Improving the Robustness of Time Series Neural Networks from Adversarial Attacks Using Time Warping.*

Yoh Yamashita and Brian Kenji Iwana[0000−0002−5146−6818]

Graduate School of Information Science and Electrical Engineering
Kyushu University, Fukuoka, Japan
yoh.yamashita@human.ait.kyushu-u.ac.jp, iwana@ait.kyushu-u.ac.jp

Abstract. Time series neural networks have been shown to be weak against adversarial attacks. This study aims to enhance the robustness of time series neural networks in order to defend against such attacks. To do so, we introduce a new defense method called a Random Warping Self-Ensemble (RWSE). The RWSE has two main components. First, a novel random time warping layer to add randomness to trained models in order to disrupt the adversarial attack. Second, the use of self-ensembling increases robustness and maintains the accuracy of the network. The proposed RWSE does not require any special or extra training, can be used with most time series neural networks, including already trained ones, and does not require any extra trainable parameters. We demonstrate that the RWSE is effective in helping reduce the effects of four gradient-based adversarial attacks on five time series datasets.

Keywords: Adversarial Attacks · Robust Neural Networks · Time Warping.

1 Introduction

Deep neural networks have had many successes in pattern recognition and classification [29], including time series recognition and classification [35, 2], forecasting [33], and signal processing [20]. However, recent research has shown that neural networks are weak against adversarial examples [32], including time series [12]. Adversarial examples are patterns that would normally be correctly classified by a trained recognition system but are subjected to adversarial attacks which intentionally cause misclassifications. These adversarial attack algorithms create such adversarial examples by adding adversarial noise or perturbations that are typically hardly perceivable to humans. Since the presence of an attack is difficult to detect, there are security concerns for the use of neural networks in security-sensitive systems [23].

In response to adversarial attacks on time series neural networks, various defense mechanisms have been proposed. These algorithms aim to automatically

* This work was partially supported by MEXT-Japan (Grant No. 23K16949).

protect neural networks from such attacks, regardless of whether there is knowledge of an attack or not. One prevalent approach involves employing ensemble networks as a defense strategy [31, 15]. However, utilizing ensemble networks often demands multiple trained models, which might pose constraints on resource-limited systems and may not be universally applicable across all network architectures. Therefore, methods such as Random Self-Ensemble (RSE) [25] have been proposed to create ensembles but using only one network. To do this, RSE uses noise layers so that the output of the network changes and can be ensembled.

However, most defense methods, including RSE, are proposed for image-based neural networks. Little research exists on adversarial attack and defense algorithms for time series neural networks [12]. Time series and sequence recognition are important due to their wide range of applications, such as signals, biometrics, speech, etc. Accordingly, it is important to ensure the robustness of time series recognition models.

In this paper, we propose a new defense algorithm, specifically suitable for time series called Random Warp Self-Ensemble (RWSE). Given adversarial examples created from adversarial perturbation, the proposed RWSE can add robustness to trained models. Specifically, we introduce a new time warping-based layer that adds randomness to the features by warping the time steps of the internal representation of each input. Not only does this remove some of the effects of the specifically designed adversarial noise due to the randomness, but it also allows for the network to be used with self-ensembling. We show that the proposed RWSE makes temporal neural networks more robust to adversarial noise without sacrificing accuracy on non-attacked data.

Furthermore, the proposed method is only applied in the test step. It does not require specific training and can be used for any trained neural network with a fully connected layer. It also does not increase the number of trainable parameters. Thus, the proposed method can be widely applicable to time series neural networks.

The contributions of this paper are as follows.

- We propose a new warping layer that randomly warps the time steps of the internal representation within a time series neural network. The warping layer is only used during the test time and can be widely used with many neural networks, including already trained networks.
- We use the proposed random warping layer to add randomness for a self-ensemble network. The randomness allows for different outputs to be predicted given the same network with the same weights.
- We demonstrate that the proposed RWSE outperforms similar defense algorithms on four time series datasets. The RWSE is able to protect a temporal Convolutional Neural Network (CNN) from well-known adversarial attacks.
- The code for the proposed method can be found at https://github.com/uchidalab/random-warping-self-ensemble.

2 Related Work

2.1 Adversarial Attacks

Most of the research on adversarial attacks has focused on image recognition. Szegedy et al. [32] demonstrated the vulnerability of well-known image benchmarks to adversarial attacks. They generated adversarial examples by minimizing the perturbation needed to misclassify images. Similarly, DeepFool [27] perturbs the image toward the nearest class of hyperplanes. Other methods include using gradient information, such as the Fast Gradient Sign Method (FGSM) [14], Basic Iterative Method (BIM) [22], and Projected Gradient Descent (PGD) [26]. There are also ensemble attacks such as AutoAttack [9] that use multiple attack methods.

However, there is much less research on adversarial attacks in the field of time series recognition. Carlini et al. [4] showed that it is possible to encode hidden commands in speech recognition systems. Fawaz et al. [12] assess the performance of FGSM and BIM techniques using the time series classification datasets. There has also been work to create adversarial examples in time series classification without neural networks [28].

2.2 Defense Against Attacks

Defense algorithms are designed to enhance the robustness of neural networks against attackers. There are various methods for defending against adversarial attacks, each with differing levels of success [23]. For instance, training with adversarial examples can improve robustness [14]. Another approach is to limit the impact of adversarial perturbations by using defensive techniques such as defense distillation [16], feature squeezing [36], and denoising [24]. Additionally, employing ensembles and modular networks can be effective in avoiding attacks trained for specific gradients [31, 15, 25, 37].

3 Adversarial Attacks

3.1 Threat Model

This paper focuses on *white-box* attacks, or attacks with full knowledge of the trained model. Specifically, given a neural network model $f(\mathbf{x})$ with input $\mathbf{x}$, a white-box attack has access to all information about $f(\mathbf{x})$, including its parameters, gradients, etc.

The objective of the attack is to find an adversarial sample $\mathbf{x}_{adv}$ that is similar to $\mathbf{x}$ but is misclassified by $f(\mathbf{x})$. Furthermore, the similarity between $\mathbf{x}_{adv}$ and $\mathbf{x}$ must be within a budget ϵ, in order to be undetectable by an untrained eye.

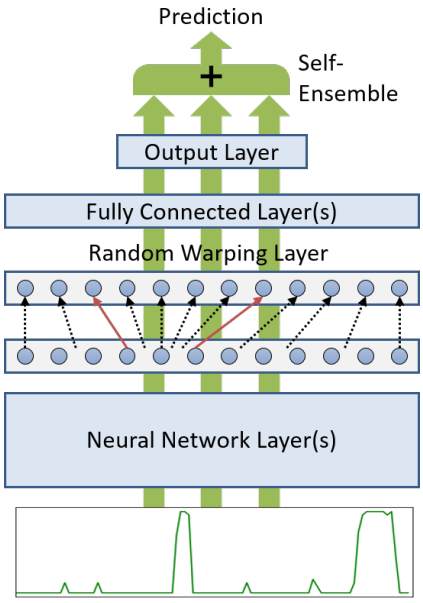

Fig. 1. A diagram of the proposed method. In this example, the input is fed to the network three times, and each time the random warping layer warps the time steps differently. The output of each prediction added to form the self-ensemble. The red arrows in the random warping layer represent the knots.

3.2 Gradient-based Adversarial Attacks

A gradient-based attack is a type of adversarial attack on neural networks where the attacker manipulates the model's input data by adding a small amount of carefully crafted adversarial noise. The noise is created by utilizing information about the gradient of the model's loss function.

In our experiments, we use the five most popular gradient-based attacks: FGSM [14], BIM [22], PGD [26], CW [5], and AutoAttack [9]. FGSM perturbs the input data in the direction of the sign of the gradient of the loss function with respect to the input. BIM takes this idea and uses FGSM in an iterative manner. PGD is similar to BIM, but it aims to minimize perceptibility and CW formulates the adversarial noise as an optimization problem between moving the classification and ensuring the perturbed noise is within a constraint. The CW attack is generally respected as one of the most difficult attacks for neural networks [25].

Finally, the last adversarial attack used is AutoAttack. AutoAttack uses an ensemble of four attacks including, two different Auto-PGD methods, Square Attack [1], and Fast Adaptive Boundary (FAB) attack [8].

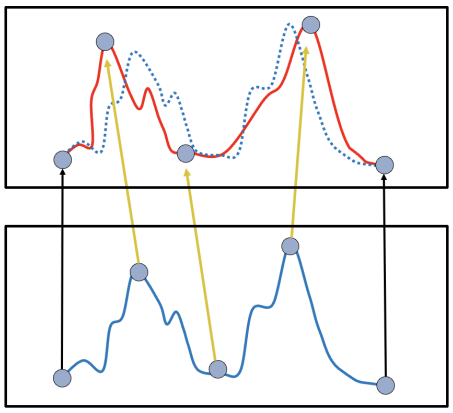

Fig. 2. Processing image of time warp layer.

4 Random Warping Self-Ensemble (RWSE)

The proposed method has two main components. The first is the use of a proposed random warping layer. The random warping layer is a layer that can be included in the structure of a trained neural network. The purpose of the layer is to randomly warp the time steps of the internal representation. Second, we use networks with the random warping layer in a self-ensemble [25]. An overview of the proposed model is shown in Fig. 1.

4.1 Random Warping Layer

Since the adversarial perturbation added to $\mathbf{x}$ is based on the specific gradient of the loss given $\mathbf{x}$, we propose a nonlinear random time warping layer in an attempt to reduce the impact of the adversarial perturbations. The idea is that the adversarial perturbations are specific to the gradients of the network. By warping the internal representation, it is possible to disturb the alignment between the perturbations in the input and the attacked gradients. At the same time, we aim to only warp in the time dimension enough to disturb the alignment without affecting the overall accuracy.

The proposed random warping layer is based on the time warping data augmentation technique [34], but unlike data augmentation, the time warping occurs within the first fully-connected layer of the network. The input features of the random warping layer are treated as a time series and a time warping algorithm is used to warp the time steps based on a smooth curve with I number of knots, or:

$$\mathbf{x}' = x_{\tau(1)}, \ldots, x_{\tau(t)}, \ldots, x_{\tau(T)}, \tag{1}$$

where $\tau(\cdot)$ is a warping function defined by a cubic spline $S(\mathbf{u})$ with knots $\mathbf{u} = u_1, \ldots, u_i, \ldots, u_I$. The magnitude of the knots u_i are taken from $u_i \sim \mathcal{N}(1, \sigma^2)$.

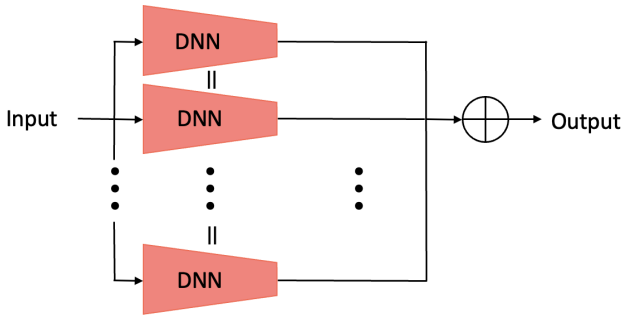

Fig. 3. A diagram of a self-ensemble method. Self-ensemble uses the same input with the same model, and ensembles the results. However, due to a random factor in the model, the output of each model is different for the ensemble.

Using the knots, the time steps $x_{\tau(t)}$ are moved. Then, the rest of the times steps are resampled based on a smooth cubic spline-based curve.

The result is a warped time series with a smooth transition between the knots as shown in the Fig. 2. This random warping layer is used between the convolution and the flattening before the first fully connected layer of the network. The purpose of the random warping is to disrupt the adversarial perturbations. In gradient-based adversarial attacks, adversarial perturbations are added to the data in the original input space. However, due to the warping within the representation, the gradients specific to the input no longer correspond directly to the input elements. Furthermore, while warping is occurring, the characteristics of the local features of original data remains intact.

During training, any standard temporal neural network can be used and the random warping layer is only used during testing. The neural network can be any architecture as long as the features are represented structurally for the fully connected layer (i.e. the model does not remove the spatial information using Global Average Pooling).

4.2 Self-Ensemble

Ensemble methods combine multiple different classifiers to form the prediction. By combining multiple predictions, the robustness can be increased over any of the single classifiers [11]. Furthermore, ensemble methods have been shown to help defend against adversarial attacks [31]. To have a meaningful ensemble, classifiers should be able to produce different predictions from each other. In order to do this, ensemble methods can be constructed using different classifiers or models [21] or models can be constructed using bagging [3].

However, using multiple models increases the computational requirements. Having multiple models means that each model must be trained and the weights

stored. Consequently, self-ensemble methods were proposed. self-ensemble methods are methods that use a single classifier but with added randomness to create distinct predictions. The previous self-ensemble methods use test time Gaussian noise within the networks [31, 25].

The random warping layer adds test time randomness which allows for the proposed method to utilize self-ensembling. This is because given the same trained network, different predictions is possible. Specifically, during testing, the same input is fed to the RWSE N number of times. Each time, the random warping layer randomly warps the time steps of the features and a slightly different prediction will be obtained.

Finally, as shown in Algorithm 1, the output of each prediction is summed, or:

$$\hat{y} = \text{argmax}\left(\sum_{n=1}^{N} f_n(\mathbf{x})\right), \tag{2}$$

where $f_n(\cdot)$ is the post-softmax output of the network and $\mathbf{x}$ is the input that may or may not be attacked. It should be noted that we use the addition of the outputs instead of voting like some ensemble methods.

Algorithm 1 Random Warping Self Ensemble

function RWSE($\mathbf{x}$)
  $sum \leftarrow 0$
  for $n = 1, N$ **do**
    $\hat{y}_n \leftarrow \text{DNN}_{\text{RW}}(\mathbf{x})$
    $sum \leftarrow sum + \hat{y}_n$
  end for
  $\hat{y} \leftarrow \text{argmax}(sum)$
  return $\hat{y}$
end function

function $\text{DNN}_{\text{RW}}(\mathbf{x})$
  $\mathbf{z} \leftarrow f(\mathbf{x})$ ▷ Feature Extraction Layers (e.g. Convolutional Layers)
  $\mathbf{u} \leftarrow u_1, \ldots, u_i, \ldots, u_I | u_i \sim \mathcal{N}(1, \sigma^2)$
  $\tau \leftarrow S(\mathbf{u})$ ▷ Cubic Spline
  $\tau' \leftarrow \text{clip}(\tau(1), \ldots, \tau(t), \ldots, \tau(T), 0, T)$
  $\mathbf{z}' \leftarrow z_{\tau'(1)}, \ldots, z_{\tau'(t)}, \ldots, z_{\tau'(T)}$
  $\mathbf{z}' \leftarrow \text{FC}(\mathbf{z}')$ ▷ Fully Connected Layers
  $\hat{y}_n \leftarrow \text{softmax}(\mathbf{z}')$
  return $\hat{y}_n$
end function

5 Experimental Results

5.1 Datasets

To evaluate the proposed method, experiments were conducted on five datasets, ElectricDevices [10], FordA [10], InsectSound [6], RightWhaleCalls [7], and FruitFlies [13]. The datasets were selected as time series datasets that had sufficiently large training sets suitable for neural networks and that represent a range of signals and time series.

Table 1. Comparison of the Datasets

Dataset	Type	Class	Length	Train	Test
ElectricDevices [10]	Device	7	96	8,926	7,711
FordA [10]	Sensor	2	500	3,601	1,320
InsectSound [6]	Audio	10	600	25,000	25,000
RightWhaleCalls [7]	Audio	2	4,000	10,934	1,962
FruitFlies [13]	Audio	3	5,000	17,259	17,259

5.2 Architecture and Settings

In the experiments, a temporal CNN is used as the backbone for defense methods. This temporal CNN consists of four 1D convolutions, each followed by batch normalization [17], rectified linear unit (ReLU) activation, and max pooling. The first block has 64 filters, and the subsequent blocks have 128 filters. Following the convolutional layers, two fully connected layers are used. The first fully connected layer comprises 512 nodes with ReLU activation, while the second serves as the output layer with a number of nodes equal to the number of classes and softmax activation. Between the two fully connected layers, dropout with a probability of 0.5 is applied.

To train all networks, we use the Adam optimizer [19] with an initial learning rate of 0.001. The network is trained for 10,000 iterations with a batch size of 256. A single CNN is trained, and six test sets (original test set, test sets with AutoAttack, FGSM, BIM, PGD, and CW attacks) are used. The trained CNN is used with defense methods only during testing.

5.3 Adversarial Attacks

For the adversarial attack methods, we used AutoAttack, FGSM, BIM, PGD, and CW. As for the hyperparameters, FGSM and BIM have a maximum distortion of $\epsilon = 0.2$, while for BIM and PGD use a step size of $\alpha = 0.05$ with $I = 10$ iterations.

Table 2. Accuracy (%) Without Attacks

Method	ElectricDevices	FordA	InsectSound	RightWhaleCalls	FruitFlies
No Defense	67.2	93.9	76.6	87.0	96.4
Input Noise	65.3	93.9	**77.2**	87.0	96.4
RSE	56.0	93.6	55.6	86.9	93.6
TTA	62.9	91.9	72.9	**87.3**	96.5
Median Filter	45.0	93.9	68.7	73.2	95.9
RW (Proposed)	**67.4**	**95.2**	74.7	85.9	96.1
RWSE (Proposed)	**67.4**	94.9	76.0	85.6	**96.7**

5.4 Defense Methods

To evaluate the proposed method, we compare it to similar methods to defend against adversarial attacks. The following models were used in the evaluation:

- *No Defense*: This trial is the original CNN that is attacked by the adversarial attacks.
- *Input Noise*: This evaluation adds Gaussian noise to the input in an attempt to cancel the adversarial noise [31]. A standard deviation of $\sigma = 0.15$ was used for the noise.
- *Random Self-Ensemble (RSE)*: Adapted from an image-based defense method, RSE [25] adds a noise layer after every convolutional layer in the CNN. For the self-ensemble, five networks are used. For the experiments, a standard deviation of $\sigma = 0.15$ was used.
- *Test Time Augmentation (TTA)*: TTA [30] is an ensemble of networks with each network using a different augmentation method during test time. To match the proposed method and RSE, five networks in the ensemble is used. The inputs of the five networks are the original time series features, jittering, magnitude warping, time warping, and window warping with the parameters suggested by Iwana and Uchida [18].
- *Median Filter*: The median filter is a smoothing filter that removes noise by taking the median of a sliding window. In the experiment, the filter size was set to 3.
- *Random Warping (RW)*: Random Warping uses the proposed random warping layer but no ensemble. This trial demonstrates the benefits of self-ensembling. The time warping path is defined by a smooth cubic spline-based curve with four knots with random magnitudes with a $\sigma = 0.2$.
- *Random Warping Self-Ensemble (RWSE)*: RWSE uses five networks with random warping layers and ensembles the results through addition.

In all of the evaluations, a normal CNN is used for training. The defense methods are only applied in the testing step.

5.5 Results

Accuracy in the no-attack case is shown in Table 2. Ideally, defense methods should not decrease the accuracy of the unmodified test set in the ideal case.

Table 3. Accuracy (%) Under Adversarial Attacks

Method	ElectricDevices	FordA	InsectSound	RightWhaleCalls	FruitFlies
Under FGSM Attack					
No Defense	46.4	59.2	50.0	22.5	66.5
Input Noise	46.5	63.7	41.2	22.9	66.7
RSE	44.7	63.7	41.2	22.9	71.6
TTA	**48.1**	63.9	52.2	44.9	74.1
Median Filter	38.3	60.5	48.7	32.5	68.8
RW (Proposed)	47.8	63.9	52.2	44.9	**76.1**
RWSE (Proposed)	**48.1**	**72.6**	**56.1**	**57.5**	75.9
Under BIM Attack					
No Defense	35.8	50.5	20.3	21.8	57.8
Input Noise	36.0	51.4	21.0	22.4	58.0
RSE	**40.8**	57.5	31.4	21.8	64.7
TTA	38.8	56.0	29.5	38.5	**68.3**
Median Filter	32.4	52.1	**35.2**	27.3	61.1
RW (Proposed)	37.5	62.9	32.4	48.0	67.9
RWSE (Proposed)	37.8	**63.6**	32.7	**50.5**	67.7
Under PGD Attack					
No Defense	33.6	50.5	18.7	22.1	57.0
Input Noise	33.9	51.1	19.4	22.3	57.1
RSE	**39.8**	61.0	30.7	21.9	63.8
TTA	37.9	56.1	28.4	38.2	**67.8**
Median Filter	30.6	52.1	**35.2**	27.4	60.5
RW (Proposed)	35.6	64.3	31.2	49.9	67.4
RWSE (Proposed)	35.7	**64.7**	31.9	**51.4**	67.3
Under CW Attack					
No Defense	33.4	12.2	30.8	26.1	6.94
Input Noise	38.8	30.6	62.0	21.0	59.3
RSE	**46.8**	72.2	42.7	60.1	73.7
TTA	46.0	56.5	58.3	73.1	81.9
Median Filter	39.9	50.2	63.4	55.2	80.8
RW (Proposed)	46.6	80.3	63.6	81.0	86.1
RWSE (Proposed)	46.6	**84.2**	**68.1**	**83.0**	**87.7**
Under AutoAttack					
No Defense	24.0	53.8	12.9	22.4	16.3
Input Noise	25.2	53.6	13.2	22.4	19.9
RSE	26.9	55.5	13.9	21.6	43.4
TTA	25.4	54.6	13.5	22.2	20.2
Median Filter	**27.3**	52.7	15.7	25.0	23.2
RW (Proposed)	24.4	56.7	16.4	**49.5**	50.4
RWSE (Proposed)	24.3	**56.8**	**16.7**	48.5	**50.9**

The experimental results show that the accuracy remains comparable to that of the CNN under normal conditions. This is a good result, because in the problem set, it is unknown whether an attack is taking place or not. Thus, it is important to maintain the accuracy on normal data.

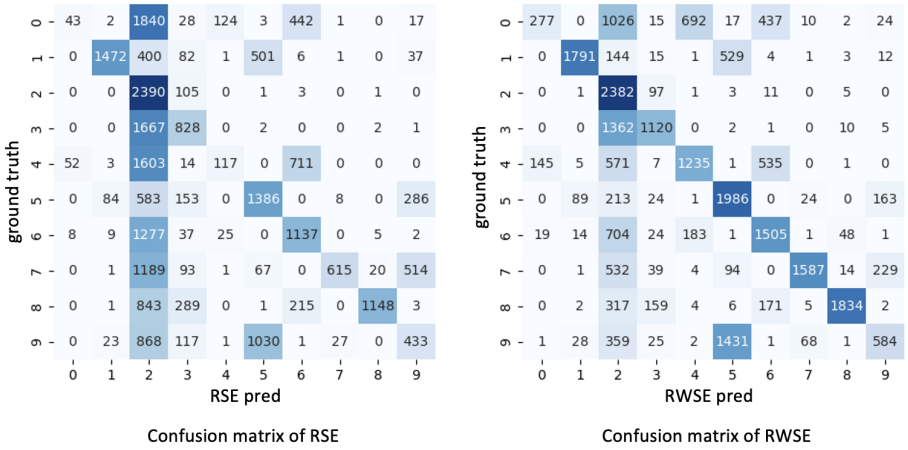

Fig. 4. Confusion matrix of RSE and RWSE for dataset InsectSound under an FGSM attack.

Next, we conduct AutoAttack, FGSM, BIM, PGD, and CW attacks, and the results are shown in Table 3. All four sections of the table have the same method, and for these sections, a robust method would close the gap between the accuracy of the CNNs under attack and those not under attack. As a result, for all attacks, all datasets show an improvement in accuracy with the proposed method over the normal case. Overall, the proposed RWSE outperforms the main competitor, RSE, for all datasets in each attack except ElectricDevices. Furthermore, RWSE outperforms RW without the ensemble. This indicates that the self-ensemble method also contributes to the improvement in robustness. Attempts to remove the adversarial noise in the input space directly, such as Input Noise and Median Filter, rarely improved accuracy by significant amounts in normal conditions. These operations proved ineffective against adversarial attacks.

5.6 Discussion

From the results shown in Tables 2 and 3, it can be concluded that the proposed RWSE is effective as a defense method against time-series data, since it can improve the accuracy against attacks while maintaining the original accuracy. In particular, the InsectSound dataset showed a significant improvement in accuracy over the no-protection condition, and was more accurate than the RSE of the competing method. The confusion matrices of RSE and the proposed method RWSE for the FGSM attack are shown in Fig. 4 in the confusion matrix of RSE, most of the model's predictions output Class 2. On the other hand, the RWSE confusion matrix shows a only slight increase in Class 2. This shows one instance that warping instead of noise is better suited at disrupting

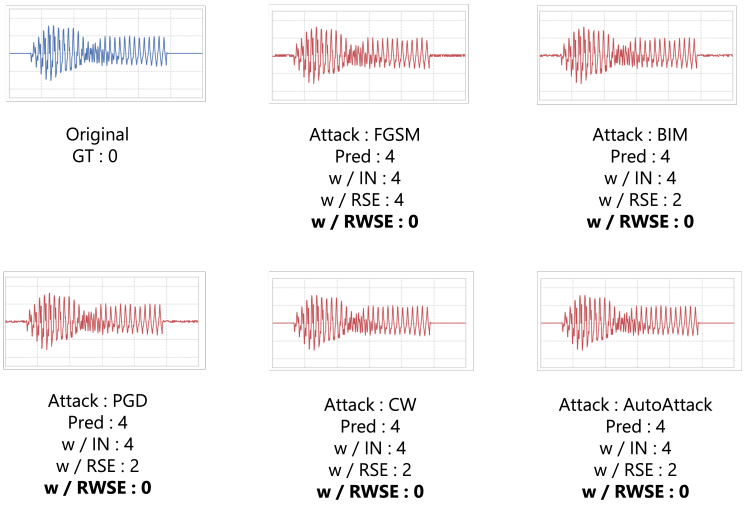

Fig. 5. Example data from InsectSound for each attack and their predictions with robustness measures.

the adversarial perturbations that would cause the network to overfit to Class 2.

Figure 5 shows the model predictions for each defense method for the attack data. In comparison methods such as Input Noise, RSE, and TTA, the misclassified results are nearly identical to the output of the no attack condition. This indicates that processing in the input space alone leaves residual effects of the attack. The importance of the warp layer inserted before the FC layer can be seen in the reduction of the effect of the perturbation caused by the attack.

Next, in order to see the difference between the proposed RWSE and the main competitor, RSE, we visualized the results by dividing them into those that could be correctly classified only by RSE, those that could be correctly classified only by RWSE, those that could be correctly classified by both, and those that could not be correctly classified by either, as shown in Fig. 6. This figure shows that the data that neither RSE nor RWSE could protect, were generally shorter regions of interest in length. On the other hand, the cases that had longer region of interests were correctly protected by the defenses. The data that could be correctly classified using only RSE tended to be relatively short in length, and the data that could be correctly classified using only RWSE included data of various lengths. This indicates that while RWSE can handle a large range of region of interests, the RSE model was able to handle smaller patterns slightly better.

5.7 Limitations

The proposed method did not work well with ElectricDevices. For ElectricDevices, the main competitor, RSE, was more accurate than RWSE for all attacks

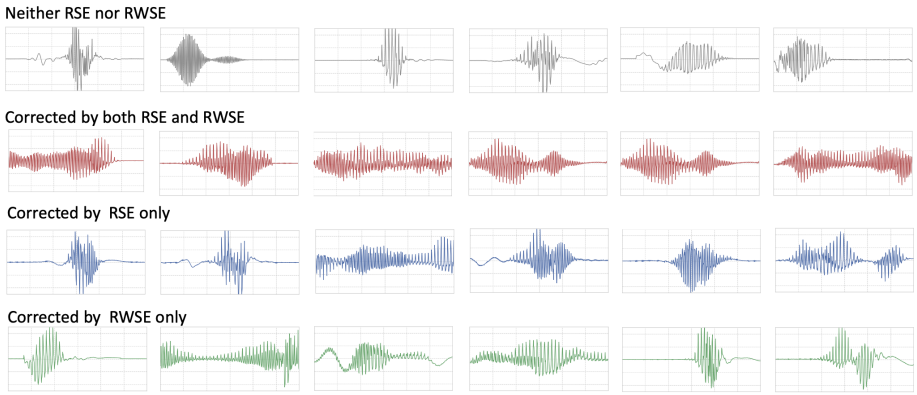

Fig. 6. Examples of data from InsectSound where models have been improved by RSE or RWSE under a CW attack.

except the FGSM attack. ElectricDevices has unusually noisy Class 2 data. This perturbation would have caused most of the data to be classified as class 2, which would have prevented the proposed method from destroying the perturbation. Another factor is the length of the data. As shown in Table 1, the ElectricDevices data set is made of time series that are shorter than the other data sets. In this experiment, all parameters in the warp layer were the same. Therefore, the same warping process used for the short data set as for the other long data sets may have resulted in a loss of original characteristics.

Adversarial attacks add noise to change the class. Therefore, the ensemble may have contributed to erroneous output as a result of a large number of data distributed near the class boundaries. By ensembling, a more generalized prediction is possible.

5.8 Ablation

In this section, we examine the effects that the parameters have on the proposed method. Specifically, we conducted experiments on the relationship between the number of networks and accuracy in self-ensemble, and on the parameters of random warping used in the proposed method.

First, we conducted experiments on the parameter σ in the time warping process. This parameter corresponds to the degree of warp movement in the time warp process. In other words, if σ is small, it is equivalent to warping to data that is close to the adversarial data, so no improvement in accuracy can be expected. Figure 7 shows the results of the experiment. For all datasets, the lowest accuracy was observed for small values of σ. After an upward trend, the accuracy decreased again for larger values of σ. Each attack had the highest accuracy when $\sigma = 0.2$ for all data sets. This indicates that $\sigma = 0.2$ is the best

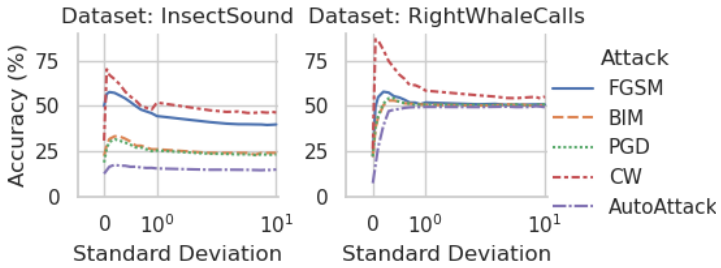

Fig. 7. Effect of the amount of time warping.

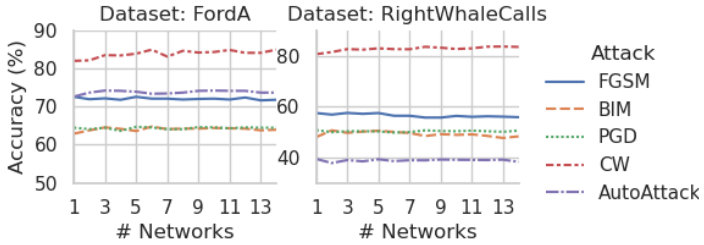

Fig. 8. Accuracy of the ensemble with different number of networks.

for the time warp used in the proposed RWSE method, regardless of the attack method.

Experiments were then conducted on the relationship between the number of networks used in the RWSE self-ensemble and accuracy. Figure 8 shows the results. The number of networks corresponds to RW when the number of networks is 1. Figure 8 shows that for each attack, the number of networks tends to increase slightly up to 5, while it does not change much for larger numbers. Although the ensemble method increased accuracy, the improvement in accuracy was small. This is because the ensemble method increases the confidence of ambiguous data, such as those at class boundaries, by majority voting. The ensemble did not function strongly because data distributed near the boundary of another class in the feature space due to adversarial attacks were pulled back to the original class domain due to the destruction of adversarial perturbations by the time warp.

6 Conclusion

In this paper, we proposed a method to improve the robustness of time series neural networks against adversarial perturbations, called a Random Warping Self-Ensemble (RWSE). The RWSE uses a self-ensemble of networks with a random warping layer. It does not require training models or parameters and can

be easily incorporated into any already trained time series neural network that contains a fully connected layer. To evaluate the proposed RWSE, we compared it to other test-time defenses against five adversarial attacks, AutoAttack, FGSM, BIM, PGD, and CW on five time series datasets.

References

1. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: ECCV. pp. 484–501 (2020)
2. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)
3. Breiman, L.: Bagging predictors. Machine learning **24**(2), 123–140 (1996)
4. Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., Zhou, W.: Hidden voice commands. In: USENIX. pp. 513–530 (2016)
5. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. Ieee (2017)
6. Chen, Y., Why, A., Batista, G., Mafra-Neto, A., Keogh, E.: Flying insect classification with inexpensive sensors. J. Insect Behav. **27**(5), 657–677 (2014). https://doi.org/10.1007/s10905-014-9454-4
7. Cornell Research Foundation, Inc.: The marinexplore and cornell university whale detection challenge (October 2022), https://www.kaggle.com/competitions/whale-detection-challenge/data
8. Croce, F., Hein, M.: Minimally distorted adversarial examples with a fast adaptive boundary attack. In: ICLR. pp. 2196–2205 (2020)
9. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: ICML. pp. 2206–2216 (2020)
10. Dau, H.A., Keogh, E., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G., Hexagon-ML: The ucr time series classification archive (October 2018), https://www.cs.ucr.edu/ẽamonn/time_series_data_2018/
11. Dietterich, T.G.: Ensemble methods in machine learning. In: IWMCS. pp. 1–15 (2000)
12. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Adversarial attacks on deep neural networks for time series classification. In: IJCNN (2019). https://doi.org/10.1109/ijcnn.2019.8851936
13. Flynn, M.: Classifying Dangerous Species Of Mosquito Using Machine Learning. Ph.D. thesis, University of East Anglia (2022)
14. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
15. Guo, C., Rana, M., Cisse, M., Van Der Maaten, L.: Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117 (2017)
16. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 **2**(7) (2015)
17. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML. pp. 448–456 (2015)
18. Iwana, B.K., Uchida, S.: An empirical survey of data augmentation for time series classification with neural networks. PLOS ONE (2021). https://doi.org/10.1371/journal.pone.0254841

19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
20. Kiranyaz, S., Ince, T., Abdeljaber, O., Avci, O., Gabbouj, M.: 1-d convolutional neural networks for signal processing applications. In: ICASSP. pp. 8360–8364 (2019)
21. Kolen, J., Pollack, J.: Back propagation is sensitive to initial conditions. NeurIPS **3** (1990)
22. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236 (2016)
23. Liang, H., He, E., Zhao, Y., Jia, Z., Li, H.: Adversarial attack and defense: A survey. Electronics **11**(8), 1283 (2022). https://doi.org/10.3390/electronics11081283
24. Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., Zhu, J.: Defense against adversarial attacks using high-level representation guided denoiser. In: CVPR. pp. 1778–1787 (2018)
25. Liu, X., Cheng, M., Zhang, H., Hsieh, C.J.: Towards robust neural networks via random self-ensemble. In: ECCV. pp. 381–397 (2018). https://doi.org/10.1007/978-3-030-01234-2_23
26. Madry, A., Makelov, A., Schmidt, ., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
27. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: DeepFool: A simple and accurate method to fool deep neural networks. In: CVPR (2016). https://doi.org/10.1109/cvpr.2016.282
28. Oregi, I., Ser, J.D., Perez, A., Lozano, J.A.: Adversarial sample crafting for time series classification with elastic similarity measures. In: IDC. pp. 26–39 (2018). https://doi.org/10.1007/978-3-319-99626-4_3
29. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural networks **61**, 85–117 (2015)
30. Shanmugam, D., Blalock, D., Balakrishnan, G., Guttag, J.: When and why test-time augmentation works. arXiv preprint arXiv:2011.11156 (2020)
31. Strauss, T., Hanselmann, M., Junginger, A., Ulmer, H.: Ensemble methods as a defense to adversarial perturbations against deep neural networks. arXiv preprint arXiv:1709.03423 (2017)
32. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
33. Torres, J.F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., Troncoso, A.: Deep learning for time series forecasting: A survey. Big Data **9**(1), 3–21 (2021). https://doi.org/10.1089/big.2020.0159
34. Um, T.T., Pfister, F.M.J., Pichler, D., Endo, S., Lang, M., Hirche, S., Fietzek, U., Kulić, D.: Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In: ACM ICMI. pp. 216–220 (2017). https://doi.org/10.1145/3136755.3136817
35. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: A strong baseline. In: IJCNN. pp. 1578–1585 (2017). https://doi.org/10.1109/ijcnn.2017.7966039
36. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155 (2017)
37. Yu, Y., Yu, P., Li, W.: Auxblocks: defense adversarial examples via auxiliary blocks. In: IJCNN. pp. 1–8 (2019)