Recurrence-based Disentanglement for Detecting Adversarial Attacks on Timeseries Classifiers

Mubarak G. Abdu-Aguye
Mohamed Bin Zayed University of Artificial Intelligence
Abu Dhabi, UAE
Mubarak.Abdu-Aguye@mbzuai.ac.ae

Karthik Nandakumar
Mohamed Bin Zayed University of Artificial Intelligence
Abu Dhabi, UAE
Karthik.Nandakumar@mbzuai.ac.ae

***Abstract*—Time series classifiers based on deep neural networks (DNNs) are highly vulnerable to carefully crafted perturbations called adversarial attacks, which are capable of completely degrading their accuracy. The primary challenge in detecting such adversarial samples is the difficulty in disentangling the underlying signal from the added perturbations. In this work, we propose a novel technique for detecting adversarial attacks against deep timeseries classifiers. Firstly, we show that a recurrence plot (RP) representation can effectively disentangle adversarial perturbations in time series data as local artifacts in the image domain. Secondly, we demonstrate that these artifacts can be easily amplified or suppressed using image morphological operations, without impacting the true signal information. Consequently, the distributions of RP features (before and after morphological operations) do not change for benign samples, while they begin to diverge for adversarial samples. Finally, we train a normalcy model to encode the distribution of RP features of benign samples and employ outlier detection in the parameter space to detect adversarial samples. Evaluations based on four adversarial attacks (FGSM, BIM, MIM and PGD) and on all 85 datasets in the 2015 UCR TS archive, show that the proposed method outperforms the state-of-the-art and is 3.65× faster on average.**

I. Introduction

Deep learning models for computer vision are vulnerable to specially-crafted perturbations that interfere with their operation [1]. Such perturbations are called *adversarial attacks* and can completely degrade the accuracy of such models. It has also been established that attacks crafted for some model may also significantly affect other unseen models [2]. Recently, adversarial attacks were also shown to exist for deep timeseries classification models [3]. Because of their transferability and accuracy degradation, adversarial attacks are a clear danger to virtually all DNN models, especially those used in sensitive applications. Furthermore, this danger is more acute for non-visual modalities such as timeseries signals, which are natively hard to interpret and therefore easier to corrupt. Hence, there is a strong need for defense mechanisms against adversarial attacks in the timeseries domain.

Though a multitude of methods exist for timeseries analysis including statistical (e.g., moment estimation), time-domain (e.g., autoregressive modelling) and frequency-domain techniques (e.g., the Fourier transform), these approaches fail to detect adversarial samples because of the infinitesimal magnitude of the perturbations. However, the fact that such seemingly-insignificant perturbations can affect deep classifiers even in transfer scenarios intuitively suggests that they affect some fundamental property/dimension of the data itself. This motivates further investigation into other timeseries analysis techniques in a bid to identify a suitable domain for adversarial detection.

One common characteristic of most timeseries data is the existence of recurrence relationships, which can be represented using a recurrence plot [4]. In this work, we propose a novel approach for detecting adversarial attacks against deep timeseries classifiers using recurrence plots. Firstly, we show that adversarial perturbations get disentangled as local artifacts in the recurrence plots and these artifacts can be directly amplified or suppressed in the image domain. Secondly, we build a normalcy model based on the distributions of the recurrence plots of benign samples and use an outlier detector for adversarial sample detection.

II. Preliminaries and Related Work

a) Adversarial Attacks on Timeseries Data: Let a timeseries sample of length T be denoted as $\mathbf{x}$, such that $\mathbf{x}(t)$ gives the value of the timeseries at some timestep $t \in [1 \cdots T]$. In deep timeseries classification, this sample is fed into classifier f parameterized by weights θ to produce a predicted label $\hat{y}$ i.e. $\hat{y} = f_\theta(\mathbf{x})$. An adversary generates an adversarial perturbation η, which is computed using some attack technique (e.g. FGSM [1]), and adds it pointwise to the given signal (i.e. $\hat{\mathbf{x}}(t) = \mathbf{x}(t) + \eta(t)$) to obtain its adversarial counterpart $\hat{\mathbf{x}}$ [3]. The magnitude of the adversarial perturbation is usually bounded under some norm p in order to keep it as imperceptible as possible i.e. $||\eta||_p \leq \epsilon$, where ϵ represents the maximum magnitude of the perturbation. Note that adding the perturbation η causes misclassification i.e. $f_\theta(\mathbf{x}) \neq f_\theta(\hat{\mathbf{x}})$.

b) Adversarial Attack Detection in Timeseries Data: The goal of adversarial attack detection is to determine if the given signal has been adversarially perturbed or not. To the best of our knowledge, there are only two approaches for detecting adversarial attacks against deep timeseries classifiers. [5] formulated the problem as an instance of *unsupervised* outlier detection. Their approach involved computing complexity and chaotic measures of adversarially-unperturbed samples, then learning a normalcy model over such features. The learned model was then subsequently used to distinguish between

ISBN: 978-9-4645-9360-0

normal and adversarial samples. In an evaluation involving 85 datasets and two attack methods, they obtain results on 72 datasets and report detection accuracies reaching 97%. The other approach was proposed by [6], where they considered the adversarial detection problem in 3 *supervised* settings: binary classification (i.e. normal vs adversarial), 2n-classification (i.e. n normal and n adversarial) and an ensemble approach combining the two. They also evaluate their method on the same datasets, reporting higher accuracy in most cases.

c) Recurrence Plots for Timeseries Analysis: In the timeseries domain, recurrence plots have also been used for timeseries classification in [7]. Their approach involves converting timeseries data into recurrence plots, and using a nearest-neighbor scheme involving a compression-based distance, outperform Euclidean Distance and Dynamic Time Warping. They further investigate the use of traditional texture extraction techniques on recurrence plots in [8] and obtain results outperforming comparable timeseries classification techniques. Similarly, [9], [10] have applied convolutional neural networks to recurrence plots also for classification purposes. These works report generally improved accuracy over some state of the art baselines, albeit at additional computational cost. *To the best of our knowledge, our proposed approach is the first ever attempt to use recurrence relationships for the detection of adversarial attacks in the timeseries domain.*

III. Proposed Approach

The first step of the proposed approach is to disentangle a given timeseries sample into signal and perturbation components. Based on the assumption that the timeseries sample $\mathbf{x}$ contains some underlying structure, we consider recurrence plot (RP) [4] as our representation method due to its simplicity and widespread amenability. Moreover, a RP directly provides a semi-interpretable visualization of the underlying structure in the timeseries. Therefore, an arbitrary timeseries $\mathbf{x}$ can be mapped to a 2-dimensional recurrence plot $[R]_{T\times T}$ via the following relation:

$$R = ||\mathbf{x}_T(t) - \mathbf{x}_T(u)||. \tag{1}$$

The above equation is evaluated for all timesteps $(t, u) \in [1 \cdots T]$. As shown in Figure 1, adversarial perturbations show up as small local artifacts in a recurrence plots.

For adversarial sample detection, the perturbation component η is of primary interest. Specifically, we propose two abstract operations to amplify it and suppress it. These operations can be expressed concisely as:

$$\hat{\mathbf{x}} = \mathbf{x} + \alpha.\eta, \tag{2}$$

where the amplification operation is analogous to setting α greater than 1 and the suppression operation sets α to less than 1. In the case of a benign sample (which lacks any perturbation), these operations have virtually no effect on the sample. Conversely, for adversarial samples, the amplification operation will enhance the perturbation, making it more observable/perceptible while the suppression operation will make it more stealthy. Therefore, analyzing the output of these two operations can differentiate between a benign sample $\mathbf{x}$ and an adversarial sample $\hat{\mathbf{x}}$. However, the design of such amplification and suppression operators must be such that they do not negatively impact the legitimate information in the sample, but rather only affect the adversarial perturbation/noise. Although some candidate functions exist for this purpose in the timeseries domain (e.g. low-pass filters), they must be specifically designed and tuned to guarantee that they do not harm the underlying information in the timeseries.

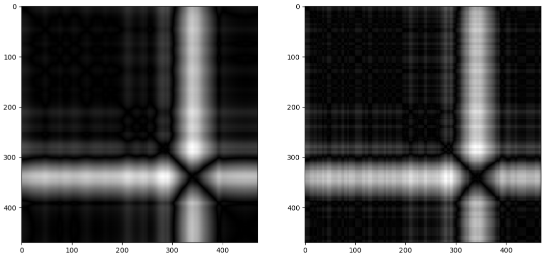


Fig. 1: Recurrence plots of unperturbed sample (left) vs perturbed sample (right). The perturbed sample clearly shows small local artifacts representing adversarial perturbations.

Since the recurrence plot is two-dimensional, it can therefore be treated as an image. This also enables the use of image processing operations to implement the suppression and amplification operators. In particular, morphological opening and closing are designed to de-emphasize or amplify small artifacts in images, directly corresponding to the functionality/behavior of the suppression and amplification operators proposed. Also note that the morphological operations have locality, which ensures that they better preserve information in local regions as opposed to filtering operations in the time domain, which are more concerned with global preservation. In this work, given some sample $\mathbf{x}$, we compute its recurrence plot R and obtain two variants R^O and R^C by applying morphological opening and closing on it, respectively.

The recurrence plots of benign samples can be considered as samples drawn from distribution f_R i.e. $R \sim f_R$. For the adversarial counterpart of $\mathbf{x}$ i.e. $\hat{\mathbf{x}}$, its resulting recurrence plot $\hat{R}$ will have a slightly different distribution $\hat{f_R}$ due to the presence of the adversarial perturbation i.e. $\hat{R} \sim \hat{f_R}$. Moreover, this difference can be (de)emphasized through the use of the suppression and amplification operators described previously. To model f_R, we select the generalized Gaussian distribution (GGD) due to its simplicity (e.g. it is parameterized by just two values: mean β and variance γ) and flexibility. However, we first preprocess the recurrence plot by computing mean-subtracted contrast-normalized (MSCN) coefficients from it. The MSCN coefficients better capture the behavior of the plot in different local regions and are known to correlate to perceptual quality [11], [12]. For pixel position (t, u) in R, the MSCN coefficients are computed as:

$$R_p(t,u) = \frac{R(t,u) - \mu(t,u)}{\sigma(t,u) + C}, \tag{3}$$

where $R(t,u)$ is the recurrence plot value at position (t,u) and C = 1 to prevent numerical degeneracy. The local mean μ and local contrast σ are given as:

$$\begin{aligned} \mu(t,u) &= \sum_{k=-k}^{K} \sum_{l=-L}^{L} w_{k,l} I_{k,l}(t,u) \\ \sigma(t,u) &= \sqrt{\sum_{k=-k}^{K} \sum_{l=-L}^{L} w_{k,l}(I_{k,l}(t,u) - \mu(t,u))^2} \end{aligned} \tag{4}$$

where $w = \{w_{k,l} | k = -K, \dots, K, l = -L, \dots, L\}$ is a 7x7 2D Gaussian function rescaled to unit volume.

Next, we fit a GGD model to the preprocessed recurrence plot R_p, yielding mean β and variance γ, representing f_R. We also perform feature extraction on the two variants R^O and R^C as well, yielding a total of 6 features i.e. $[\beta, \gamma, \beta^O, \gamma^O, \beta^C, \gamma^C]$. We adopt this approach rather than simply taking the differences between the parameters because different datasets behave differently under the amplification and suppression operators. By using the raw parameters directly, the downstream method learns dataset-specific behaviors, allowing for better adaptation. It must be emphasized that the proposed approach can also extend to the multivariate case i.e. each axis/variable can be considered separately for feature extraction, after which all the features can be concatenated into a single vector for later usage.

In practice, adversarial examples will not usually be available to users. Hence, we adopt a similar methodology to [5] and frame our adversarial detection task as an instance of outlier detection. Therefore, we perform feature extraction over *benign* samples, and then use the derived features to train the normalcy model. This model will then be used to distinguish between normal and adversarial samples subsequently. We adopt the One-Class Support Vector Machine [13] as our normalcy model due to its efficacy and low data requirements.

IV. Experiments, Results and Discussion

A. Experimental Methodology: We consider our method in a gray-box setting i.e. under the assumption that the attacker is unaware of our detection technique. For fair comparison, we select [5] as our baseline since their method is the state of the art under the outlier detection paradigm, and is also evaluated in a gray-box setting. We therefore evaluate the performance of our method on the same datasets (i.e. all 85 datasets in the 2015 UCR Time Series Archive [14] and their adversarially-perturbed versions generated by [3]). The adversarially-perturbed datasets were generated for a ResNet classifier using L_∞-norm constrained FGSM and BIM [15] attacks, with the maximum attack magnitude ϵ set to 0.1. For the BIM attack, it was run for 10 iterations with a step size (α) of 0.05. Since we use the same type of normalcy model as in [5], we use an identical configuration i.e. we set its contamination parameter to 0.1, use a radial-basis function (RBF) kernel and compute the kernel coefficient in a variance-aware manner. We consider detection accuracy as the performance metric as in the baseline, and evaluate our method in the same experimental settings:

1) Normal + FGSM: In this scenario, equal numbers of previously-unseen normal samples and FGSM-perturbed samples are fed to the normalcy model. This scenario measures the ability of the method to distinguish between normal and FGSM-perturbed samples.
2) Normal + BIM: In this scenario, equal numbers of previously-unseen normal samples and BIM-perturbed samples are fed to the normalcy model. This scenario measures the ability of the method to distinguish between normal and BIM-perturbed samples.
3) Normal + FGSM + BIM: In this scenario, equal numbers of previously-unseen normal data, FGSM-perturbed data and BIM-perturbed data are mixed, forming a combination that is one-third normal and two-thirds adversarial. This scenario measures the ability of the model to truly characterize the normal data.

B. Evaluation Metrics: We evaluate the performance of the proposed technique via statistical hypothesis testing. We adopt the framework presented in [16], which involves a Friedman test to confirm that the two methods have statistically-different performances, followed by post-hoc analysis via a Wilcoxon Signed Rank test to rank their (relative) performances. As the Friedman test needs a minimum of 3 candidates, we include a random guessing baseline as was used in [5]. We present the results as a critical difference diagram showing the ranking of each method relative to the competing methods. The lower the rank of a method, the higher its performance.

We also evaluate the speed of the proposed approach relative to the baseline, in order to illustrate the practical feasibility of our technique. We focus on the feature extraction time per sample since inference/classification times are generally assumed to be insignificant. We compute the feature extraction time by randomly selecting a fixed number of samples (20), timing the feature extraction process across the chosen samples and computing the average feature extraction time per sample. We repeat this procedure five times per dataset, and compute the average time taken across these five trials. This experiment is carried out on a consumer-grade laptop with a dual-core 2.1GHz processor and 12GB of RAM.

C. Results: The resulting critical difference diagram is shown in Figure 2. It can be seen that the proposed technique significantly outperforms both baselines. Notably, it works on all 85 datasets. This is in contrast to the baseline, which failed (i.e. produced degenerate values as features) on 13 datasets. This further confirms the superiority and general applicability of the proposed method relative to the baseline.

We also report the results obtained from the timing experiment described in Section III. We compute speedup as the ratio of the mean per-sample feature extraction time using the base-

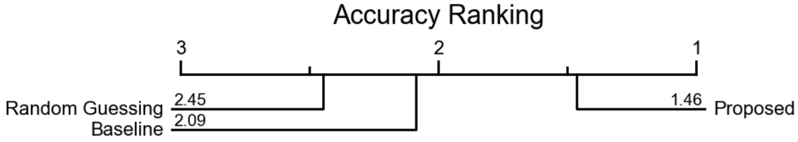


Fig. 2: Critical Difference Diagram showing rank of Proposed Method relative to Baselines on FGSM, BIM and FGSM+BIM attacks (Lower rank is better).

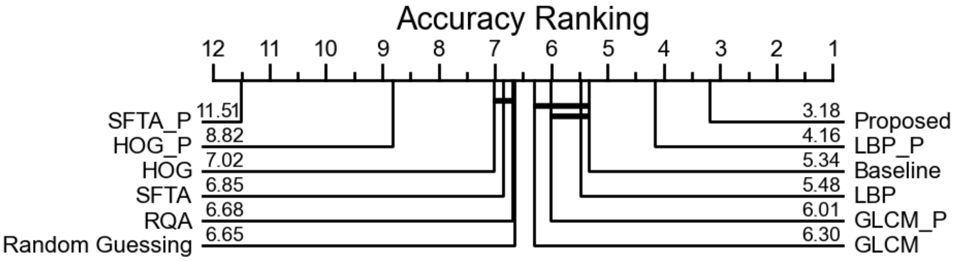


Fig. 3: Critical Difference diagram for Texture descriptors, RQA and proposed method. Suffix ”˙P” used to indicate variants with preprocessing.

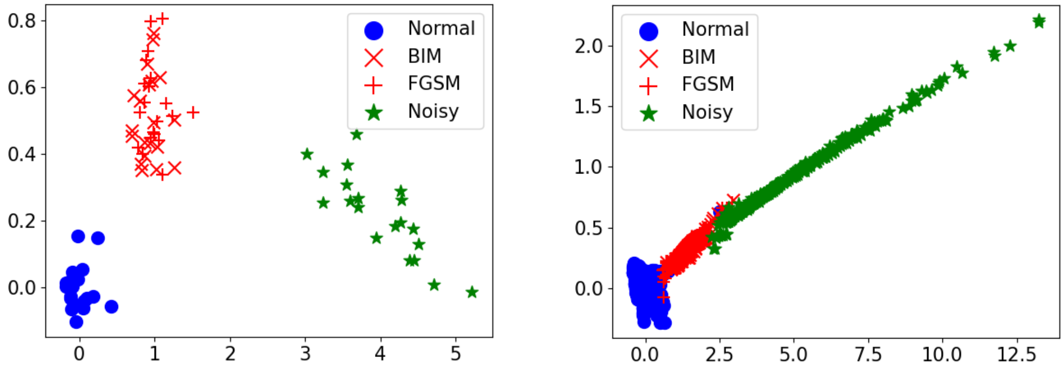
line method to the mean per-sample feature extraction time using the proposed method. The proposed method achieves up to a 7× speedup in the best case. The obtained speedup reduces with increased sample lengths, since the physical size of the recurrence plot grows as the square of the sample length. For the longest timeseries considered (2709 timesteps), the speedup drops to around 0.62×. On average, the proposed method delivers a **3.65×** speedup relative to the baseline method. Further optimizations are possible e.g. computing the features for the three recurrence plots in parallel. This shows the superior feasibility of our method in real-world scenarios, which are often time- or compute-constrained.

D. Comparative Studies: Since the recurrence plot can also be considered as a 2D texture pattern, we compare our method against four common texture descriptors: Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), Segmentation-based Fractal Texture Analysis (SFTA) and Gray-Level Co-occurence Matrix (GLCM), as well as a suite of 13 measures derived from recurrence quantification analysis (RQA). For the former set, we evaluate the recurrence plot alone, and the recurrence plot together with its preprocessed versions (as described above). As seen from the results (Figure 3), the proposed method outperforms all 9 comparative methods, justifying the proposed approach. Notably, some of the texture-based methods compete with and in one case, outperform the current state of the art method [5]. This can be attributed to their use of a two-dimensional representation (which is more informative) rather than the one-dimensional representation used in [5].

E. Adversarial Detection Fidelity: A natural question regarding the proposed method is its ability to distinguish between genuine adversarial attacks and simple random noise. We posit that random noise comes with a different appearance (in the recurrence plots) than adversarial attacks, and is therefore distinguishable by our method. To confirm this, we carry out the following experiment: given some dataset, we consider the samples in its testing portion and their adversarially-perturbed counterparts. We then generate noisy versions of the benign samples by adding Additive White Gaussian Noise (AWGN) at the same signal-to-noise ratio as the adversarially-perturbed samples (i.e. 20dB). This yields three sets of samples: benign, adversarial and noisy samples. We then perform feature extraction as usual on each of these sets. We plot the feature space in two dimensions using the IsoMap method [17] in order to determine whether the noisy samples can be distinguished from the adversarial (and benign) samples. The resulting plots are shown in Figures 4a and 4b (for the 2-class “BeetleFly” and 50-class “50words” datasets respectively). From the plots the noisy samples can be seen to be clearly distinct from the benign and adversarial samples, confirming the proposed method’s ability to differentiate random noise from adversarial attacks even in challenging scenarios.

F. Performance on More Advanced Attacks: We also compare the performance of the proposed method to the baseline on more advanced attacks. Specifically, we consider the Momentum Iterative Method (MIM) [18] and Projected Gradient Descent (PGD) [19] attacks, as these are considered to be more powerful than the FGSM and BIM attacks. We generate MIM and PGD attacks for each dataset using the same parameters as those in the original set (i.e. $\epsilon = 0.1$, 10 iterations). However, to make the attacks even more subtle, we adjust the step size (α) to 0.01 rather than 0.05 as used by [3]. In practice, we generate attacks for each dataset five (5) times, and run the detectors on each set of generated samples. Finally, we report the detectors’ performances averaged over these 5 iterations/experiments.

The resulting critical difference diagram is shown in Figure 5. Once again, the proposed method maintains its superiority



(a)

(b)

Fig. 4: (a) shows the feature space of normal, adversarial and noisy samples for the 2-class BeetleFly dataset (b) 50-class FiftyWords dataset. Noisy samples (green stars) are clearly distinct from normal samples (blue dots) and adversarial samples (red crosses) (best viewed in color).

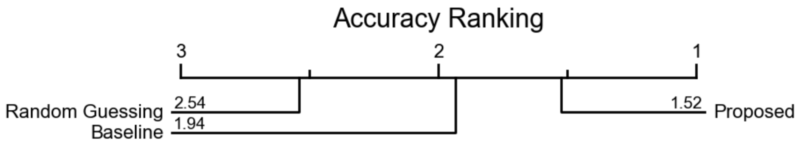


Fig. 5: Critical Difference diagram showing proposed method's performance on MIM and PGD attacks.

even in this more challenging setting.

V. Conclusion & Future Work

We propose a novel method for detecting adversarial attacks against deep timeseries classifiers, based on modeling (the appearance of) recurrence plots. We carried out extensive experiments involving 4 attacks and a large suite of diverse datasets to evaluate its detection efficacy and speed of operation. We obtained results confirming the superiority of the proposed method relative to the state of the art under both metrics.

In future work, we intend to evaluate its efficacy in detecting more subtle attacks (e.g the Carlini-Wagner attack) in both gray-box and defense-aware settings.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[2] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

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

[4] J.-P. Eckmann, S. O. Kamphorst, D. Ruelle, *et al.*, "Recurrence plots of dynamical systems," *World Scientific Series on Nonlinear Science Series A*, vol. 16, pp. 441–446, 1995.

[5] M. G. Abdu-Aguye, W. Gomaa, Y. Makihara, and Y. Yagi, "Detecting adversarial attacks in time-series data," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3092–3096, IEEE, 2020.

[6] J. Teraoka and K. Tamura, "Detecting adversarial examples for time series classification and its performance evaluation," in *Intelligent Decision Technologies*, pp. 569–581, Springer, 2021.

[7] D. F. Silva, V. M. D. Souza, and G. E. Batista, "Time series classification using compression distance of recurrence plots," in *2013 IEEE 13th International Conference on Data Mining*, pp. 687–696, 2013.

[8] V. M. Souza, D. F. Silva, and G. E. Batista, "Extracting texture features for time series classification," in *2014 22nd International Conference on Pattern Recognition*, pp. 1425–1430, 2014.

[9] Y. Zhang, Y. Hou, S. Zhou, and K. Ouyang, "Encoding time series as multi-scale signed recurrence plots for classification using fully convolutional networks," *Sensors*, vol. 20, no. 14, p. 3818, 2020.

[10] N. Hatami, Y. Gavet, and J. Debayle, "Bag of recurrence patterns representation for time-series classification," *Pattern Analysis and Applications*, vol. 22, no. 3, pp. 877–887, 2019.

[11] W. S. Geisler, "Visual perception and the statistical properties of natural scenes," *Annu. Rev. Psychol.*, vol. 59, pp. 167–192, 2008.

[12] A. Galdran, T. Araújo, A. M. Mendonça, and A. Campilho, "Retinal image quality assessment by mean-subtracted contrast-normalized coefficients," in *European Congress on Computational Methods in Applied Sciences and Engineering*, pp. 844–853, Springer, 2017.

[13] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[14] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The ucr time series classification archive," 2015.

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

[16] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data mining and knowledge discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[17] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[18] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193, 2018.

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