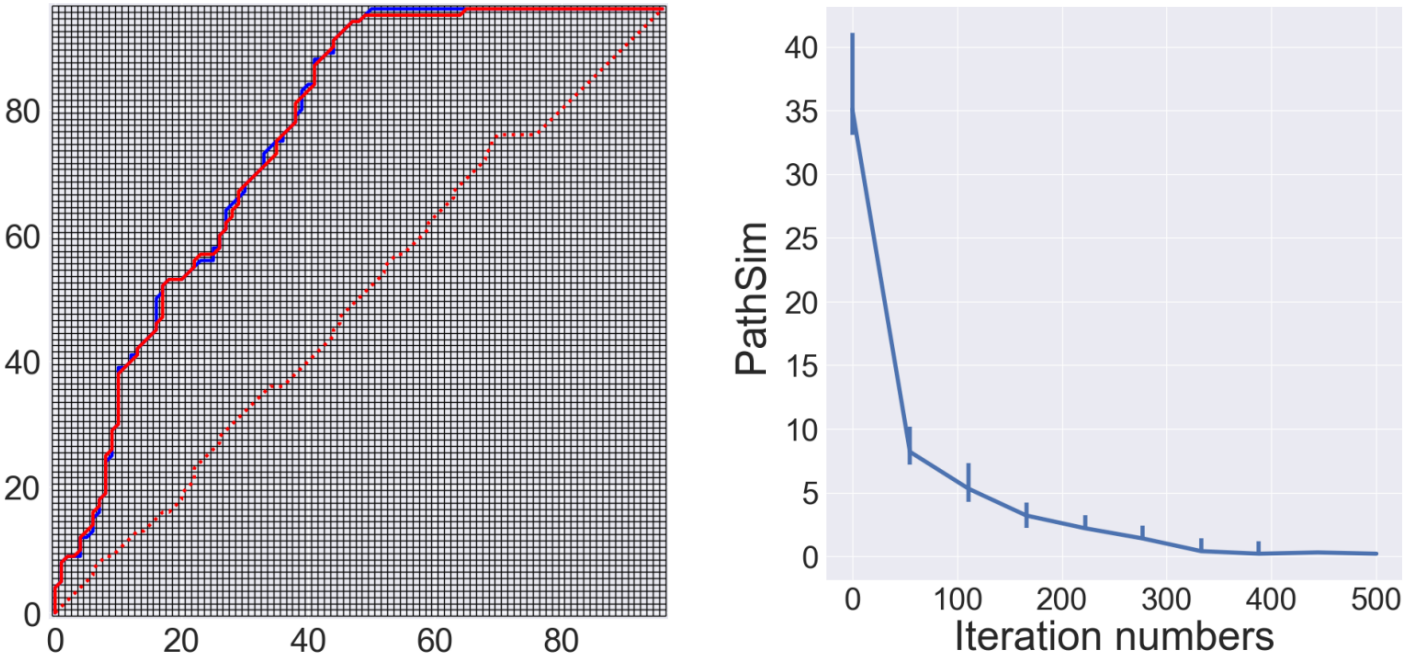
Path-specific distance optimization

- We define a new metric *PathSim* as a metric between two alignment paths *P1* and *P2* in the DTW cost matrix that satisfies the distance axioms
- We define $P_i = \{c_1^i, \dots, c_{len(P_i)}^i\}$



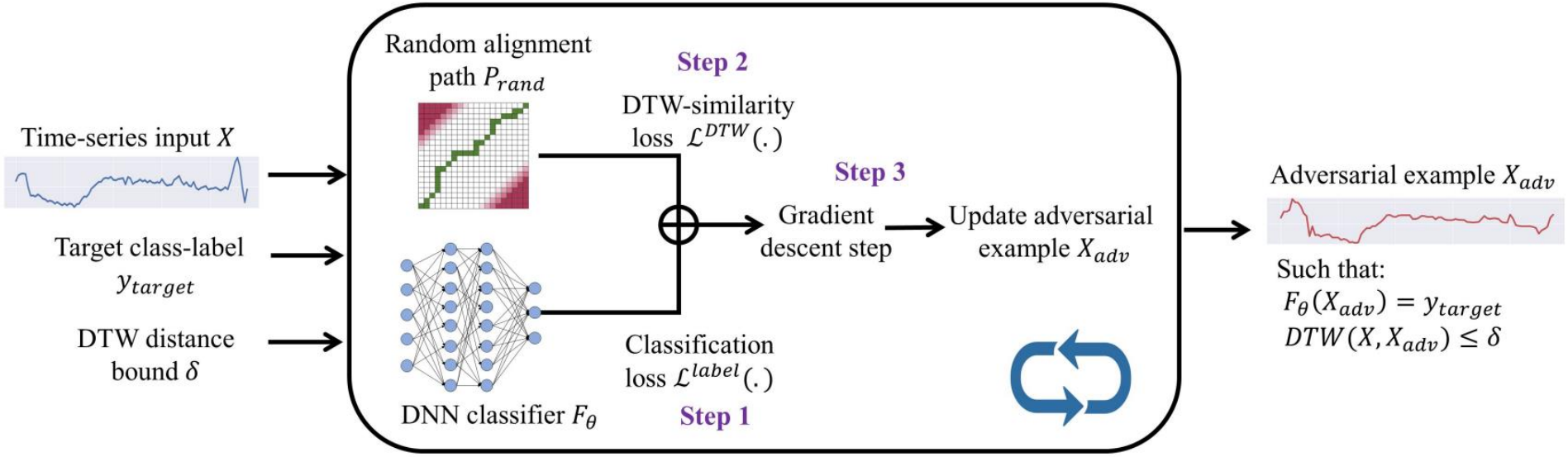
$$\texttt{PathSim}\,(P_1, P_2) = \frac{1}{2T}\left(\sum_{c_i^1}\min_{c_j^2}\|c_i^1 - c_j^2\|_1 + \sum_{c_i^2}\min_{c_j^1}\|c_i^2 - c_j^1\|_1\right)$$

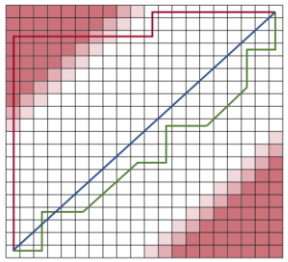
Path-specific distance optimization

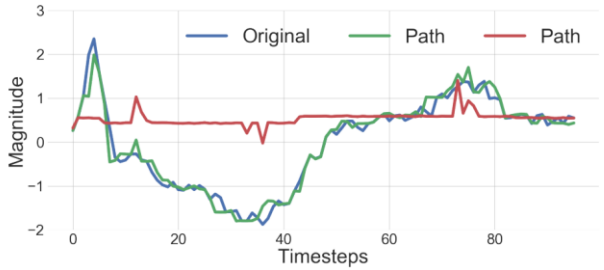


Example of the empirical convergence of the optimal alignment path between the optimized example and the original example at the start of the algorithm (dotted red path) and at the end (red path) to the given random alignment path (blue path).

DTW-AR Framework



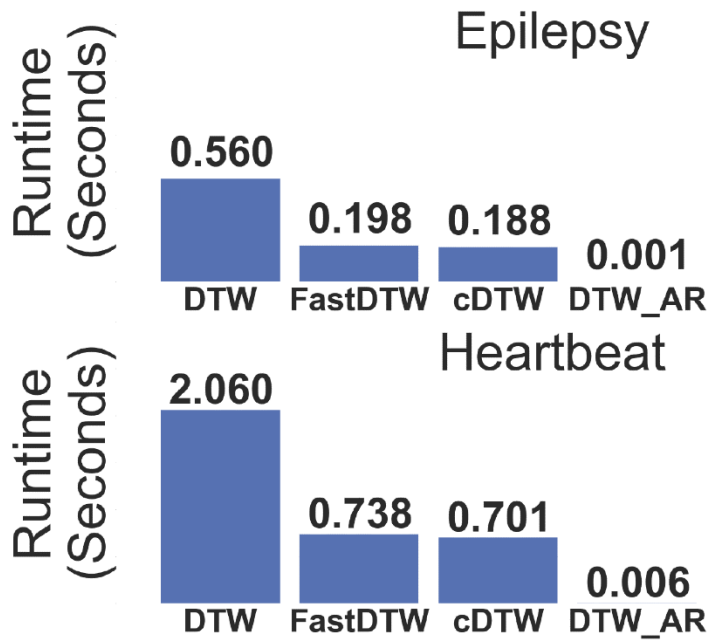




- The choice of the alignment range is important for better example generations

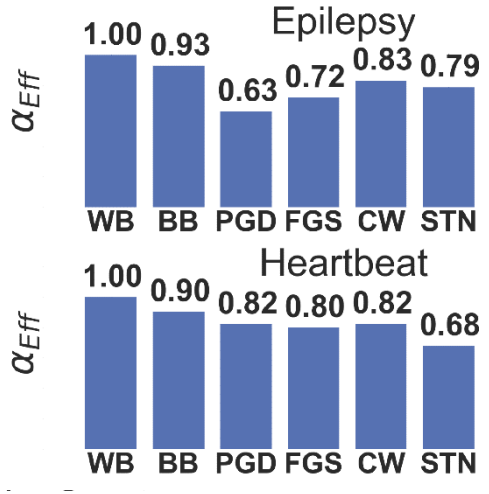
DTW-AR Results: Computational Efficiency

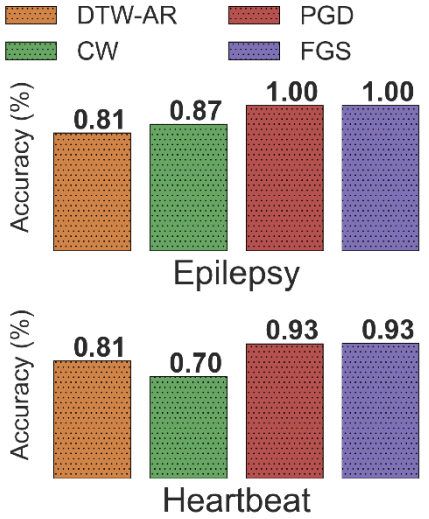
- Overall computational cost is significantly reduced using DTW-AR



Average runtime per iteration

DTW-AR Results: Effectiveness of Adversarial Examples





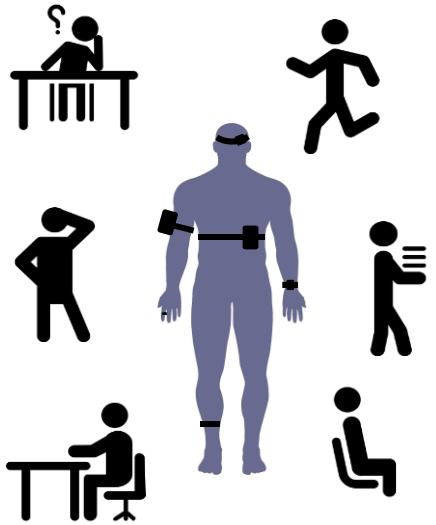
- DTW-AR is capable of fooling DNNs w/o existing adversarial training.

- DTW-AR is effective in predicting the original label of adversarial examples with high accuracy.

Application of adversarial frameworks in wearable sensors-enabled ML applications

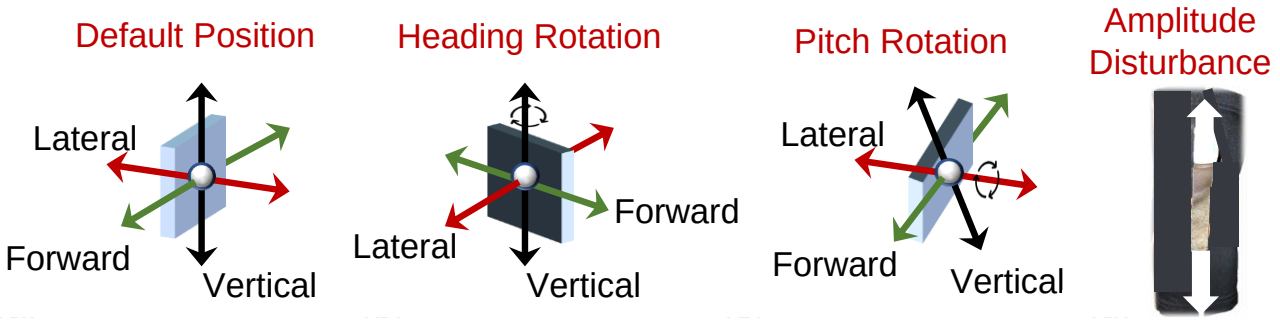
Human Activity Recognition (HAR) Application

- HAR identifies activities, such as walking, sitting, driving, jogging
- HAR can provide valuable insight to health specialists
- Applications of HAR
  - Patient rehabilitation
  - Fall detection
  - Physical activity promotion



(Natural) Sensor Disturbances in HAR

- We investigate four major classes of sensor disturbances that can affect data recording



  - Heading rotation
    - Changes sensor values in forward and lateral directions
  - Pitch rotation
    - Changes sensor values in forward and vertical directions
  - Amplitude and sensor hardware disturbances

Statistical Optimization Framework

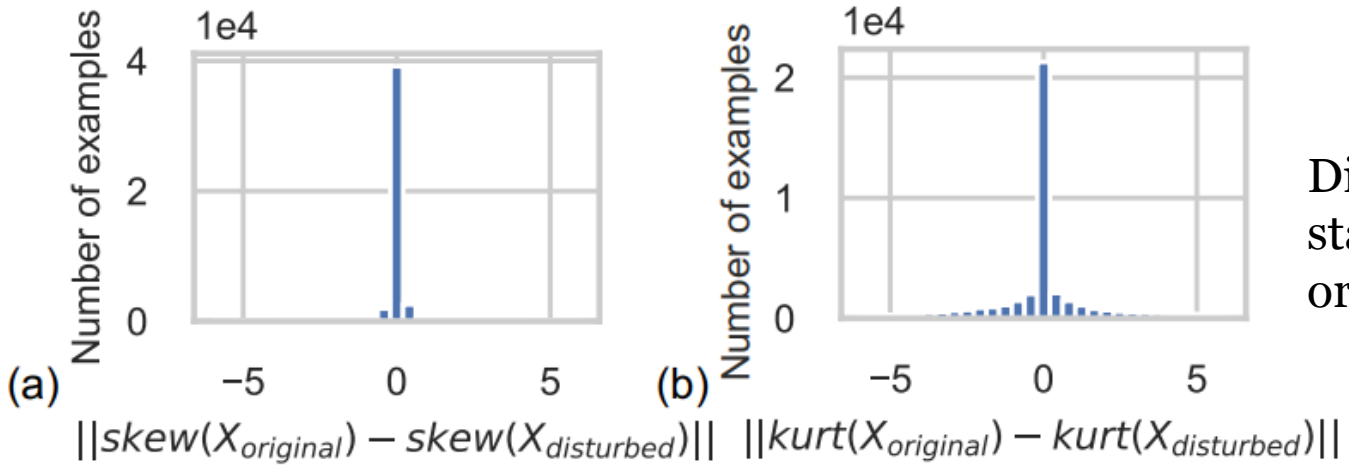
- We propose StatOpt, an instantiation of the general statistical optimization framework TSA-STAT for ML-based HAR on wearable systems.
- The goal of StatOpt is to create new training examples to capture the overall structure of natural sensor disturbances that may occur in sensor observations and use them to improve the classifier reliability.
- It is challenging to define the disturbance explicitly
- We employ statistical features S_i to generate training examples

Statistical Optimization Framework

➢Create adversarial examples such that $\left|\left|S_i(X) - S_i(X_{adv})\right|\right| < \epsilon$

1. Body acceleration
   - Total acceleration across all three directions of motion
   - Generally, body acceleration is same even if sensor is disturbed
2. Skewness
   - Measures the symmetry of the distribution
3. Kurtosis
   - Measures the distribution tail of the input values

Accuracy of StatOpt Data Generation



Distribution of the difference between the statistical features of the disturbed and the original data.

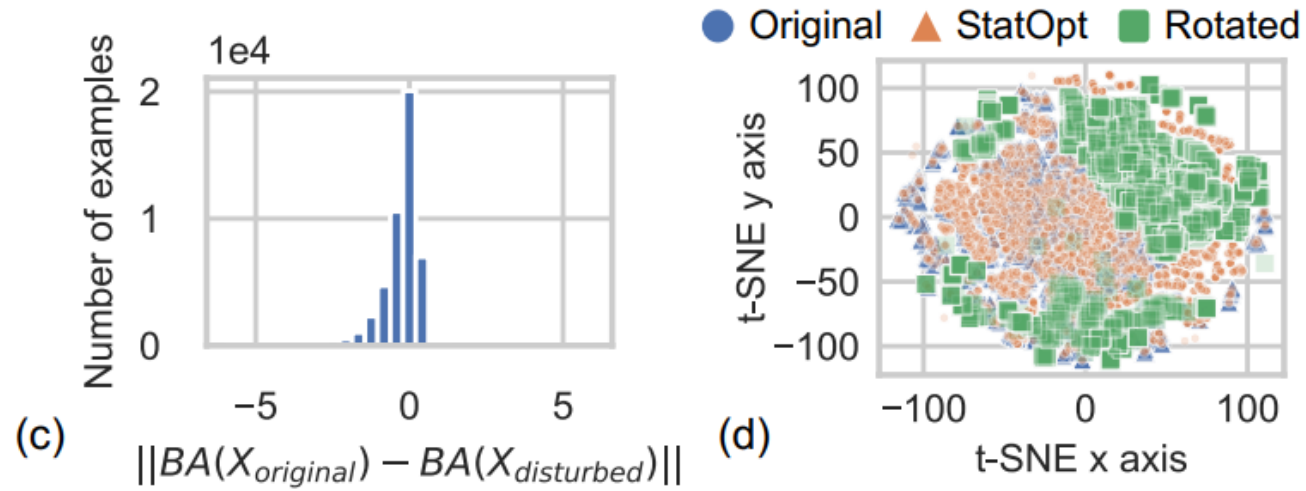
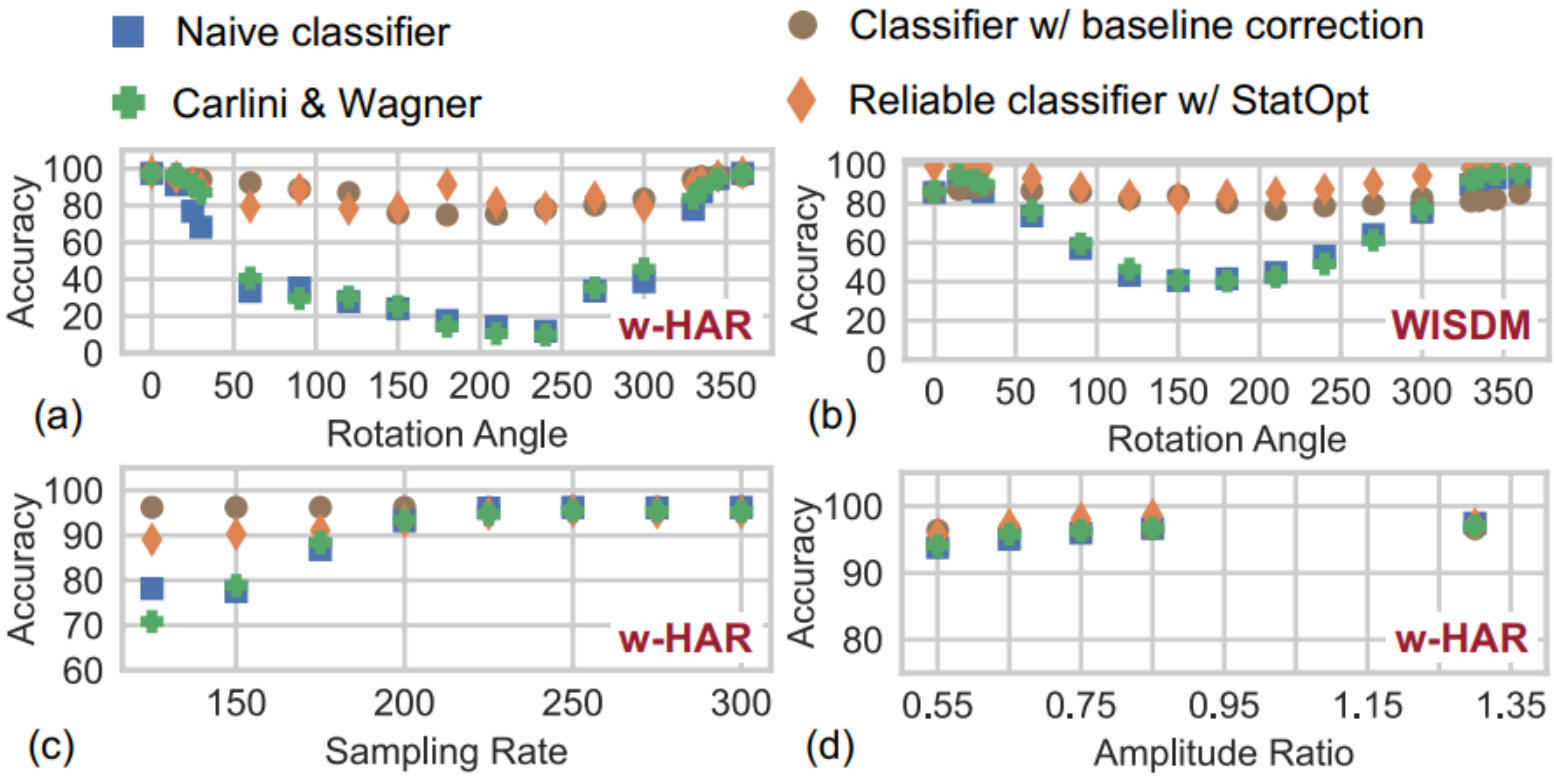


Illustration of the t-SNE for the observed sensor data, examples generated by StatOpt, and data with sensor disturbances.

Classification Performance Comparison



Accuracy comparison between the standard classifier, baseline, and StatOpt-enabled reliable classifier

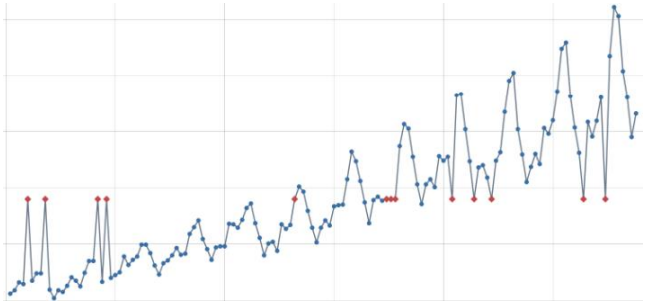
Implementation Overhead

- We characterize the implementation overhead on TI-CC2650
- StatOpt has no overhead for data recovery

		Baseline		StatOpt	
Disturbance	Block	Exe. Time (ms)	Energy (mJ)	Exe. Time (ms)	Energy (mJ)
All (1×/activity)	Classifier	85.60	0.94	85.60	0.94
Heading (1×/session)	Walk	3000.00	34.68	-	-
	PCA	24.93	0.29	-	-
Pitch (1×/session)	Stand	3000.00	34.71	-	-
	Gravity detect	1.90	0.02	-	-
All (1×/activity)	Resampling	46.08	0.52	-	-
	Correction	2.31	0.03	-	-

Missing Data in HAR

- Random Missing data
  - Isolated missing samples not clustered around any particular time instance.
  - Occurs due to limited communication bandwidth and buffer overflow in a sensor.

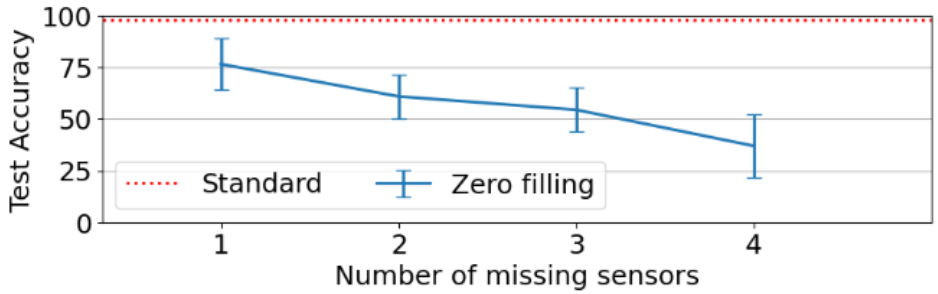


- Block Missing data
  - A sequence of samples are missing
  - Occurs when one or more sensors have to go into a low-power state

Challenges of ML Algorithm for Missing Data

- Using zero inputs for missing data, we get sensor data as

$$\tilde{X}_{\{j\}} = \begin{cases} 0^T, & \text{if} \quad i \in \{j\} \\ X_i, & \text{if} \quad i \notin \{j\} \end{cases}$$



- Generative imputation networks recover raw data for classification
  - However, some applications do not need exact data recovery
  - Moreover, generative networks incur high overhead
- Trade-off between accuracy & overhead
- Maximize the accuracy without imputing the data exactly
  - Reduce memory overhead of imputation by avoiding generative networks

Accuracy-Preserving Imputation (AIM)

- AIM is based on two key insights
  - Exact data recovery is not needed if the application accuracy is preserved
  - Can train the ML model to be robust to small deviations from the exact data
- AIM obtains a single imputation pattern for each missing data scenario
  - We push the classifier to predict the correct labels from the available channels
- Write the imputation pattern as

$$\mathcal{I}_{\{j\}} = \begin{cases} \mathcal{I}_i, & \text{if} \quad i \in \{j\} \\ X_i, & \text{if} \quad i \notin \{j\} \end{cases} \qquad \text{and } F_\theta(X) \sim F_\theta(\mathcal{I}_{\{j\}})$$

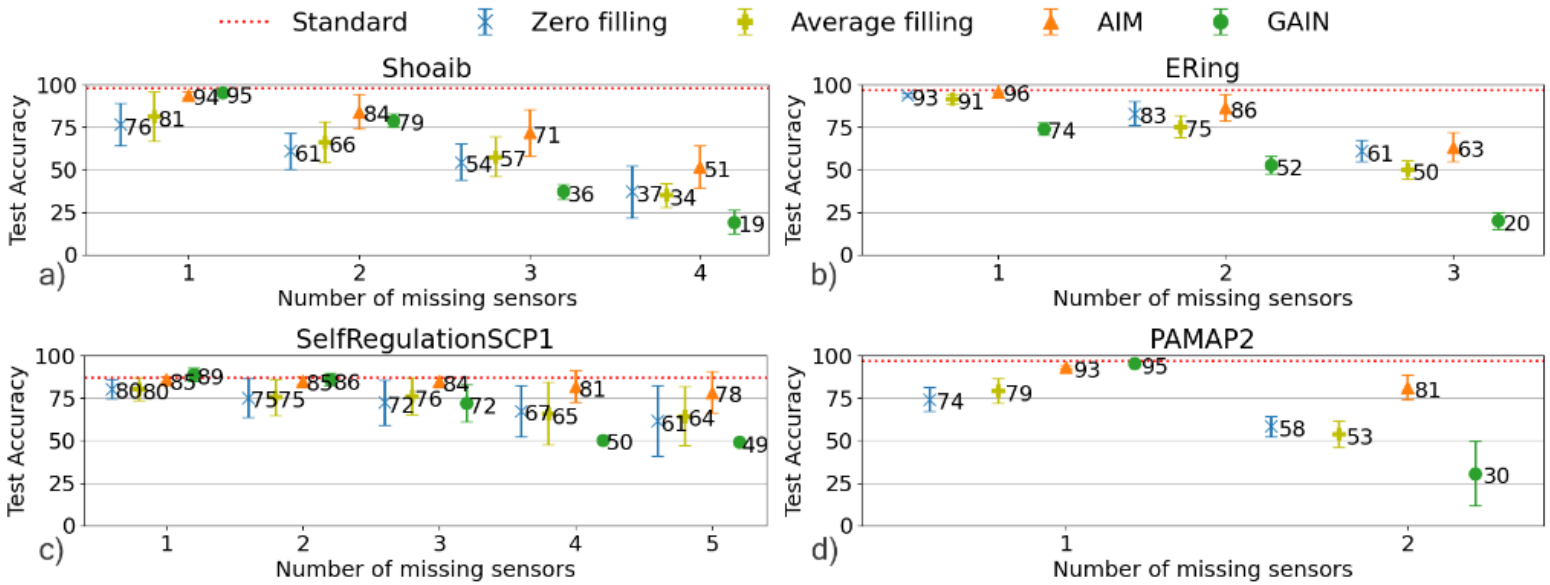
Imputation Pattern Search

- Given $\{j\}$: We find $\mathcal{I}_{\{j\}} s.t. \forall X, F_\theta(X) \approx F_\theta\left(\mathcal{I}_{\{j\}}\right)$
- Search objective

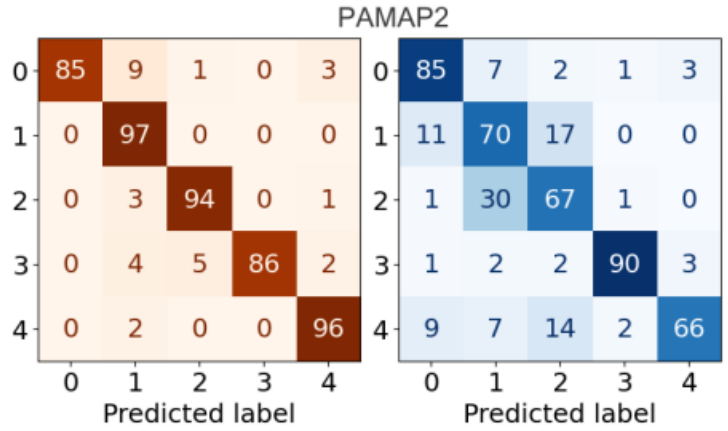
$$\min_{\mathcal{I}_{j\in\{j\}}} MSE\left(Logits\left(F_\theta(X)\right), Logits\left(F_\theta\left(\mathcal{I}_{\{j\}}\right)\right)\right)$$

- Minimize objective to find imputation patterns
- Similar performance of F_θ on original input X (no missingness) and input with imputation $\mathcal{I}_{\{j\}}$

Classification Accuracy with AIM



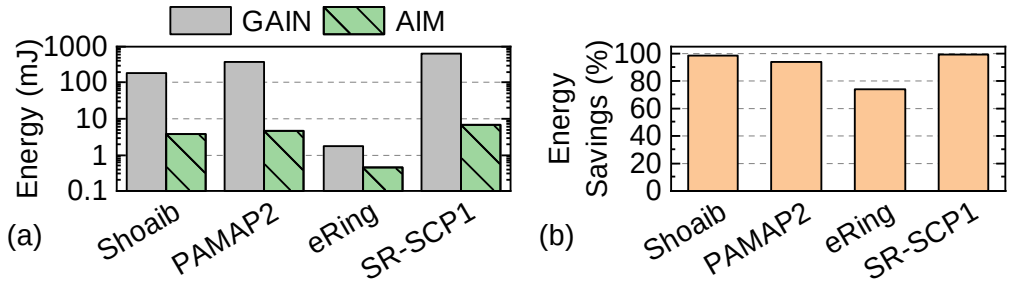
Accuracy (Mean and standard deviation) of the robust-trained ML classifier via different imputation methods on all combinations of missing sensors.



Normalized confusion matrix using AIM (red) imputation and zero-filling (blue).

Implementation Overhead

- Key advantage of AIM is lower energy and memory overhead
- AIM consumes less than 10 mJ per imputation
- Energy savings are close to 98% for all datasets except eRing
  - eRing has lower energy savings about 74% (has lower computation requirements for both GAIN and AIM)
- AIM can improve the battery life of wearable health monitoring devices by an order of magnitude



Dataset	AIM Memory (MB)	GAIN Memory (MB)
Shoaib	0.180	25
PAMAP2	0.055	60
eRing	0.007	0.19
SR-SCP1	0.667	81

Training Robust Deep Models for Time-Series Domain

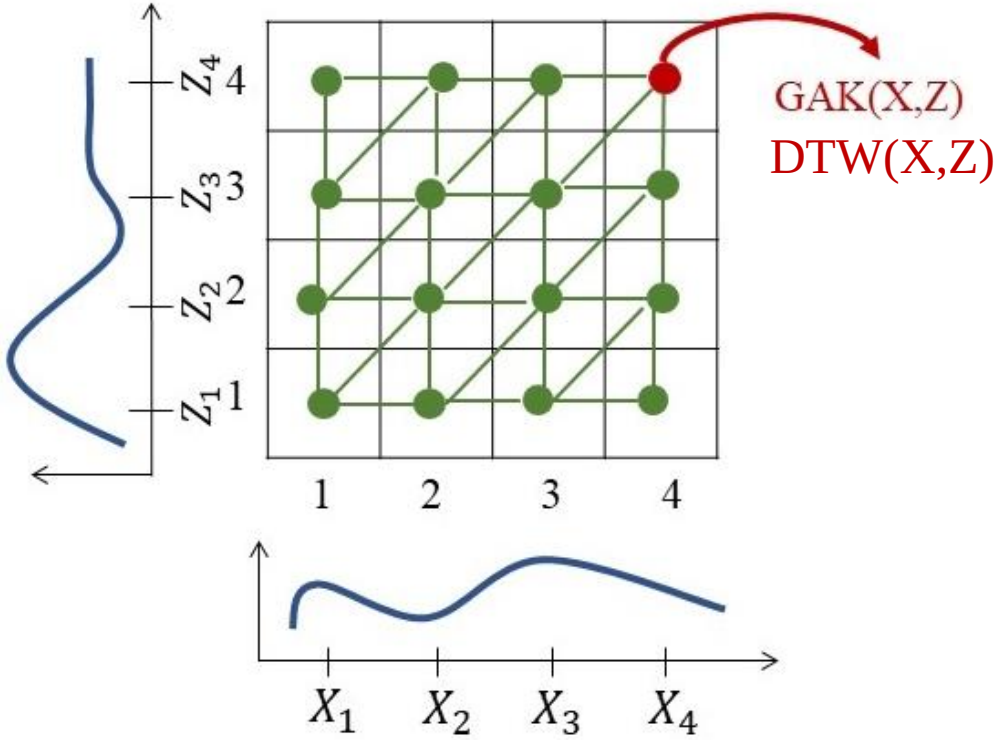
Training Robust Deep Models for Time-Series Domain

- Adversarial training employs augmented data such as adversarial examples and input perturbations.
  - The most successful empirical defense
  - Requires a significant amount of augmented data per example
- Training via explicit loss function to capture the robustness criteria and optimize it.
  - Formal formulation of adversarial training

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \max_{\epsilon} l(F_\theta(X+\epsilon), y) \text{ such that } ||\epsilon|| \leq \delta$$

Robust Training via Elastic Measures

- Instead of a single **optimal** alignment between x and x', we desire a measure that takes all possible alignments $\pi \in A$ into consideration



$$k_{\mathrm{GAK}}(x, x') = \sum_{\pi \in \mathcal{A}} \exp\left(- \frac{d_\pi(x, x')}{\nu} \right)$$

$$d_\pi(x, x') = \sum_{i=1}^{|\pi|} \mathrm{dist}(x_{\pi_1(i)}, x'_{\pi_2(i)})$$

Proposed approach: Explicit training for robustness

- A principled framework referred as RObust Training for Time-Series (RO-TS) to create robust DNNs for time-series data.

The inner maximization problem serves the role of an attacker whose goal is to find adversarial examples that achieves the largest loss.

$$\min_{w \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \max_{a_i} \ell(f(x_i + a_i, w), y_i)$$
$$\text{s.t. } d(x_i, x_i + a_i) \leq \varepsilon$$

The outer minimization problem serves the role of a defender whose goal is to find the optimal parameters of the deep model

Optimization Challenges

- GAK gradient estimation is computationally expensive
- Randomly sample a constant number of alignments $\pi \in A$ at each iteration to estimate the gradient
- Sampling paths leads to biased gradient estimate → SGDA does not hold

Method: SCAGDA Optimization Algorithm

- Novel stochastic compositional alternating gradient descent ascent algorithm
- Solves a family of nonconvex-nonconcave min-max compositional problems

$$\min_{w} \max_{a_i} \frac{1}{n} \sum_{i=1}^{n} \phi_i(w, a_i) := f_i(w, a_i) - g\left(\frac{1}{m} \sum_{j=1}^{m} h_{i,j}(a_i)\right)$$

$$\min_{w \in \Theta} \max_{a_i} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i + a_i, w), y_i) + \lambda \log\left(k_{\mathrm{GAK}}(x_i, x_i + a_i)\right)$$

- No previous analysis on min-max optimization with compositional structure

RO-TS Instantiation of SCAGDA

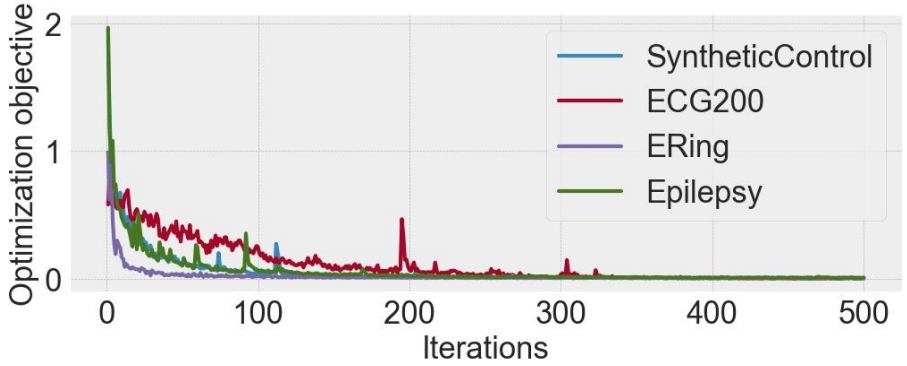
$$\min_{w \in \Theta} \max_{a_i} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i + a_i, w), y_i) + \lambda \log \left(k_{\mathrm{GAK}}(x_i, x_i + a_i) \right)$$

∇_a for gradient ascent

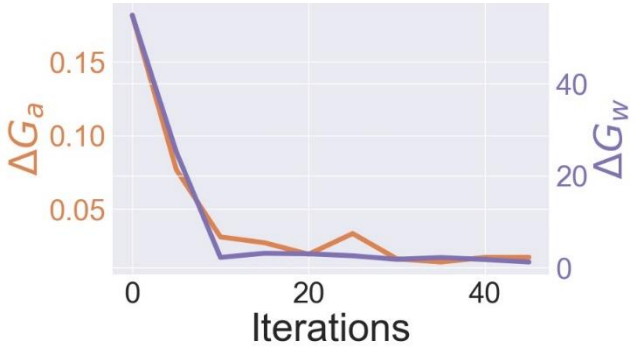
∇_w for gradient descent

- Do not require all alignments for GAK estimation
  - ✓ Efficient training
  - ✓ Higher scalability

RO-TS Theoretical Results



Empirical convergence of ROTS algorithm.

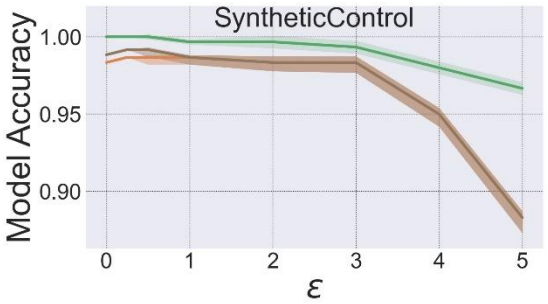


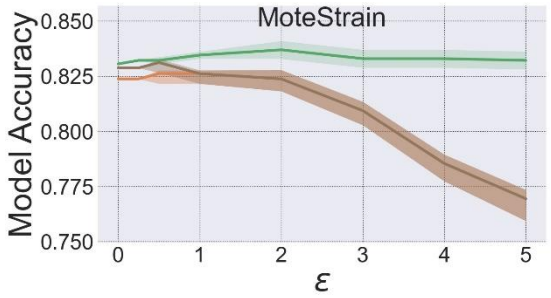
The accuracy gap in the gradients over weights G_w and over perturbations G_a using 5% vs. all of the alignments

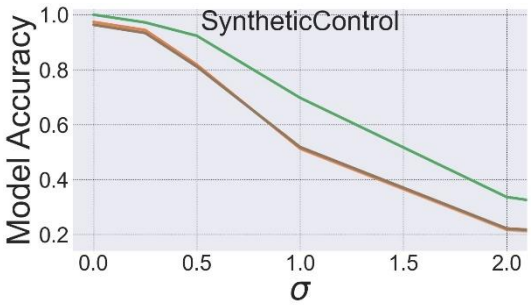
Comparison of the computational runtime between both settings

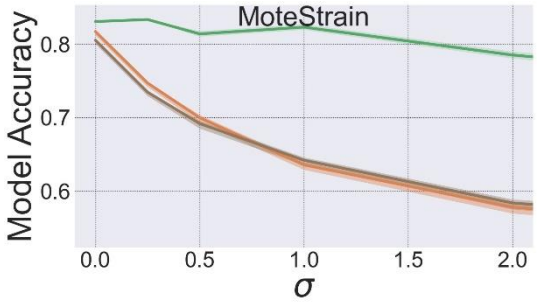
RO-TS Training Effectiveness – Setting #1

RO-TS vs. Adversarial training









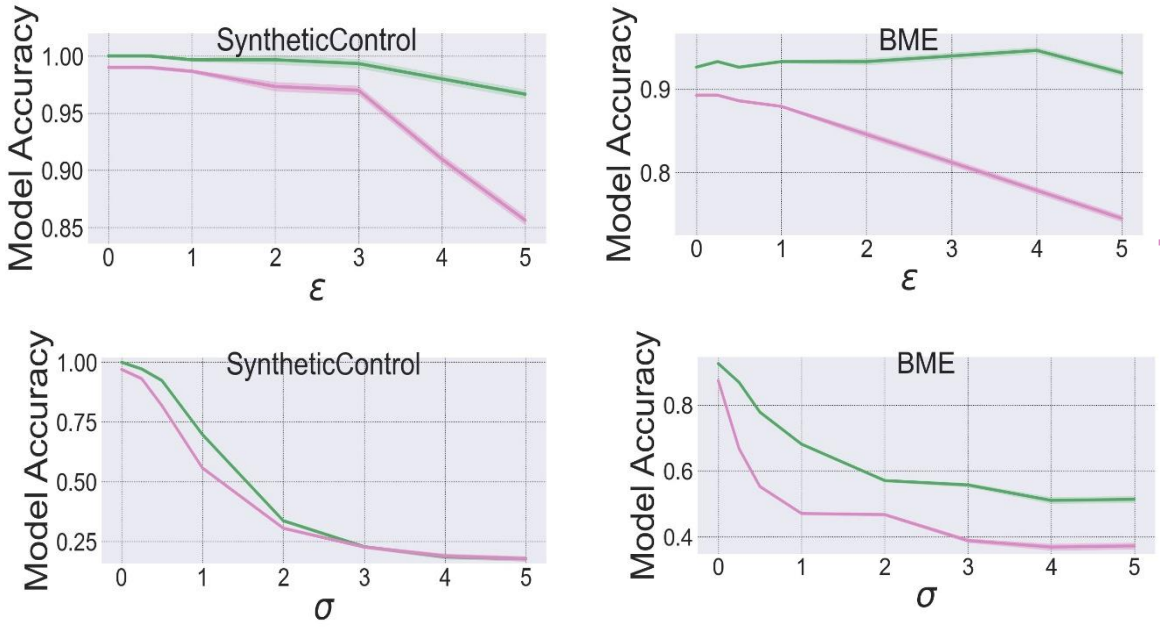
Comparison of

- RO-TS algorithm
- Fast Gradient Sign
- Projected Gradient Descent

using Gaussian perturbation σ and adversarial perturbation ε on input

RO-TS Training Effectiveness – Setting #2

RO-TS: GAK vs. l_2



Comparison of

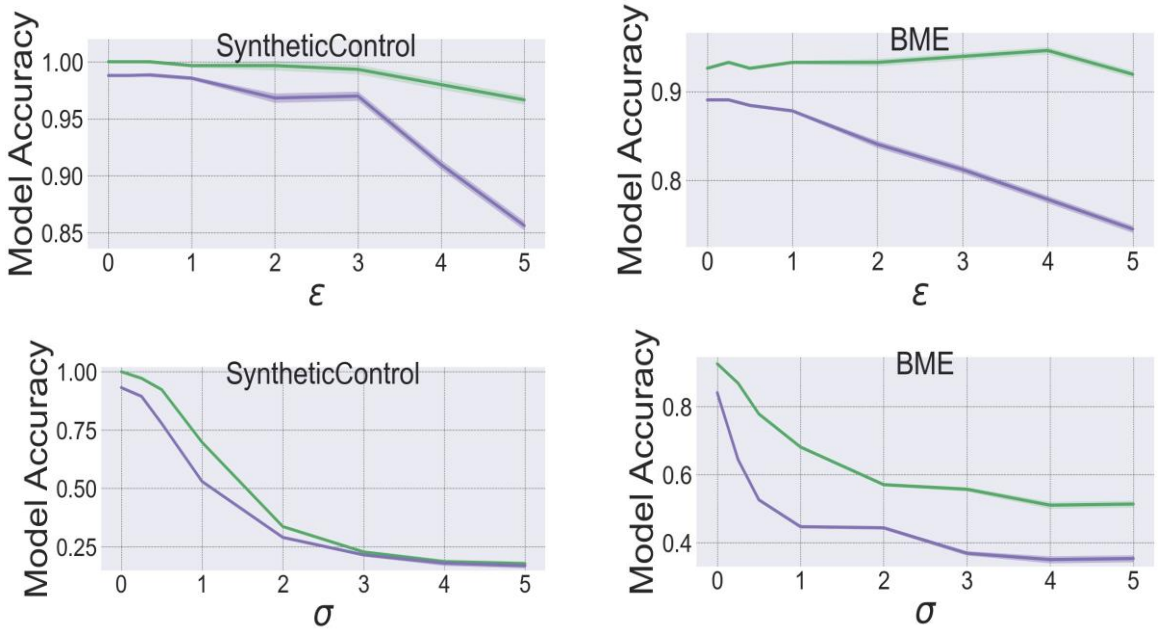
RO-TS algorithm

l_2

using Gaussian perturbation σ and adversarial perturbation ε on input

RO-TS Training Effectiveness – Setting #3

RO-TS vs. Stability training



Comparison of

- RO-TS algorithm
- Stability Training

using

Gaussian perturbation σ and adversarial perturbation ε on input

Conclusion

Summary

- Our comprehensive study shows that these deep learning methods are significantly vulnerable to adversarial attacks during real-world deployment

- We have proposed two different approaches, namely TSA-STAT and DTW-AR, to create more effective adversarial examples for the time-series domain

- We have proposed a novel derivation of a theoretically certified bound for adversarial robustness based on the TSA-STAT framework that applies to any deep model for the time-series domain

- We have shown applications of robust deep learning for time-series data on wearable devices for Human Activity recognition applications
  - Natural perturbations and imputation of missing sensor data

Summary

- We proposed a novel algorithm to train robust deep neural networks for time-series domain (RO-TS) and the theoretically-sound stochastic compositional alternating gradient descent and ascent (SCAGDA) algorithm that carefully leverages the structure of the optimization problem to solve it efficiently


thank you