Dynamic Time Warping Based Adversarial Framework for Time-Series Domain

Taha Belkhouja ©, Student Member, IEEE, Yan Yan, Member, IEEE, and Janardhan Rao Doppa ©, Member, IEEE

Abstract—Despite the rapid progress on research in adversarial robustness of deep neural networks (DNNs), there is little principled work for the time-series domain. Since time-series data arises in diverse applications including mobile health, finance, and smart grid, it is important to verify and improve the robustness of DNNs for the time-series domain. In this paper, we propose a novel framework for the time-series domain referred as Dynamic Time Warping for Adversarial Robustness (DTW-AR) using the dynamic time warping measure. Theoretical and empirical evidence is provided to demonstrate the effectiveness of DTW over the standard euclidean distance metric employed in prior methods for the image domain. We develop a principled algorithm justified by theoretical analysis to efficiently create diverse adversarial examples using random alignment paths. Experiments on diverse real-world benchmarks show the effectiveness of DTW-AR to fool DNNs for time-series data and to improve their robustness using adversarial training.

Index Terms—Adversarial examples, deep neural networks, dynamic time warping, robustness, time series.

I. INTRODUCTION

To deploy deep neural network (DNN) based systems in important real-world applications such as healthcare, we need them to be robust [1], [2], [3]. Adversarial methods expose the brittleness of DNNs [3], [4] and motivate methods to improve their robustness. There is little principled work for the time-series domain [43] even though this type of data arises in many real-world applications including mobile health [5], finance [6], and smart grid analytics [7]. The time-series modality poses unique challenges for studying adversarial robustness that are not seen in images [8] and text [9]. The standard approach of imposing an \( l_p \)-norm bound to create worst possible scenarios from a learning agent’s perspective does not capture the true similarity between time-series instances. Consequently, \( l_p \)-norm constrained perturbations can potentially create adversarial examples that correspond to a completely different class label. There is no prior work on filtering methods in the signal processing literature to automatically identify such adversarial candidates. Hence, adversarial examples from prior methods based on \( l_p \)-norm will confuse the learner when they are used to improve the robustness of DNNs. In other words, the accuracy of DNNs will degrade on real-world data after adversarial training.

This paper proposes a novel adversarial framework for time-series domain referred as Dynamic Time Warping for Adversarial Robustness (DTW-AR) to address the above-mentioned challenges. DTW-AR employs the dynamic time warping measure [10], [11] as it can be used to measure a realistic distance between two time-series signals (e.g., invariance to shift and scaling operations) [11], [12]. For example, a signal that has its frequency changed due to Doppler effect would output a small DTW measure to the original signal. However, if euclidean distance is used, both signals would look very dissimilar, unlike the reality. We theoretically analyze the suitability of DTW measure over the euclidean distance. Specifically, the space of candidate adversarial examples in the DTW space is a superset of those in euclidean space for the same distance bound. Therefore, DTW measure provides a more appropriate bias than the euclidean space for the time series domain and our experiments demonstrate practical benefits of DTW-based adversarial examples. While certain time-series classification tasks can be solved using low-complexity algorithms such as INN-DTW and avoid the adversarial setting, we find that deep models are better suited for multivariate time-series data. Due to the rising complexity of time-series data in several applications (e.g., mobile health [5], Human activity recognition [13], or sleep monitoring [14]), low-complexity algorithms such as kNN-DTW can potentially perform badly on high-dimensional multivariate data as we demonstrate in Section B, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3224754. Therefore, the adversarial setting remains applicable for time-series domain.

To create targeted adversarial examples, we formulate an optimization problem with the DTW measure bound constraint and propose to solve it using an iterative gradient-based approach. However, this simple method has two drawbacks. First, this method allows us to only find one valid adversarial example out of multiple solution candidates from the search space because it operates on a single optimal alignment. Second, we need to compute DTW measure in each iteration as the optimal DTW alignment path changes over iterations. Since the number of iterations are typically large and DTW computation is expensive, the overall algorithm becomes prohibitively slow. To
The key contribution of this paper is the development of any given distance metric (e.g., $\| \cdot \|_p$ norm) = \{Z\}_{\in\mathbb{X}} \in\mathbb{R}^n \times \mathbb{T}$ and using the following equations:

- The optimal alignment path (shown in green color) is $\{(i,j)\}_{i \leq T, j \leq T}$.
- The sequence of cells $X_{i,j}$ is called an adversarial example of $F$.

Fig. 1. Illustration of DTW alignment between two uni-variate signals $X$ and $Z$ of length 4. The optimal alignment path (shown in green color) is $P = \{(1,1), (2,2), (3,2), (4,2), (4,3), (4,4)\}$.

$X_{adv}$ is called an adversarial example of $X$ if:

$$\{ X_{adv} \mid \text{DIST}(X_{adv}, X) \leq \delta \text{ and } F_\theta(X) \neq F_\theta(X_{adv}) \}$$

where $\delta$ defines the neighborhood of highly-similar examples for input $X$ using a distance metric $\text{DIST}$ to create worst-possible outcomes from the learning agent’s perspective. Note that adversarial examples depend on the target concept because it defines the notion of invariance we care about.

Challenges for Time-Series Data. The standard $l_p$-norm distance does not capture the unique characteristics (e.g., fast-pace oscillations, sharp peaks) and the appropriate notion of invariance for time-series signals. Hence, $l_p$-norm based perturbations can lead to a time-series signal that semantically belongs to a different class-label. Our experiments show that small perturbations result in adversarial examples whose $l_2$ distance from the original time-series signal is greater than the distance between time-series signals from two different class labels (see Section V-B). Therefore, we need to study new methods by exploiting the structure and unique characteristics of time-series signals.

DTW Measure. The DTW measure between two uni-variate signals $X$ and $Z \in \mathbb{R}^T$ is computed via a cost matrix $C \in \mathbb{R}^{T \times T}$ using a dynamic programming (DP) algorithm with time-complexity $O(T^2)$. The cost matrix is computed recursively using the following equation:

$$C_{i,j} = d(X_i, Z_j) + \min \{ C_{i-1,j}, C_{i,j-1}, C_{i-1,j-1} \}$$

where $d(\cdot, \cdot)$ is any given distance metric (e.g., $\| \cdot \|_p$ norm). The DTW measure between signals $X$ and $Z$ is $\text{DTW}(X, Z) = C_{T,T}$. The sequence of cells $P = \{c_{i,j} = (i,j)\}$ contributing to $C_{T,T}$ is the optimal alignment path between $X$ and $Z$. Fig. 1 provides illustration for an optimal alignment path. We note that the diagonal path corresponds to the Euclidean distance metric.

For the multi-variate case, where $X$ and $Z \in \mathbb{R}^{n \times T}$, to measure the DTW measure using (1), we have $d(X_i, Z_j) \in \mathbb{R}^n$ [15]. We define the distance function $\text{dist}_P(X, Z)$ between time-series inputs $X$ and $Z$ according to an alignment path $P$ using the following equations:

$$\text{dist}_P(X, Z) = \sum_{(i,j) \in P} d(X_i, Z_j)$$

**II. BACKGROUND AND PROBLEM SETUP**

Let $X \in \mathbb{R}^{n \times T}$ be a multi-variate time-series signal, where $n$ is the number of channels and $T$ is the window-size of the signal. We consider a DNN classifier $F_\theta : \mathbb{R}^{n \times T} \rightarrow \mathbb{Y}$, where $\theta$ stands for parameters and $\mathbb{Y}$ is the set of classification labels. Table I summarizes the different mathematical notations used in this paper.

| Variable | Definition |
|----------|------------|
| $F_\theta$ | DNN classifier with parameters $\theta$ |
| $\mathbb{R}^{n \times T}$ | Time-series input space, where $n$ is the number of channels and $T$ is the window-size |
| $X_{adv}$ | Adversarial example generated from time-series input $X \in \mathbb{R}^{n \times T}$ |
| $\mathbb{Y}$ | Set of output class labels |
| $DTW(\cdot, \cdot)$ | Dynamic time warping distance |
| $P$ | Alignment path: a sequence of cost matrix cells $\{(i,j)\}_{i \leq T, j \leq T}$ |
| $C$ | Alignment cost matrix generated by dynamic programming with elements $C_{i,j}$ |
| $\delta$ | Distance bound constraint |

$X_{adv}$ is called an adversarial example of $X$ if:

$$\{ X_{adv} \mid \text{DIST}(X_{adv}, X) \leq \delta \text{ and } F_\theta(X) \neq F_\theta(X_{adv}) \}$$

### Table I: Mathematical Notations Used in This Paper
Hence, the DTW measure between \( X \) and \( Z \) is given by:

\[
DTW(X, Z) = \min_{P} \text{dist}_P(X, Z)
\]

III. DYNAMIC TIME WARPING BASED ADVERSARIAL ROBUSTNESS FRAMEWORK

The DTW-AR framework creates targeted adversarial examples for time-series domain using the DTW measure as illustrated in Fig. 2. For any given time-series input \( X \), DNN classifier \( F_\theta \), and distance bound \( \delta \), we solve an optimization problem to identify an adversarial example \( X_{adv} \) which is within DTW measure \( \delta \) to the original time-series signal \( X \). In what follows, we first provide empirical and theoretical results to demonstrate the suitability of DTW measure over euclidean distance for adversarial robustness studies in the time-series domain (Section III-A). Next, we introduce the optimization formulation based on the DTW measure to create adversarial examples and describe its main drawbacks (Section III-B). Finally, we explain our key insight of using stochastic alignment paths to successfully overcome those drawbacks to efficiently create diverse adversarial examples and provide theoretical justification (Section III-C).

A. Effectiveness of DTW Measure Measure

Empirical Justification. As we argued before, the standard \( l_2 \) distance is impractical for adversarial learning in time-series domain. Perturbations based on euclidean distance can result in adversarial time-series signals which semantically belong to a different class-label. Based on the real-world data representation provided in Fig. 4, we create and show in Fig. 3 an intuitive illustration of suitability of DTW over \( l_2 \) distance to explain the advantages of DTW as a similarity measure.

It shows the difference in the true data distribution in euclidean space (i.e., \( l_2 \) is used as the similarity measure) and in DTW space (i.e., DTW is used as the similarity measure) for two classes shown in red and green colors. The concentric circles represent the close-similarity area of each input instance (i.e., center) using the corresponding distance measure. This abstraction is tightly based on the observations made on real-world data. We employ multi-dimensional scaling (MDS), a visual representation of dissimilarities between sets of data points [16], to compare DTW and euclidean spaces. MDS is a dimensionality reduction method that preserves the distances between data points in the original space. Fig. 4 shows MDS results of SC dataset. DTW space exhibits better clustering for same-class data than euclidean space.
to the euclidean space, as provided in Fig. 4. An adversarial example for an SC data point in the green-labeled class is more likely to semantically belong to the red-labeled distribution in the euclidean space. However, in the DTW space, the adversarial example is more likely to remain in the green-labeled space, while only being misclassified by the DNN classifier due the adversarial problem.

**Theoretical Justification.** We prove that the DTW measure allows DTW-AR to explore a larger space of candidate adversarial examples when compared to perturbations based on the euclidean distance, i.e., identifies blind spots of prior methods. This result is based on the fact that the point-to-point alignment (i.e., euclidean distance) between two time-series signals is not always the optimal alignment. Hence, the existence of adversarial examples which are similar based on DTW and may not be similar based on the euclidean distance. To formalize this intuition, we provide Observation 1. We characterize the effectiveness of DTW-AR based attack as better for their ability to extend the space of attacks based on the euclidean distance and their potential to fool DNN classifiers that rely on euclidean distance for adversarial training. Our experimental results demonstrate that DTW-AR generates effective adversarial examples to fool the target DNN classifiers by leveraging the appropriate bias of DTW measure.

**Observation 1.** Let $l_2$ be the equivalent of euclidean distance using the cost matrix in the DTW space. $\forall X \in \mathbb{R}^{n \times T}$, a constrainted DTW space for adversarial examples is a strict superset of a constrained euclidean space for adversarial examples. If $X \in \mathbb{R}^{n \times T}$:

$$\left\{ X_{adv} | DTW(X, X_{adv}) \leq \delta \right\} \supset \left\{ X_{adv} | \|X - X_{adv}\|_2 \leq \delta \right\}$$

As an extension of Observation 1, the above theorem states that in the space where adversarial examples are constrained using a DTW measure bound, there exists more adversarial examples that are not part of the space of adversarial examples based on the euclidean distance for the same bound (i.e., blind spots). This result implies that DTW measure has an appropriate bias for the time-series domain. We present the proofs of both Observation 1 and Theorem 1 in the Appendix, available in the online supplemental material. Hence, our DTW-AR framework is potentially capable of creating more effective adversarial examples than prior methods based on $l_2$ distance for the same distance bound constraint. These adversarial examples are potentially more effective as they are able to break deep models by leveraging the appropriate bias of DTW measure.

However, to convert this potential to reality, we need an algorithm that can efficiently search this larger space of attacks to identify most or all adversarial examples which meet the DTW measure bound. Indeed, developing such an algorithm is one of the key contributions of this paper.

### B. Naive Optimization Based Formulation and Challenges to Create Adversarial Examples

To create adversarial examples to fool the given DNN $F_\theta$, we need to find an optimized perturbation of the input time-series $X$ to get $X_{adv}$. Our approach is based on minimizing a loss function $\mathcal{L}$ using gradient descent that achieves two goals.

1. Misclassification goal: Adversarial example $X_{adv}$ to be misclassified by $F_\theta$ as a target class-label $y_{target}$; and 2) DTW similarity goal: close DTW-based similarity between time-series $X$ and adversarial example $X_{adv}$.

To achieve the mis-classification goal, we employ the formulation of [17] to define a loss function:

$$L^{label}(X_{adv}) = \max_{y \neq y_{target}} (S_{y}(X_{adv}))$$

$$-S_{y_{target}}(X_{adv}) \rho$$

where $\rho < 0$. It ensures that the adversarial example will be classified by the DNN as class-label $y_{target}$ with a confidence $|\rho|$ using the output of the pre-softmax layer $\{S_{y}\}_{y \in Y}$.

To achieve the DTW similarity goal, we need to create $X_{adv}$ for a given time-series input $X$ such that $DTW(X, X_{adv}) \leq \delta$. We start by a naive optimization over the DTW measure using the Soft-DTW measure $SDTW(X, X_{adv})$ [18]. Hence, the DTW similarity loss function is:

$$L^{DTW}(X_{adv}) = SDTW(X, X_{adv})$$

The final loss function $\mathcal{L}$ we want to minimize to create optimized adversarial example $X_{adv}$ is:

$$\mathcal{L}(X_{adv}) = L^{label}(X_{adv}) + L^{DTW}(X_{adv})$$

We operate under white-box setting and can employ gradient descent to minimize the loss function in (*) over $X_{adv}$. This approach works for black-box setting also. In this work, we consider the general case where we do not query the black-box target DNN classifier. We show through experiments that the created adversarial examples can generalize to fool other black-box DNNs.

**Challenges of Naive Approach.** Recall that our overall goal is to identify most or all targeted adversarial time-series examples that meet the DTW measure bound. This will allow us to improve the robustness of DNN model using adversarial training. This naive approach has two main drawbacks.

- **Single adversarial example.** The method allows us to only find one valid adversarial example out of multiple solution candidates from the search space because it operates on a single optimal alignment path. Using a single alignment path (whether the diagonal path for euclidean distance or the optimal alignment path generated by DTW), the algorithm will be limited to the adversarial examples which use that single alignment. In Fig. 5, we provide a conceptual illustration of $S_{ADV}(X)$, the set of all adversarial examples $X_{adv}$ which meet the distance bound constraint $DTW(X, X_{adv}) \leq \delta$. In the euclidean space, using $l_2$ norm is sufficient to explore the entire search space around the original input to create adversarial examples. However, in the DTW space, each colored section in $S_{ADV}(X)$ can only be found using a subset of candidate alignment paths.
Algorithm 1: DTW-AR Based Adversarial Algorithm.

Input: time-series $X$; DNN classifier $F_{\theta}$; target class-label $y_{\text{target}}$; learning rate $\eta$; maximum iterations MAX

Output: adversarial example $X_{\text{adv}}$

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

- **High computational cost.** DTW is non-differentiable and approximation methods are often used in practice. These methods require $O(n^2T^2)$ to fill the cost matrix and $O(T)$ to backtrack the optimal alignment path. These steps are computationally-expensive. Gradient-based optimization iteratively updates the adversarial example $X_{\text{adv}}$ to achieve the DTW similarity goal, i.e., $DTW(X, X_{\text{adv}}) \leq \delta$, and the mis-classification goal, i.e., $F_{\theta}(X_{\text{adv}}) = y_{\text{target}}$. Standard algorithms such as projected gradient descent (PGD) [1] and Carlini & Wagner (CW) [17] require a large number of iterations to generate valid adversarial examples. This is also true for the recent computer vision specific adversarial algorithms [19], [20]. For time-series signals arising in many real-world applications, the required number of iterations to create successful attacks can grow even larger. We need to compute DTW measure in each iteration as the optimal DTW alignment path changes over iterations. Therefore, it is impractical to use the exact DTW computation algorithm to create adversarial examples. We also show that the existing optimized approaches to estimate the DTW measure remain computationally expensive for an adversarial framework. We provide results to quantify the runtime cost in our experimental evaluation.

C. Stochastic Alignment Paths for the DTW Similarity

Goal and Theoretical Justification

In this section, we describe the key insight of DTW-AR to overcome the above-mentioned two challenges and provide theoretical justification.

To overcome the above-mentioned two challenges of the naive approach, we propose the use of a random alignment path to create adversarial attacks on DNNs for time-series domain. The key idea is to select a random alignment path $P$ and to execute our adversarial algorithm while constraining over $dist_P(X, X_{\text{adv}})$ instead of $DTW(X, X_{\text{adv}})$. This choice is justified from a theoretical point-of-view due to the special structure in the problem to create DTW based adversarial examples. Using the distance function $dist_P(X, X_{\text{adv}})$, we redefine (6) as follows:

$$L^{\text{DTW}}(X_{\text{adv}}, P) = \alpha_1 \times dist_P(X, X_{\text{adv}}) - \alpha_2 \times dist_{P_{\text{diag}}}(X, X_{\text{adv}})$$  \hspace{1cm} (7)

where $\alpha_1 > 0$, $\alpha_2 \geq 0$, $P_{\text{diag}}$ is the diagonal alignment path equivalent to the euclidean distance, and $P$ is a given alignment path ($P \neq P_{\text{diag}}$). The first term of (7) is defined to bound the DTW similarity of adversarial example $X_{\text{adv}}$ to a threshold $\delta$ as stated in Observation 2. The second term represents a penalty term to account for adversarial example with close euclidean distance to the original input $X$ and pushes the algorithm to look beyond adversarial examples in the euclidean space. The coefficients $\alpha_1$ and $\alpha_2$ contribute in defining the position of the adversarial output in the DTW and/or euclidean space. If $\alpha_2 \rightarrow 0$, the adversarial example $X_{\text{adv}}$ will be highly similar to the original input $X$ in the DTW space with no consideration to the euclidean space. Hence, the adversarial example may be potentially adversarial in the euclidean space also. However, if $\alpha_2 > 0$, the adversarial output will be highly similar to the original input in the DTW space but out of the scope of adversarial attacks in the euclidean space (i.e., a blind spot). Recall from Theorem 1 that DTW space allows more candidate adversarial examples than euclidean space. Hence, this setting allows us to find blind spots of euclidean space based attacks.

The DTW-AR approach to create adversarial examples is shown in Algorithm 1. We note that the naive approach that uses Soft-DTW with the Carlini & Wagner loss function is a sub-case of DTW-AR as shown below:

$$SDTW(X, X_{\text{adv}}) = L^{\text{DTW}}(X_{\text{adv}}, P_{\text{DTW}}) = 1 \times dist_{P_{\text{DTW}}}(X, X_{\text{adv}}) - 0 \times dist_{P_{\text{diag}}}(X, X_{\text{adv}})$$  \hspace{1cm} (8)

where $P_{\text{DTW}}$ is the optimal DTW alignment path.

Observation 2. Given any alignment path $P$ and two multivariate time-series signals $X, Z \in \mathbb{R}^{n \times T}$. If we have $dist_P(X, Z) \leq \delta$, then $DTW(X, Z) \leq \delta$.

Observation 2 states that $dist_P(X, Z)$ defined with respect to a path $P$ is always an upper bound for $DTW(X, Z)$, since DTW uses the optimal alignment path. Hence, when the alignment path is fixed, the time-complexity is reduced to a simpler similarity measure that requires only $O(n.T)$, which results in significant computational savings due to repeated calls within the adversarial algorithm.
Our stochastic alignment method also improves the search strategy for finding multiple desired adversarial examples. Suppose \( S_{ADV}(X) \) is the set of all adversarial examples \( X_{adv} \) which meet the distance bound constraint \( DTW(X, X_{adv}) \leq \delta \). Each adversarial example in \( S_{ADV}(X) \) can be found using only a subset of candidate alignment paths. By using a stochastic alignment path, we can leverage the large pool of different alignment paths to uncover more than one adversarial example from \( S_{ADV}(X) \). On the other hand, if the exact DTW computation based algorithm was feasible, we would only find a single \( X_{adv} \), as DTW based algorithm operates on a single optimal alignment path.

**Theoretical Tightness of Bound.** While Observation 2 provides an upper bound for the DTW measure, it does not provide any information about the tightness of the bound. To analyze this gap, we need to first define a similarity measure between two alignment paths to quantify their closeness. We define \( PathSim \) as a similarity measure between two alignment paths \( P_1 \) and \( P_2 \) in the DTW cost matrix of time-series signals \( X, Z \in \mathbb{R}^{n \times T} \). Let \( P_1 = \{c_1, \ldots, c_T\} \) and \( P_2 = \{c'_1, \ldots, c'_T\} \) represent the sequence of cells for paths \( P_1 \) and \( P_2 \) respectively.

\[
PathSim(P_1, P_2) = \frac{1}{2T} \left( \sum_{c_i} \min_{c'_j} \|c_i^1 - c'_j^1\|_1 + \sum_{c_i} \min_{c'_j} \|c_i^2 - c'_j^2\|_1 \right)
\]

As \( PathSim(P_1, P_2) \) approaches 0, \( P_1 \) and \( P_2 \) are very similar, and they will be the exact same path if \( PathSim(P_1, P_2) = 0 \). For \( X, Z \in \mathbb{R}^{n \times T} \), two very similar alignment paths corresponds to a similar feature alignment between \( X \) and \( Z \). Theorem 2 shows the tightness of the bound given in Observation 2 using the path similarity measure defined above.

**Theorem 2.** For a given input \( X \in \mathbb{R}^{n \times T} \) and a random alignment path \( P_{rand} \), the resulting adversarial example \( X_{adv} \) from the minimization over \( dist_{P_{rand}}(X, X_{adv}) \) is equivalent to minimizing over \( DTW(X, X_{adv}) \). For any \( X_{adv} \) generated by DTW-AR using \( P_{rand} \), we have:

\[
\begin{align*}
\text{PathSim}(P_{rand}, P_{DTW}) &= 0 \\
\text{dist}_{P_{rand}}(X, X_{adv}) &= DTW(X, X_{adv})
\end{align*}
\]

where \( P_{DTW} \) is the optimal alignment path found using DTW computation between \( X \) and \( X_{adv} \).

**Similarity Measure PathSim Definition.**

For DTW-AR, we rely on a stochastic alignment path to compute \( dist_P \) defined in (2). To improve our understanding of the behavior of DTW-AR framework based on stochastic alignment paths, we propose to define a similarity measure that we call \( PathSim \). This measure quantifies the similarities between two alignment paths \( P_1 \) and \( P_2 \) in the DTW cost matrix for two time-series signals \( X, Z \in \mathbb{R}^{n \times T} \). If we denote the alignment path sequence \( P_1 = \{c_1, \ldots, c_T\} \) and \( P_2 = \{c'_1, \ldots, c'_T\} \), then we can measure their similarity as defined in (9).

This definition is a valid similarity measure as it satisfies all the distance axioms [21]:

**Non-Negativity.** By definition, \( PathSim(P_1, P_2) \) is a sum of \( l_1 \) distances, which are all positives. Hence, \( PathSim(P_1, P_2) \geq 0 \).

**Unicity:** \( PathSim(P_1, P_2) = 0 \)

\[
\iff \frac{1}{2T} \left( \sum_{c_i} \min_{c'_j} \|c_i^1 - c'_j^1\|_1 + \sum_{c_i} \min_{c'_j} \|c_i^2 - c'_j^2\|_1 \right) = 0
\]

As we have a sum equal to 0 of all positive terms, we can conclude that each term \( \min \| \cdot \|_1 \) is equal to 0:

\[
\iff \forall i : \|c_i^1 - c_i^2\|_1 = 0
\]

As both paths have the same sequence of cells, we can safely conclude that \( PathSim(P_1, P_2) = 0 \iff P_1 = P_2 \):

**Symmetric Property:**

\[
PathSim(P_1, P_2) = \frac{1}{2T} \left( \sum_{c_i} \min_{c'_j} \|c_i^1 - c'_j^1\|_1 + \sum_{c_i} \min_{c'_j} \|c_i^2 - c'_j^2\|_1 \right) = PathSim(P_2, P_1)
\]

Note that the triangle inequality is not applicable as the alignment path spaces does not support additive operations. This similarity measure quantifies the similarity between two alignment paths as it measures the \( l_1 \) distance between the different cells of each path. The multiplication factor \( 1/2T \) is introduced to prevent scaling of the measure for large \( T \) values for a given time-series input space \( \mathbb{R}^{n \times T} \).

In Fig. 6, we visually show the relation between \( PathSim \) measure and the alignment path for a given cost matrix. We observe that when \( PathSim(P_1, P_2) \to 0 \), \( P_1 \) and \( P_2 \) are very similar, and they will be the exact same path if \( PathSim(P_1, P_2) = 0 \). For \( PathSim(P_1, P_2) \gg 0 \), the alignment path will go through different cells which are far-placed from each other in the cost matrix.

**Empirical Tightness of Bound.** Fig. 7 shows that over the iterations of the DTW-AR algorithm, the updated adversarial example yields to an optimal alignment path that is more similar to the input random path. This result strongly demonstrate that Theorem 2 holds empirically.

**Corollary 1.** Let \( P_1 \) and \( P_2 \) be two alignment paths such that \( PathSim(P_1, P_2) > 0 \). If \( X_{adv}^1 \) and \( X_{adv}^2 \) are the adversarial examples generated using DTW-AR from any given time-series \( X \) using paths \( P_1 \) and \( P_2 \) respectively such that
BELKHOUA et al.: DYNAMIC TIME WARPING BASED ADVERSARIAL FRAMEWORK FOR TIME-SERIES DOMAIN

Fig. 6. Visualization of PathSim values along different example alignment paths in $\mathbb{R}^{n \times 10}$ (First row) and $\mathbb{R}^{n \times 100}$ (Second row) spaces.

Fig. 7. (a) Example of the convergence of the optimal alignment path between the adversarial 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). (b) PathSim score of the optimal alignment path between the adversarial example and the original example and the given random path for the ECG200 dataset averaged over multiple random alignment paths.

$DTW(X, X^1_{adv}) = \delta$ and $DTW(X, X^2_{adv}) = \delta$, then $X^1_{adv}$ and $X^2_{adv}$ are not necessarily the same.

Theorem 2 shows that the adversarial example generation using DTW-AR is equivalent to the ideal setting where it is possible to optimize $DTW(X, X_{adv})$. The above corollary extends Theorem 2 to show that if we employ different alignment paths within Algorithm 1, we will be able to find more adversarial examples which meet the distance bound in contrast to the naive approach.

IV. RELATED WORK

Adversarial Methods. Prior work on adversarial examples mostly focus on image and text domains [8], [9]. Such methods include Carlini & Wagner attack [17], boundary attack [22], and universal attacks [23]. Recent work focuses on regularizing adversarial example generation methods to obey intrinsic properties of images [24], [25], [26]. In NLP domain, methods to fool text classifiers employ the saliency map of input words to generate adversarial examples while preserving meaning to a human reader in white-box setting [27]. DeepWordBug [28] employs a black-box strategy to fool classifiers with simple character-level transformations. Since characteristics of time-series (e.g., fast-pace oscillations, sharp peaks) are different from images and text, prior methods are not suitable to capture the appropriate notion of invariance for time-series domain.

Adversarial Robustness. Adversarial training is one of the strongest empirical defense methods against adversarial attacks [1], [29]. This involves employing attack methods to create adversarial examples to augment the training data for improving robustness. Stability training [30] is an alternative method that explicitly optimizes for robustness by defining a loss function that evaluates the classifier on small perturbations of clean examples. This method yield to a deep network that is stable against natural and adversarial distortions in the visual input. There are other defense methods which try to overcome injection of adversarial examples [31], [32], [33]. However, for time-series domain, as $l_p$-norm based perturbations may not guarantee preserving the semantics of true class label, adversarial examples may mislead DNNs during adversarial training resulting in accuracy degradation.

Adversarial Attacks for Time-Series Domain. There is little to no principled prior work on adversarial methods for time-series. Fawaz et al. [2] employed the standard Fast Gradient Sign method [34] to create adversarial noise with the goal of reducing the confidence of deep convolutional models for classifying

1In a concurrent work, Belkhouja et al., developed an adversarial framework for using statistical features [44], [45] and another min-max optimization methods [46] to explicitly train robust deep models for time-series domain using global alignment kernels.
uni-variate time-series. Network distillation is employed to train a student model for creating adversarial attacks [3]. However, this method is severely limited: it can generate adversarial examples for only a small number of target labels and cannot guarantee generation of adversarial example for every input. [4] tried to address adversarial examples with elastic similarity measures, but does not propose any elastic-measure based attack algorithm.

**Time-Series Pre-Processing Methods.** A possible solution to overcome the euclidean distance concerns is to introduce pre-processing steps that are likely to improve the existing frameworks. Simple pre-processing steps such as MinMax-normalization or z-normalization only solves problems such as scaling problem. However, they do not address any concern about signal-warping or time-shifts. Other approaches rely on learning feature-preserving representations. A well-known example is the GRAIL [35] framework. This framework aims to learn compact time-series representations that preserve the properties of a pre-defined comparison function such as DTW. The main concern about feature-preserving pre-processing steps is that the representation learnt is not reversible. In other words, a real-world time-series signal cannot be generated from the estimated representation. The goal of adversarial attacks is to create real-world-time-series that can be used to fool any DNN. Such challenges would limit the usability and the generality of methods based on pre-processing steps to study the robustness of DNNs for time-series data.

In summary, existing methods for time-series domain are lacking in the following ways: 1) they do not create targeted adversarial attacks; and 2) they employ $l_p$-norm based perturbations which do not take into account the unique characteristics of time-series data.

**V. Experiments and Results**

We empirically evaluate the DTW-AR framework and discuss the results along different dimensions.

**A. Experimental Setup**

**Datasets.** We employ the UCR datasets benchmark [36]. We present the results on five representative datasets (Atrial-Fibrillation, Epilepsy, ERing, Heartbeat, Rack-etsSports) from diverse domains noting that our findings are general as shown by the results on remaining UCR datasets in the Appendix, available in the online supplemental material. We employ the standard training/validation/testing splits from these benchmarks.

**Configuration of Algorithms.** We employ a 1D-CNN architecture for the target DNNs. We operate under a white-box (WB) setting for creating adversarial examples to fool WB model. To assess the effectiveness of attacks, we evaluate the attacks under the black-box (BB) setting and to fool BB model. Neural architectures of both BB and WB models are in the Appendix, available in the online supplemental material. The adversarial algorithm has no prior knowledge/querying ability of target DNN classifiers. Target DNNs include: 1) DNN model with a different architecture trained on clean data (BB); 2) DNNs trained using augmented data from baselines attacks that are not specific to image domain: Fast Gradient Sign method (FGS) [34], Carlini & Wagner (CW) attack [17], and Projected Gradient Descent (PGD) [1]; and 3) DNN models trained using stability training [30] (STN) for learning robust classifiers.

**Evaluation Metrics.** We evaluate attacks using the efficiency metric $\alpha_{Eff} \in [0,1]$ over the created adversarial examples. $\alpha_{Eff}$ (higher means better attacks) measures the capability of adversarial examples to fool a given DNN $F_D$ to output the target class-label. $\alpha_{Eff}$ is calculated as the fraction of adversarial examples that are predicted correctly by the classifier:

$$\alpha_{Eff} = \frac{\# \text{Adv. examples s.t. } F(X) = \text{target class}}{\# \text{Adv. examples}}.$$  

We evaluate adversarial training by measuring the accuracy of the model to predict ground-truth labels of adversarial examples. A DNN classifier is robust if it is successful in predicting the true label of any given adversarial example.

**B. Results and Discussion**

**Spatial Data Distribution With DTW.** We have shown in Fig. 4 how the data from different class labels are better clustered in the DTW space compared to the euclidean space. These results demonstrate that DTW suits better the time-series domain as generated adversarial examples lack true-label guarantees. Moreover, euclidean distance based attacks can potentially create adversarial examples that are inconsistent for adversarial training. Our analysis showed that for datasets such as WISDM, there are time-series signals from different classes with $l_2$-distances $\leq 2$, while PGD or FGS require $\epsilon \geq 2$ to create successful adversarial examples for more than 70% time-series instances. We provide in the Appendix, available in the online supplemental material, an additional visualization of the adversarial examples using DTW.

**Admissible Alignment Paths.** The main property of DTW alignment is the one-to-many match between time-steps to identify similar warped pattern. Intuitively, if an alignment path matches few time-steps from the first signal with too many steps in the second signal, both signals are not considered similar. Consequently, the optimal path would be close to the corners of the cost matrix. Fig. 8 provides a comparison between two adversarial signals generated using a green colored path closer to the diagonal versus a red colored path that is close to the corners. We can see that the red path produces an adversarial example that is not similar to the original input. Hence, we limit the range of the random path $P_{rand}$ used to a safe range omitting the cells at the top and bottom halves of the top-left and bottom-right corners.
**Multiple Diverse Adversarial Examples Using DTW-AR.** In Section III-B, we argued that using stochastic alignment paths, we can create multiple diverse adversarial time-series examples within the same DTW measure bound. DTW-AR method leverages the large pool of candidate alignment paths to uncover more than one adversarial example as illustrated in Fig. 5. To further test this hypothesis, we perform the following experiment. We sample a subset of different (using PathSim) alignment paths \( \{P_{\text{rand}}\} \) and execute DTW-AR algorithm to create adversarial examples for the same time-series \( X \). Let \( X_{\text{adv}} \) be the adversarial example generated from \( X \) using \( P_{\text{rand}} \). We measure the similarities between the generated \( \{X_{\text{adv}}\} \) using DTW and \( l_2 \) distance. If the distance between two adversarial examples is less than a threshold \( \epsilon_{\text{sim}} \), then they are considered the same adversarial example.

Table II shows the percentage of adversarial examples generated using different alignment paths from a given time-series signal that are not similar to any other adversarial example. We conclude that DTW-AR algorithm indeed creates multiple different adversarial examples from a single time-series signal for the same DTW measure bound.

**Empirical Justification for Theorem 2.** We provided a proof for the gap between creating an adversarial example using the proposed DTW-AR algorithm and an ideal DTW algorithm. In Fig. 7(a), we provide an illustration of the optimal alignment path update using DTW-AR. This experiment was performed on the ECG200 dataset as an example (noting that we observed similar patterns for other datasets as well): the blue path represents the selected random path to be used by DTW-AR and the red path represents the optimal alignment path computed by DTW. At the beginning, the optimal alignment path (dotted path) and the random path are dissimilar. However, as the execution of DTW-AR progresses, the updated adversarial example yields to an optimal alignment path similar to the random path. In Fig. 7(b), we show the progress of the PathSim score as a function of the iteration numbers of Algorithm 1.

This figure shows the convergence of the PathSim score to 0. These strong results confirm the main claim of Theorem 2 that the resulting adversarial example \( X_{\text{adv}} \) from the minimization over \( \text{dist}_{P_{\text{rand}}}(X, X_{\text{adv}}) \) is equivalent to minimizing over \( \text{DTW}(X, X_{\text{adv}}) \).

| \( \epsilon_{\text{sim}} \) \( l_2 \) norm | \( \epsilon_{\text{sim}} \) DTW |
|---|---|
| 0.01 | 0.05 | 0.1 | 0.01 | 0.05 | 0.1 |
| Atrial Fibrillation | 98% | 90% | 87% | 100% | 100% | 98% |
| Epilepsy | 99% | 96% | 93% | 100% | 100% | 99% |
| E Ring | 99% | 95% | 93% | 100% | 100% | 98% |
| Heartbeat | 99% | 94% | 92% | 100% | 100% | 99% |
| RacketSports | 99% | 94% | 92% | 100% | 100% | 96% |

**Loss Function Scaling.** As the final loss function is using two different terms to create adversarial attacks, the absence of a scaling parameter can affect the optimization process. In Fig. 10, we demonstrate that empirically, the first term of the (*) plateaus at \( \rho \) before minimizing \( L_{\text{DTW}} \). The figure shows the progress of both \( L_{\text{label}} \) and \( L_{\text{DTW}} = \alpha_{\text{TIR}} \times \text{dist}_P(X, X_{\text{adv}}) \) over the first 100 iterations of DTW-AR algorithm. We conclude that there is no need to scale the loss function noting that our findings were similar for other time-series datasets. In the general case, if a given application requires attention to scaling both terms (\( L_{\text{DTW}} \) and \( L_{\text{label}} \)), the learning rate can be adjusted to two different values: Instead of having \( \eta \times \nabla L = \eta \times \nabla L_{\text{label}} + \eta \times \nabla L_{\text{DTW}} \), we can use a learning rate pair \( \eta = (\eta_1, \eta_2) \) and gradient descent step becomes \( \eta \times \nabla L = \eta_1 \times \nabla L_{\text{label}} + \eta_2 \times \nabla L_{\text{DTW}} \).

**Effectiveness of Adversarial Attacks.** Results of the fooling rate of DTW-AR generated attacks for different models are shown in Fig. 9. We observe that under the white-box setting (WB model), we have \( \alpha_{\text{Eff}} = 1 \). This shows that for any \( y_{\text{target}} \), DTW-AR successfully generates an adversarial example for every input in the dataset. For black-box setting (BB model) and other models using baseline attacks for adversarial training, we see that DTW-AR attack is highly effective for most cases. We conclude that these results support the theoretical claim made in Theorem 1 by showing that standard \( l_2 \)-norm based attacks have blind spots and the DTW bias is appropriate for
The importance of $\alpha_2$ in (7) is shown to improve the fooling rate of adversarial examples. To implement DTW-AR adversarial attacks, we have fixed $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$ for (7). The importance of $\alpha_2$ is to push the algorithm to create adversarial examples out of the scope of the euclidean space as shown in Fig. 11.

The adversarial examples with $\alpha_2 \neq 0$ evade DNNs with adversarial training baselines better than the examples with $\alpha_2 = 0$.

**DTW-AR Based Adversarial Training.** Our hypothesis is that $l_2$-based perturbations lack true-label guarantees and can degrade the overall performance of DNNs. Fig. 12 shows the accuracy of different DNNs after adversarial training on clean data. This performance is relative to the clean testing set of each dataset. We observe that all $l_2$-based methods degrade the performance using adversarial training for at least one dataset. However, for many datasets, the performance is visibly improved using DTW-AR based adversarial training. Compared to standard training (i.e., no augmented adversarial examples), the performance on AtrialFibrillation improved using DTW-AR while it declined with other methods; and on HeartBeat, DTW-AR based training improves from 70% to 75%.

To evaluate the accuracy of DTW-AR in predicting the ground-truth label of adversarial examples, we create adversarial examples using a given attack algorithm and label each example with the true class-label of the corresponding clean time-series input. Fig. 13 shows the results of DTW-AR based adversarial training using WB architecture.

In this experiment, we consider the adversarial examples that have successfully fooled the original DNN (i.e., no adversarial training). We observe that DNNs using DTW-AR for adversarial training are able to predict the original label of adversarial examples with high accuracy. We can see how FGS and PGD attacks cannot evade the DTW-AR based trained deep model for almost any dataset. These results show that DTW-AR significantly improves the robustness of deep models for time-series data to
evoke attacks generated by DTW-AR and other baseline attacks. For the adversarial training, we employ several values to create adversarial examples to be used in the training phase. We have set $\alpha_1 \in [0.1, 1]$ and $\alpha_2 \in [0, 1]$. In Fig. 14, we show the role of the term $\alpha_2$ of (7) in the robustness of the DNN. $\alpha_2 \in [0, 1]$ ensures diverse DTW-AR examples to increase the robustness of a given DNN. When set to 0, we see that there is no significant difference in the performance against baseline attacks. However, the DNN cannot defend against all DTW-AR attacks. We can also observe that the setting where $\alpha_2$ is strictly different than 0 is the worst, as the DNN does not learn from the adversarial examples that are found in the euclidean space by DTW-AR or the given baselines.

**Naive Approach: Carlini & Wagner With Soft-DTW.** Recall that naive approach uses DTW measure within the Carlini & Wagner loss function. SoftDTW [18] allows us to create a differentiable version of DTW measure. Hence, we provide results for this naive approach to verify if the the use of Soft-DTW with existing euclidean distance based methods can solve the challenges for the time-series domain mentioned in this paper. We compare the DTW-AR algorithm with the CW-SDTW that plugs Soft-DTW within the Carlini & Wagner algorithm instead of the standard $l_2$ distance. CW-SDTW has the following limitations when compared against DTW-AR:

- The time-complexity of Soft-DTW is quadratic in the dimensionality of time-series input space, whereas the distance computation in DTW-AR is linear.
- The CW-SDTW attack method is a sub-case of the DTW-AR algorithm. If DTW-AR algorithm uses the optimal alignment path instead of a random path, the result will be equivalent to a CW-SDTW attack.
- For a given time-series signal, CW-SDTW will output one single adversarial example and cannot uncover multiple adversarial examples which meet the DTW measure bound. However, DTW-AR algorithm gives the user control over the alignment path and can create multiple diverse adversarial examples.

In conclusion, both challenges that were explained in the Challenges of Naive approach in Section III-A cannot be solved using CW-SDTW. As a consequence, the robustness goal aimed by this paper cannot be achieved using solely CW-SDTW. Indeed, our experiments support this hypothesis. Fig. 15 shows that DTW-AR is successful to fool a DNN that uses adversarial examples from CW-SDTW for adversarial training. This shows that our proposed framework is better than this naive baseline. Fig. 16 shows that DTW-AR significantly improves the robustness of deep models for time-series as it is able to evade attacks generated by CW-SDTW. Both these experiments demonstrate that CW-SDTW is neither able to create stronger attacks nor a more robust deep model when compared to DTW-AR.

**Comparison With Karim et al. [3].** The approach from Karim et al. [3] employs network distillation to train a student model for creating adversarial attacks. However, this method is severely limited: only a small number of target classes yield adversarial examples and the method does not guarantee the

Fig. 14. Results of DTW-AR based adversarial training to predict the true labels of adversarial examples generated by DTW-AR and the baseline attack methods. The adversarial examples considered are those that successfully fooled DNNs that do not use adversarial training.
As shown in our experiments, DTW-AR generates at least one adversarial example for any variant of DTW. In Fig. 17, we demonstrate that using a limited number of adversarial examples in the white-box setting, to test the effectiveness of this attack against DTW-AR, Fig. 17 shows the success rate of deep model from DTW-AR based adversarial training to predict the true label of adversarial attacks generated using method from [3].

DTW-AR outperforms [3] due to following reasons:

- DTW-AR generates at least one adversarial example for every input $X \in \mathbb{R}^{n \times T}$ as shown in our experiments.
- Adversarial examples created by DTW-AR are highly effective against deep models relying on [3] for adversarial training as this baseline fails to create adversarial examples for many inputs and target classes (shown in [3]).
- Adversarial examples created by the method from [3] does not evade deep models from DTW-AR based adversarial training.

**Computational Runtime of DTW-AR versus DTW.** As explained in the technical section, optimization based attack algorithm requires a large number of iterations to create a highly-similar adversarial example. For example, $10^3$ iterations is the required default choice for CW to create successful attacks, especially, for large time-series in our experiments. The exact DTW method is non-differentiable, thus, it is not possible to perform experiments to compare DTW-AR method to the exact DTW method. Hence, we assume that each iteration will compute the optimal DTW path and use it instead of the random path. To assess the runtime of computing the DTW measure, we employ three different approaches: 1) The standard DTW algorithm, 2) The FastDTW [37] that was introduced to overcome DTW computational challenges, and 3) cDTW [38] that measures DTW in a constrained manner using warping windows. We note that FastDTW was proven to be inaccurate, and cDTW is faster and more accurate for computing DTW measure [39]. We show both baselines for the sake of completeness.

**DTW-AR Extension to Other Multivariate DTW Measures.** The DTW-AR framework relies on the distance function $dist_P(X, Z) = \sum_{(i,j) \in P} d(X_i, Z_j)$ between two time-series signals $X$ and $Z$ according to an alignment path $P$ to measure their similarity. Extending the DTW notion from univariate to multivariate is a known problem, where depending on the application, researchers’ suggest to change the definition of $dist_P(X, Z)$ to better fit the characteristics of the application at hand. In all cases, DTW-AR relies on using the final cost matrix $DTW(X, Z) = \min_P dist_P(X, Z)$ using dynamic programming $C_{i,j} = d(X_i, Z_j) + \min\{C_{i-1,j}, C_{i,j-1}, C_{i-1,j-1}\}$. Therefore, the use of different variants of $dist_P(X, Z)$ (e.g., $DTW_I$ or $DTW_D$ [15]) will only affect the cost matrix values, but will not change the assumptions and applicability of DTW-AR. Therefore, DTW-AR is general and can work with any variant of DTW. In Fig. 19, we demonstrate that using a different family of DTW ($DTW_I$) does not have a major impact on DTW-AR’s performance and effectiveness.

Fig. 15. Results for the effectiveness of adversarial examples from DTW-AR against adversarial training using examples created by CW-SDTW on different datasets.

Fig. 16. Results for the effectiveness of adversarial training using DTW-AR based examples against adversarial attacks from CW-SDTW on different datasets.

Fig. 17. Results of the success rate of deep model from DTW-AR based adversarial training to predict the true label of adversarial attacks generated using method in [3].

Fig. 18. Average runtime per iteration for standard DTW, FastDTW, cDTW, and DTW-AR (on NVIDIA Titan Xp GPU).
of DTW-AR framework using both alternative measures of multi-variate DTW does not affect the overall performance. Therefore, for a given specific application, the practitioner can configure DTW-AR appropriately.

C. Summary of Experimental Results

Our experimental results supported all the claims made in Section III. The summary list includes:

- Fig. 4 showed that DTW space is more suitable for adversarial studies in the time-series domain than euclidean distance to support Theorem 1.
- Using stochastic alignment paths, DTW-AR creates multiple diverse adversarial examples to support Corollary 1 (Table II), which is impossible using the optimal alignment path.
- Fig. 7 provides empirical justification for Theorem 2 showing that minimizing over a given alignment path is equivalent to minimizing using exact DTW method (bound is tight).
- Fig. 9 shows that adversarial examples created by DTW-AR have higher potential to break time-series DNN classifiers.
- Figs. 12 and 13 show that DTW-AR based adversarial training is able to improve the robustness of DNNs against baseline adversarial attacks.
- Figs. 15 and 16 shows that DTW-AR outperforms the naive approach CW-SDTW that uses SoftDTW with the Carlini & Wagner loss function. We also demonstrated several limitations of CW-SDTW to achieve the robustness goal aimed by this paper.
- Fig. 18 clearly demonstrates that DTW-AR significantly reduces the computational cost compared to existing approaches of computing the DTW measure for creating adversarial examples.
- Fig. 19 demonstrates that DTW-AR can generalize to any multivariate DTW measure (such as DTWAdaptive [15]) without impacting on its performance and effectiveness.

VI. Conclusion

We introduced the DTW-AR framework to study adversarial robustness of deep models for the time-series domain using dynamic time warping measure. This framework creates effective adversarial examples by overcoming the limitations of prior methods based on euclidean distance. We theoretically and empirically demonstrate the effectiveness of DTW-AR to fool deep models for time-series data and to improve their robustness. We conclude that the time-series domain needs focused investigation for studying robustness of deep models by shedding light on the unique challenges.

REFERENCES

[1] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in Proc. 6th Int. Conf. Learn. Representations, Vancouver, BC, Canada, Apr./May 30–3, 2018.
[2] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. Muller, “Adversarial attacks on deep neural networks for time-series classification,” in Proc. Int. Joint Conf. Neural Netw., 2019, pp. 1–8.
[3] F. Karim, S. Majumdar, and H. Darabi, “Adversarial attacks on time series,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 10, pp. 3309–3320, Oct. 2021.
[4] I. Oregi, J. D. Ser, A. Pérez, and J. A. Lozano, “Adversarial sample crafting for time series classification with elastic similarity measures,” in Proc. 12th Int. Symp. Intell. Distribut. Comput. Intell. Distribut. Comput., Springer, 2018, pp. 26–39.
[5] A. Ignatov, “Real-time human activity recognition from accelerometer data using convolutional neural networks,” Appl. Soft Comput., vol. 62, pp. 915–922, 2018.
[6] A. M. Özbayoglu, M. U. Güdelek, and O. B. Sezer, “Deep learning for financial applications: A survey,” Appl. Soft Comput., vol. 93, 2020, Art. no. 106384.
[7] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou, “Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids,” IEEE Trans. Ind. Inform., vol. 14, no. 4, pp. 1606–1615, Apr. 2018.
[8] Z. Kolter and A. Madry, “Tutorial adversarial robustness: Theory and practice,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2018.
[9] W. Y. Wang, S. Singh, and J. Li, “Deep adversarial learning for NLP,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol., A. Sarkar and M. Strube, Eds., Minneapolis, MN, USA, Jun. 02, 2019, pp. 1–5.
[10] H. Sakoe, “Dynamic-programming approach to continuous speech recognition,” in Proc. Int. Congr. Acoust., Budapest, 1971, pp. 64–83.
[11] M. Müller, “Dynamic time warping,” in Information Retrieval for Music and Motion, Berlin, Germany: Springer, 2007, pp. 69–84.
[12] D. J. Berndt and J. Clifford, “Using dynamic time warping to find patterns in time series,” in Proc. 3rd Int. Conf. Knowl. Discov. Data Mining, Seattle, WA, USA, 1994, pp. 359–370.
[13] E. Ramanujam, T. Perumal, and S. Padmavathi, “Human activity recognition with smartphone and wearable sensors using deep learning techniques: A review,” IEEE Sensors J., vol. 21, no. 12, pp. 13029–13040, Jun. 2021.
[14] M. Zhao, S. Yue, D. Katabi, T. S. Jaakkola, and M. T. Bianchi, “Learning sleep stages from radio signals: A conditional adversarial architecture,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 4104–4109.
[15] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, and E. Keogh, “Generalizing DTW to the multi-dimensional case requires an adaptive approach,” Data Mining Knowl. Discov., vol. 31, no. 1, pp. 1–31, 2017.
[16] A. Buja, D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen, “Data visualization with multidimensional scaling,” J. Comput. Graphical Statist., vol. 17, no. 2, pp. 444–472, 2008.
[17] N. Carlini and D. A. Wagner, “Towards evaluating the robustness of neural networks,” in Proc. IEEE Symp. Secure Privacy, San Jose, CA, USA, May 22–26, 2017, pp. 39–57.
[18] M. Cuturi and M. Blondel, “Soft-DTW: A differentiable loss function for time-series,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 894–903.
[19] C. Laidlaw and S. Feizi, “Functional adversarial attacks,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2019, pp. 10408–10418.
[20] A. Shafahi, M. Najibi, Z. Xu, J. Dickerson, L. S. Davis, and T. Goldstein, “Universal adversarial training,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 5636–5643.
[21] M. J. Cullinan, “Metric axioms and distance,” Math. Gazette, vol. 95, no. 534, pp. 414–419, 2011.
[22] W. Brendel, J. Rauber, and M. Bethge, “Decision-based adversarial attacks: Reliable attacks against black-box machine learning models,” in Proc. 6th Int. Conf. Learn. Representations, Vancouver, BC, Canada, Apr./May 30–03, 2018.

[23] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, “Universal adversarial perturbations,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Honolulu, HI, USA, Jul. 21–26, 2017, pp. 86–94.

[24] C. Laidlaw and S. Feizi, “Functional adversarial attacks,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2019, pp. 10 408–10 418.

[25] C. Xiao, J. Zhu, B. Li, W. He, M. Liu, and D. Song, “Spatially transformed adversarial examples,” in Proc. 6th Int. Conf. Learn. Representations, Vancouver, BC, Canada, Apr./May 30–03, 2018.

[26] H. Hosseini, B. Xiao, M. Jaiswal, and R. Poovendran, “On the limitation of convolutional neural networks in recognizing negative images,” in Proc. 16th Int. Conf. Mach. Learn. Appl., 2017, pp. 552–538.

[27] S. Samanta and S. Mehta, “Towards crafting text adversarial samples,” 2017, arXiv:1707.02812.

[28] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, “Black-box generation of adversarial text sequences to evade deep learning classifiers,” in Proc. IEEE Secur. Privacy Workshops, 2018, pp. 50–56.

[29] F. Tramèr, N. Carlini, W. Brendel, and A. Madry, “On adaptive attacks to adversarial example defenses,” 2020, arXiv:2002.08347.

[30] S. Zheng, Y. Song, T. Leung, and I. J. Goodfellow, “Improving the robustness of deep neural networks via stability training,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Las Vegas, NV, USA, Jun. 27–30, 2016, pp. 4480–4488.

[31] A. Athalye, N. Carlini, and D. Wagner, “Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples,” in Proc. 35th Int. Conf. Mach. Learn., J. Dy and A. Krause, Eds., Stockholmsmässan, Stockholm Sweden, Jul. 10–15, 2018, pp. 274–283.

[32] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a defense to adversarial perturbations against deep neural networks,” in Proc. IEEE Symp. Secur. Privacy, 2016, pp. 582–597.

[33] F. Tramèr, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel, “Ensemble adversarial training: Attacks and defenses,” in Proc. 6th Int. Conf. Learn. Representations, Vancouver, BC, Canada, Apr./May 30–03, 2018.

[34] A. Kurakin, I. J. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” in Proc. 5th Int. Conf. Learn. Representations, Toulon, France, Apr. 24–26, 2017.

[35] J. Paparrizos and M. J. Franklin, “GRAIL: Efficient time-series representation learning,” Proc. VLDB Endowment, vol. 12, pp. 1762–1777, 2019.

[36] A. Athalye, N. Carlini, and D. Wagner, “Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples,” in Proc. 35th Int. Conf. Mach. Learn., J. Dy and A. Krause, Eds., Stockholmsmässan, Stockholm Sweden, Jul. 10–15, 2018, pp. 274–283.

[37] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a defense to adversarial perturbations against deep neural networks,” in Proc. IEEE Symp. Secur. Privacy, 2016, pp. 582–597.

[38] F. Tramèr, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel, “Ensemble adversarial training: Attacks and defenses,” in Proc. 6th Int. Conf. Learn. Representations, Vancouver, BC, Canada, Apr./May 30–03, 2018.

[39] A. Kurakin, I. J. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” in Proc. 5th Int. Conf. Learn. Representations, Toulon, France, Apr. 24–26, 2017.

[40] H. A. Dau et al., “Optimizing dynamic time warping’s window width for time series data mining applications,” in Proc. VLDB Endowment, vol. 12, pp. 1762–1777, 2019.

[41] A. Bagnall, J. Lines, W. Vickers, and E. Keogh, “The UEA & UCR time series classification comp,” 2018, arXiv:2003.11246.

[42] T. Belkhouja and J. R. Doppa, “Adversarial framework with certified robustness for time-series domain via statistical features,” J. Artif. Intell. Res. (JAIR), vol. 73, pp. 1435–1471, 2022.

[43] D. Hussein, T. Belkhouja, G. Bhat, and J. R. Doppa, “Reliable machine learning for wearable activity monitoring: Novel algorithms and theoretical guarantees,” in Proc. 41st Int. Conf. Comput.-Aided Des. (ICCAD), 2022.

[44] T. Belkhouja, Y. Yan, and J. R. Doppa, “Training robust deep models for time-series domain: Novel algorithms and theoretical analysis,” in Proc. 36th AAAI Conf. Artif. Intell., 2022, pp. 6055–6063.

Taha Belkhouja (Student Member, IEEE) is currently working toward the PhD degree in computer science with Washington State University. His general research interest includes robust and trustworthy machine learning for safe deployment in real-world applications. His current research focuses on robustness of machine learning models for time-series domain and theoretically sound uncertainty quantification. He won a Outstanding Teaching Assistant Award from the College of Engineering, Washington State University.

Yan Yan (Member, IEEE) received the PhD degree in information systems from the University of Technology Sydney. He is an assistant professor of computer science with Washington State University. His current research interest includes developing robust machine learning systems and developing sample-efficient learning algorithms. He has published papers in top-tier conferences including NeurIPS, AAAI, IJCAI, ICCV, CVPR, and ECCV.

Janardhan Rao Doppa (Member, IEEE) received the PhD degree in computer science from Oregon State University. He is the Huie-Rogers Endowed chair associate professor with Washington State University. His research interests include both foundations of machine learning and applications to real-world problems. He won NSF CAREER award, Outstanding Paper Award from AAAI conference (2013), Best Paper Award from 'emph(ACM Transactions on Design Automation of Electronic Systems) (2021), IJCAI Early Career Award (2021), Best Paper Award from Embedded Systems Week Conference (2022), Outstanding Junior Faculty in Research Award (2020) and ReidMiller Teaching Excellence Award (2018) from the College of Engineering, Washington State University.