IMPROVE ROBUSTNESS OF DNN FOR ECG SIGNAL CLASSIFICATION: A NOISE-TO-SIGNAL RATIO PERSPECTIVE

Linhai Ma & Liang Liang
Department of Computer Science
University of Miami
Coral Gables, FL 33146, USA
{l.ma, liang.liang}@miami.edu

ABSTRACT

Electrocardiogram (ECG) is the most widely used diagnostic tool to monitor the condition of the cardiovascular system. Deep neural networks (DNNs), have been developed in many research labs for automatic interpretation of ECG signals to identify potential abnormalities in patient hearts. Studies have shown that given a sufficiently large amount of data, the classification accuracy of DNNs could reach human-expert cardiologist level. A DNN-based automated ECG diagnostic system would be an affordable solution for patients in developing countries where human-expert cardiologist are lacking. However, despite of the excellent performance in classification accuracy, it has been shown that DNNs are highly vulnerable to adversarial attacks: subtle changes in input of a DNN can lead to a wrong classification output with high confidence. Thus, it is challenging and essential to improve adversarial robustness of DNNs for ECG signal classification – a life-critical application. In this work, we proposed to improve DNN robustness from the perspective of noise-to-signal ratio (NSR) and developed two methods to minimize NSR during training process. We evaluated the proposed methods on PhysionNet's MIT-BIH dataset, and the results show that our proposed methods lead to an enhancement in robustness against PGD adversarial attack and SPSA attack, with a minimal change in accuracy on clean data.

1 INTRODUCTION

Electrocardiogram (ECG) is widely used for monitoring the condition of cardiovascular system. After many years of residency training, a cardiologist becomes experienced in reading ECG graphs, detect abnormalities and classify signals into different disease categories. This is tedious and time-consuming. Researchers found out that deep neural networks (DNNs), especially convolutional neural networks (CNNs) can be trained for ECG signal analysis with excellent classification accuracy (Kachuee et al., 2018) (Hannun et al., 2019). Therefore, with a sufficiently large amount of data and a carefully-designed network structure, DNN models could reach human expert cardiologist level for ECG signal classification, and the analysis for a patient can be done in a fraction of a second. For patients in developing countries where human-expert cardiologists are lacking, a DNN-based automated ECG diagnostic system would be an affordable solution to improve health outcomes.

However, recent studies have shown that despite the high classification accuracy of DNNs, they are susceptible to adversarial attacks in the form of small perturbations to input of the networks, and the perturbation is even imperceptible to human eyes and not expected (by humans) to change the prediction of DNNs (Akhtar & Mian, 2018). Adversarial attacks can be classified into two types based on whether the whole structure of the network is known by the attacker. It is a white box attack if the attacker knows the inner structure of the network, such as Fast Gradient Signed Method (FGSM) (Goodfellow et al., 2015) and Projected Gradient Descent (PGD)(Madry et al., 2017). It is a black box attack if the attacker has almost no knowledge of the inner structure of the network, such as transfer-based attack (Papernot et al., 2017) and SPSA (Uesato et al., 2018). These attacks

(Akhtar & Mian, 2018) pose significant threats to the deep learning systems in sensitive and life-critical application fields such as ECG classification.

To improve DNN robustness, lots of effort has been made by researchers to develop defense methods. Currently, the most popular defense strategy is adversarial training. The basic idea of adversarial training is to added noise to the training samples and the noise is from specific adversarial attacks (e.g. PGD). Through adversarial training, the network can learn some features of adversarial noises and its decision boundary is modified so that it will become difficult to push the input across the decision boundary by adding a small amount of noise. Adversarial training is straightforward but has many problems. For example, generating adversarial samples is very time-consuming, and low-quality adversarial samples can be misleading and even reduce the classification accuracy of networks. Therefore, different adversarial training based defense methods were proposed (Akhtar & Mian, 2018), which share the same basic idea and vary in how the adversarial samples are generated. Parallel to adversarial training, regularization terms can be added to loss function to reduce the sensitivity of network output with respect to the input. The regularization terms could be the gradient magnitude of loss with respect to input (Ros & Doshi-velez, 2017), or Jacobian regularization (Jakubovitz & Giryes, 2018).

2 METHODOLOGY

In this paper, we proposed two new loss functions with regularization terms to improve robustness of neural networks for ECG signal classification. Our methods aim to reduce the effect of noises added to input, from the perspective of noise-to-signal ratio (NSR), which may make the network more robust. We evaluated our methods on PhysioNet MIT-BIH Arrhythmia ECG dataset (Moody & Mark, 2001) (ECG). The results from our experiment show that our proposed methods can achieve a significant improvement in network robustness against PGD white-box attack and SPSA black-box attack, and outperforms the standard adversarial training.

2.1 ECG DATASET

The PhysioNet MIT-BIH Arrhythmia ECG Dataset contains 109446 ECG heartbeat records, and there are 5 categories: N (Normal, Left/Right bundle branch block, Atrial escape and Nodal escape), S (Atrial premature, Aberrant atrial premature, Nodal premature and Supra-ventricular premature), V (Premature ventricular contraction and Ventricular escape), F (Fusion of ventricular and normal) and Q (Paced, Fusion of paced and normal and Unclassifiable). The dataset has been divided into a training set (87554 samples) and a testing set (21892 samples), which is publically available (ECG). We further divided the training set into a 'pure' training set (70043 samples, 80%) and a validation set (17511 samples, 20%). The dataset has a large imbalance between classes, and we performed up-sampling to ensure that there are roughly the same number of samples in each class in the training set and testing set.

2.2 NEW LOSS FUNCTIONS TO REDUCE NOISE-TO-SIGNAL RATIO

In this section, we will introduce two new loss functions to improve network robustness by reducing noise-to-signal ratio (NSR). The noise refers to the adversarial noise generated from an adversarial attack. The signal refers to the output of the classification network (the logits before the softmax layer). Our methods assume that the nonlinear activation function is ReLU or its variants, and the network may have convolution layers, fully-connected layers, pooling layers, batch-normalization layers, dropout layers, and skip-connections. Since a convolution layer is equivalent to a fully-connected layer with shared weights, we can convert a CNN into a multiple layer perceptron (MLP) in theory. Therefore, we will introduce our method using MLP only in this section. Given an input sample x (a vector), the output of a classification network can be exactly expressed by a "linear" equation (Ding et al., 2018):

$$z = W^T x + b \tag{1}$$

where the weight matrix W will be different for different x, and the bias vector b will different for different x. The output is a vector$z = [z_1, \ldots, z_5]^T$where z_i is the output logit of Class i and the total number of classes is 5 in this application. Let w_i be the i^{th} column of W and b_i be the i^{th}

element of b, then we have

$$z_i = w_i^T x + b_i \tag{2}$$

The two new loss functions will be introduced in the following sub-sections using the above notations and equations.

2.2.1 Loss1

Often, the bias terms of a DNN are very small, and they can be ignored. Therefore, equation (2) is simplified to

$$z_y = w_y^T x \tag{3}$$

where y is the true class label of x. z_y is the dot product of two vectors, w_y and x. Given the magnitude of w_y, the dot product reaches the maximum when the two vectors are aligned in the same direction. In other words, we can train a classification network such that x is "memorized" by w_y, similar to a RBF network in some sense (deep RBF network is difficult to train (Goodfellow et al., 2015)). Thus, we define a regularization term to encourage the alignment of the two vectors, which is given by

$$R_1 = ||w_y - \gamma \frac{x}{x^T x}||_2^2 \tag{4}$$

where γ is a scalar coefficient of the unit vector in the direction of x. We combine the regularization term with mean square error (MSE) loss and margin loss for classification, given by

$$loss_1(x, y) = (z_y - 1)^2 + \sum_{i \neq y} (z_i - 0)^2 + \sum_i max(0, 1 - z_y + z_i) + \beta_1 R_1 \tag{5}$$

where β_1 is a scalar parameter to be determined on validation set. In this way, $|z_y| = \left|w_y^T x\right| \rightarrow 1$. As a result, γ can be fixed to 1. The margin loss and the regularization term R_1 are only used for correctly-classified samples; and for wrongly-classified samples, $loss_1$ only contains MSE loss.

This new loss may improve robustness, which can be explained intuitively by Figure 3. Given a small magnitude of adversarial noise ϵ (a vector), to maximally change the output z_y, the direction of ϵ should be aligned with the direction of w_y. Assuming ϵ and w_y are almost in the same direction, then ϵ and x will be almost in the same direction because of $loss_1$, i.e., the noise may look like the input. Therefore, if the magnitude of ϵ is small, it may not alter z_y significantly; and if the magnitude of ϵ is large enough to cause miss-classification, then ϵ becomes visually noticeable.

2.2.2 Loss2

Here, we introduce the second loss which directly minimizes noise-to-signal (NSR) ratio. During an adversarial attack, a noisy vector ϵ is generated and added to the input x, and then the output becomes

$$z_{y,\epsilon} = w_{y,\epsilon}^T (x + \epsilon) + b_{y,\epsilon} \tag{6}$$

If the noise ϵ is small enough, we can assume that: $w_y \approx w_{y,\epsilon}$ and $b_y \approx b_{y,\epsilon}$. This assumption is valid as long as the "on/off" states of ReLU units do not change significantly and pooling masks do not change significantly when adversarial noise is added. Therefore,

$$z_{y,\epsilon} \approx w_y^T x + b_y + w_y^T \epsilon = z_y + w_y^T \epsilon \tag{7}$$

Then, we define NSR and apply Hölder's inequality:

$$NSR_y = \frac{|w_y^T \epsilon|}{|z_y|} \leq \frac{||w_y||_q . ||\epsilon||_p}{|z_y|} \tag{8}$$

where $\frac{1}{p} + \frac{1}{q} = 1$. In this work, we focus on infinite norm $||\epsilon||_\infty = \epsilon_{max}$, and therefore

$$NSR_y \leq \frac{||w_y||_1 . \epsilon_{max}}{|z_y|} = R_2 \tag{9}$$

We combine the regularization term R_2 with mean square error (MSE) loss and margin loss for classification, given by

$$loss_2(x, y) = (z_y - 1)^2 + \sum_{i \neq y} (z_i - 0)^2 + \sum_i max(0, 1 - z_y + z_i) + \beta_2 log(1 + R_2) \tag{10}$$

In the experiment, ϵ_{max} is set to 1, and β_2 is determined on validation set. The margin loss and the regularization term R_2 are only used for correctly-classified samples; and for wrongly-classified samples, $loss_2$ only contains MSE loss.

3 EXPERIMENT AND RESULTS

In the experiment, we applied the two proposed losses on two DNNs to evaluate their robustness. One of the networks is the CNN proposed in (Kachuee et al., 2018), which has 30 layers in total. The other network is an MLP designed by us, and it has 8 layers. The structure of this MLP is (187-128)-RELU-(128-128)-RELU-(128-128)-RELU-(128-32)-(32-5). Based on the performance on validation sets, we set β_1 as 0.2 and set β_2 as 0.5 to obtain a good balance between robustness and accuracy. Number of epochs is 50, optimizer is "Adamax" and learning rate is 0.001.

To study the contribution of each term in the loss functions, we also performed ablation experiment. To obtain the baseline performance, the MLP/CNN were trained with Cross-entropy loss for 50 epochs, with "Adamax" optimizer, and learning rate of 0.001. The two DNNs achieved good performance in classification of ECG signals but are vulnerable to adversarial attacks, as shown in figures from Figure 1 to Figure 2. We also evaluated the other two defense methods: (1) Jacobian regularization (Jakubovitz & Giryes, 2018) and (2) standard adversarial training using 10-PGD (Shafahi et al., 2019).

To evaluate the performance, we compare our proposed methods with other methods, which includes: MLP/CNN with Jacobian regularization (Jakubovitz & Giryes, 2018) and MLP/CNN under 10-PGD standard adversarial training (Shafahi et al., 2019) with noise 0.1, 0.2 and 0.3, where 10 is number of steps of PGD attack to generate adversarial samples. In this experiment, there are totally 9 methods to be compared. They are: MLP/CNN with Cross-entropy loss denoted as "ce" in plots, MLP/CNN with mean square error (MSE) loss denoted as "mse" in plots, MLP/CNN with loss1 denoted as "loss1" , MLP/CNN with loss2 denoted as "loss2", MLP/CNN with Jacobian regularization denoted as "jacob", MLP/CNN with MSE and margin loss denoted as "mseMargin" and MLP/CNN trained with adversarial samples generated by 10-PGD adversarial attack under noise level ϵ denoted as "adv ϵ". The loss function of this adversarial training is (Goodfellow et al., 2015):

$$loss_{adv} = 0.5L_{CE}(x, y) + 0.5L_{CE}(x_\epsilon, y) \tag{11}$$

where L_{CE} is Cross-entropy loss and x_ϵ is adversarial sample. Training batch size is 128 in this experiment. Because we performed up-sampling to ensure that there are roughly the same number of samples in each class in the testing data, average recall is the same as overall accuracy. So, we only report overall accuracy of classification (denoted as "ACC") and average precision across the five classes (denoted as "PREC") as metrics of performance in this experiment.

3.1 100-PGD EVALUATION

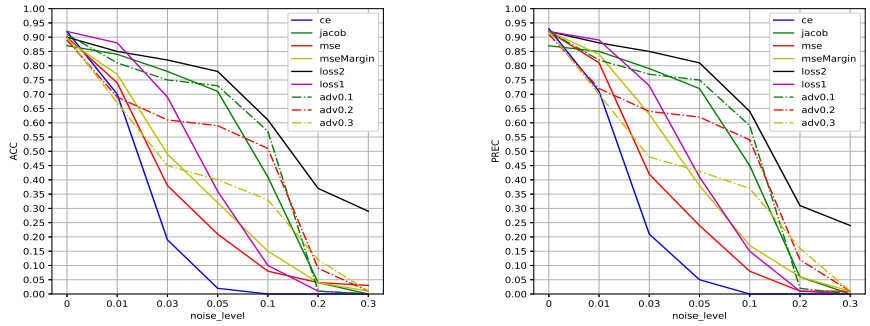
First, we used PGD adversarial attack to test these methods. Projected Gradient Descent (PGD)(Madry et al., 2017) is regarded as the strongest first-order attack. K-PGD attacks clean input x with K steps:

$$x^k = \prod \left(x^{k-1} + \alpha \cdot sign\left(\nabla_x Loss\left(x^{k-1}\right)\right)\right) \tag{12}$$

where α is the step size and x^k is the adversarial example from the k^{th} step. If the noise added to input x is larger than the given noise level ϵ, PGD will project it back to the noise level ϵ. In this experiment, we used 100-PGD to evaluate the methods. Figure 1 presents the results under 100-PGD attack, comparing our proposed methods to Jacobian regulation and 10-PGD adversarial training for MLP. It can be seen that MLP with our proposed loss2 can achieve the best accuracy and precision under all noise levels except for 0.01 (loss1 is better). Under attack on noise level 0.1, MLP with loss2 has accuracy of 61% and precision of 64%, which is the highest among all methods. Figure 2 presents the results under 100-PGD attack, by comparing our proposed methods to Jacobian regulation and 10-PGD adversarial training for CNN. CNN with loss2 has the best performance. Under the PGD attack with the noise level 0.1, CNN with loss2 can maintain an accuracy of more than 71% and a precision of approximately 78%. In this experiment, "adv0.2" and "adv0.3" are not as good as loss2 on all of the noise levels, which means these adversarially-trained CNNs are not strong enough to resist against 100-PGD attack. The other tests can be seen in Appendix.

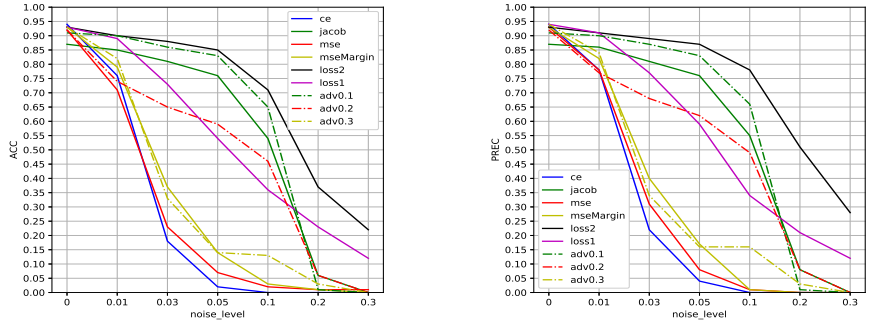
4 CONCLUSION

In this study, we proposed two methods to improve the robustness of deep neural networks for classification of ECG signals. Our methods aim to reduce the proportion of introduced noises in



(a) (b)

Figure 1: Test accuracy (a) and precision (b) for 100-PGD attack on MLP



(a) (b)

Figure 2: Test accuracy (a) and precision (b) for 100-PGD attack on CNN

the output of the neural network, namely, noise-to-signal ratio. In this way, our methods can help reduce the effect of noise on the prediction of the network and thus improve the robustness against adversarial attack. The results of the experiment have shown that our proposed loss2, outperforms all other methods under white-box and black-box attacks (PDG and SPSA), for the classification task. We hope that our approaches may faciliate the development of robust and affordable solutions for automated ECG diagnosis for developing countries.

REFERENCES

ECG Heartbeat Categorization Dataset. URL `https://www.kaggle.com/shayanfazeli/heartbeat`.

Naveed Akhtar and Ajmal Mian. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. *IEEE Access*, 6(August):14410–14430, 2018. ISSN 21693536. doi: 10.1109/ACCESS.2018.2807385.

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Max-Margin Adversarial (MMA) Training: Direct Input Space Margin Maximization through Adversarial Training. 01(1):1–34, 2018. URL `http://arxiv.org/abs/1812.02637`.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–11, 2015.

Awni Y Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H Tison, Codie Bourn, Mintu P Turakhia, and Andrew Y Ng. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, 25(1):65, 2019.

Daniel Jakubovitz and Raja Giryes. Improving dnn robustness to adversarial attacks using jacobian regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 514–529, 2018.

Mohammad Kachuee, Shayan Fazeli, and Majid Sarrafzadeh. ECG heartbeat classification: A deep transferable representation. *Proceedings - 2018 IEEE International Conference on Healthcare Informatics, ICHI 2018*, pp. 443–444, 2018. doi: 10.1109/ICHI.2018.00092.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

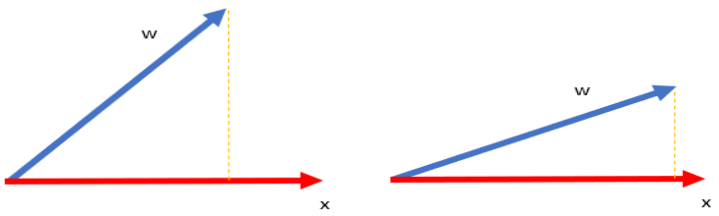
Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Andrew Slavin Ros and Finale Doshi-velez. Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing Their Input Gradients. pp. 1660–1669, 2017.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pp. 3353–3364, 2019.

Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018.

A APPENDIX



(a) (b)

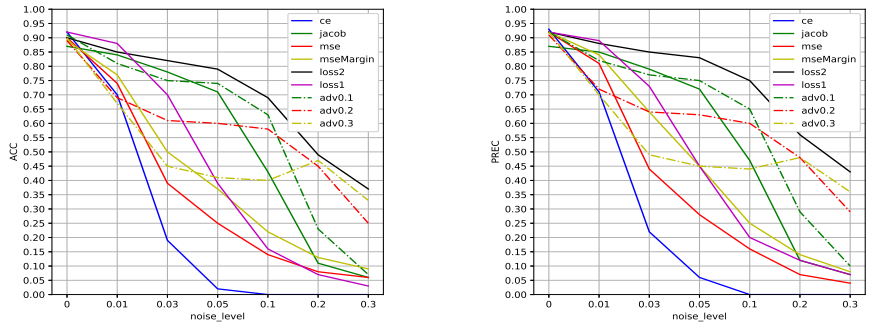
Figure 3: Visualization of 2-dimensional vector x and w. (a) shows the condition where w and x are in very different direction while (b) shows the situation where w and x are in more similar direction.

B APPENDIX

In this experiment, we used 20-PGD to evaluate all methods. Figure 4 presents the results comparing our proposed methods to Jacobian regulation and 10-PGD adversarial training for MLP under 20-PGD attack. First, MLP with our proposed loss2 achieved the best accuracy and precision in general. Under attack on noise level 0.1, MLP with loss2 can still have an accuracy 69% and precision 75%, which is the highest. Second, performance of this adversarial training is not as good as loss2. We can see that there is an intersection of "adv0.1" and "adv0.2" around noise level 0.1 and an

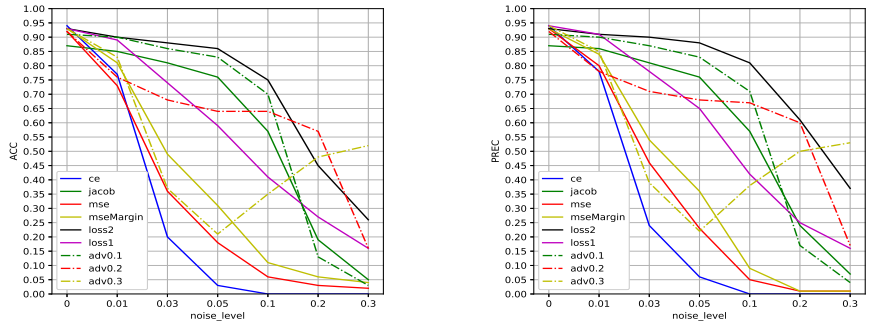
intersection of "adv0.2" and "adv0.3" around noise level 0.2. This means that model trained from adversarial samples under noise ϵ only behaves well around noise level ϵ. Then, we can notice that both our proposed losses can improve the robustness of MLP against PGD attack while keeping a high accuracy and precision in classification of clean data. MLP with loss1 have accuracy 92% and precision 92% and MLP with loss2 can achieve accuracy 90% and precision 92%. MLP with "ce" has accuracy 92% and precision 93%, and the performance dropped quickly as noise level increased

Figure 5 presents the results comparing our proposed methods to Jacobian regulation and 10-PGD adversarial training for CNN under 20-PGD attack. CNN with loss2 has the best performance as well. Under the PGD attack on noise level 0.1, CNN with loss2 can still maintain an accuracy of 75% and a precision of 81%. We can see that "adv0.2" outperforms "loss2" around only noise level 0.2 and "adv0.3", and outperforms "loss2" around only noise level 0.3, while on lower noise level it has a very weak resist to PGD attack (shown in Figure 3 (b)). This also demonstrates that model trained with adversarial samples with noise level ϵ only behaves well around noise level ϵ, which is a weakness of the standard adversarial training.



(a) (b)

Figure 4: Test accuracy (a) and precision (b) for 20-PGD attack on MLP



(a) (b)

Figure 5: Test accuracy (a) and precision (b) for 20-PGD attack on CNN

C APPENDIX

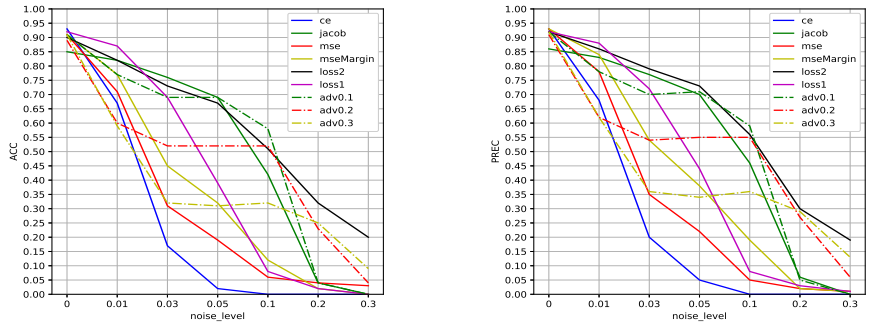
We evaluated the proposed methods with a black-box attack, SPSA adversarial attack(Uesato et al., 2018). SPSA is a gradient-free method, which is useful when the model is non-differentiable, or more generally, the gradients do not point in useful directions. This attack is applied with almost no knowledge of the inner structure of the network. It randomly generates samples around a given

input sample (e.g. an image) of the target network, uses these random samples to estimate the gradient of the target network, and then the estimated gradient can be used for attacks. SPSA runs for 100 iterations with perturbation size 0.01, Adamax learning rate 0.01, and 2048 samples for each gradient estimation. SPSA attack is very computationally expensive, and therefore we only selected the first 162 samples from each class for this experiment.

From Figure 6, we can see that loss2 has the best performance in general for the test of MLP. In Figure 6 (a), "jacob" is the best from noise level 0.03 to noise level 0.05, slightly better than that of loss2. However, "jacob" falls sharply when the noise level is larger than 0.05. When noise level is 0.1, loss2 can still have an accuracy of about 51% while "jacob" method can only achieve 42%. As the noise level becomes higher, "jacob" drops below 10%, while loss2 is always above 20%. In Figure6 (b), we can see that loss2 outperforms "jacob" method under all noise levels. So, loss2 has a better resist against SPSA black-box attack than Jacobian regularization for MLP. In Figure 6 (a), "adv0.1" only outperforms loss2 on noise levels from 0.05 to 0.1, which again confirms that model trained with adversarial samples with noise level ϵ only behaves well around noise level ϵ. In general, our proposed loss2 has a better performance against SPSA attack than all the other methods in comparing.

From Figure 7, we can see that under SPSA black-box attack, loss2 has the best performance in general. In Figure 7 (a), "jacob" method is outperformed by loss2. When noise level is 0.1, loss2 can still have an accuracy of about 60%, better than "jacob" method that can achieve only 52%. In Figure 7 (b), we can see that loss2 outperforms "jacob" methods almost on all noise levels, with precision of 74% at noise level 0.1. So, in the test on CNN, loss2 has a better resist against SPSA black-box attack than Jacobian regularization as well. In Figure 7 (a), "adv0.1" outperforms loss2 at noise level only from 0.01 to 0.1, and "adv0.2" outperforms loss2 only on noise levels from 0.1 to 0.2, which again confirms that model trained with adversarial samples with noise level ϵ only behaves well around noise level ϵ. In general, our proposed loss2 has a better performance against SPSA attack than the other methods.

The good performance of our proposed loss2, under the black-box adversarial attack, also suggests that loss2 is not doing gradient obfuscation to improve the robustness of the target neural network.

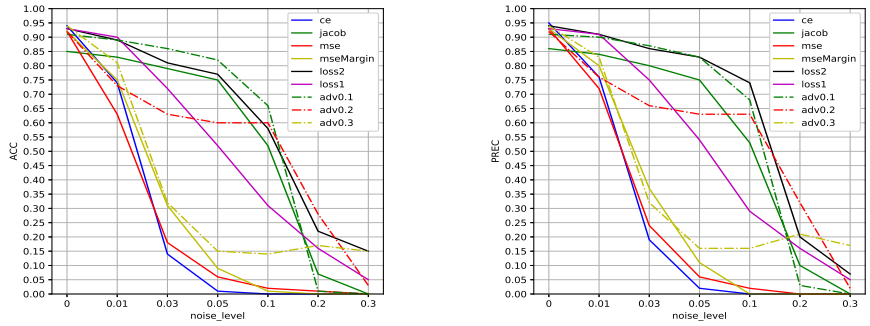


(a) (b)

Figure 6: Test accuracy (a) and precision (b) for SPSA attack on MLP

D APPENDIX

The tables corresponding to figures from Figure 2 to Figure 7 are shown in this section.



(a) (b)

Figure 7: Test accuracy (a) and precision (b) for SPSA attack on CNN

Table 1: ACC MLP 20-PGD

noise level	0	0.01	0.03	0.05	0.1	0.2	0.3
ce	0.92	0.7	0.19	0.02	0.0	0.0	0.0
jacob	0.87	0.84	0.78	0.71	0.43	0.11	0.06
mse	0.9	0.74	0.39	0.25	0.14	0.08	0.06
mseMargin	0.89	0.77	0.5	0.37	0.22	0.13	0.09
loss2	0.9	0.85	0.82	0.79	0.69	0.49	0.37
loss1	0.92	0.88	0.7	0.39	0.16	0.07	0.03
adv0.1	0.91	0.81	0.75	0.74	0.63	0.23	0.07
adv0.2	0.89	0.69	0.61	0.6	0.58	0.45	0.25
adv0.3	0.9	0.67	0.45	0.41	0.4	0.47	0.33

Table 2: PREC MLP 20-PGD

noise level	0	0.01	0.03	0.05	0.1	0.2	0.3
ce	0.93	0.71	0.22	0.06	0.0	0.0	0.0
jacob	0.87	0.85	0.79	0.72	0.47	0.12	0.07
mse	0.92	0.81	0.44	0.28	0.16	0.07	0.04
mseMargin	0.92	0.84	0.64	0.45	0.25	0.14	0.08
loss2	0.92	0.88	0.85	0.83	0.75	0.56	0.43
loss1	0.92	0.89	0.73	0.45	0.2	0.12	0.07
adv0.1	0.92	0.82	0.77	0.75	0.65	0.29	0.1
adv0.2	0.91	0.72	0.64	0.63	0.6	0.48	0.29
adv0.3	0.92	0.7	0.49	0.45	0.44	0.48	0.36

Table 3: ACC CNN 20-PGD

noise level	0	0.01	0.03	0.05	0.1	0.2	0.3
ce	0.94	0.77	0.2	0.03	0.0	0.0	0.0
jacob	0.87	0.85	0.81	0.76	0.57	0.19	0.05
mse	0.92	0.73	0.36	0.18	0.06	0.03	0.02
mseMargin	0.93	0.81	0.49	0.31	0.11	0.06	0.04
loss2	0.93	0.9	0.88	0.86	0.75	0.45	0.26
loss1	0.93	0.89	0.74	0.59	0.41	0.27	0.16
adv0.1	0.91	0.9	0.86	0.83	0.7	0.13	0.03
adv0.2	0.92	0.76	0.68	0.64	0.64	0.57	0.16
adv0.3	0.93	0.83	0.37	0.21	0.35	0.48	0.52

Table 4: PREC CNN 20-PGD

noise level	0	0.01	0.03	0.05	0.1	0.2	0.3
ce	0.94	0.78	0.24	0.06	0.0	0.0	0.0
jacob	0.87	0.86	0.81	0.76	0.57	0.24	0.07
mse	0.93	0.8	0.46	0.23	0.05	0.01	0.01
mseMargin	0.93	0.84	0.54	0.36	0.09	0.01	0.01
loss2	0.93	0.91	0.9	0.88	0.81	0.61	0.37
loss1	0.94	0.91	0.78	0.65	0.42	0.25	0.16
adv0.1	0.91	0.9	0.87	0.83	0.71	0.17	0.04
adv0.2	0.92	0.78	0.71	0.68	0.67	0.6	0.17
adv0.3	0.94	0.85	0.39	0.22	0.38	0.5	0.53

Table 5: ACC MLP 100-PGD

noise level	0	0.01	0.03	0.05	0.1	0.2	0.3
ce	0.92	0.7	0.19	0.02	0.0	0.0	0.0
jacob	0.87	0.84	0.78	0.71	0.41	0.04	0.0
mse	0.9	0.74	0.38	0.21	0.08	0.04	0.03
mseMargin	0.89	0.77	0.49	0.32	0.15	0.04	0.01
loss2	0.9	0.85	0.82	0.78	0.61	0.37	0.29
loss1	0.92	0.88	0.69	0.36	0.1	0.01	0.0
adv0.1	0.91	0.81	0.75	0.73	0.57	0.01	0.0
adv0.2	0.89	0.69	0.61	0.59	0.51	0.09	0.01
adv0.3	0.9	0.67	0.45	0.4	0.33	0.12	0.01

Table 6: PREC MLP 100-PGD

noise level	0	0.01	0.03	0.05	0.1	0.2	0.3
ce	0.93	0.71	0.21	0.05	0.0	0.0	0.0
jacob	0.87	0.85	0.79	0.72	0.45	0.06	0.0
mse	0.92	0.81	0.42	0.24	0.08	0.01	0.01
mseMargin	0.92	0.84	0.63	0.38	0.17	0.06	0.01
loss2	0.92	0.88	0.85	0.81	0.64	0.31	0.24
loss1	0.92	0.89	0.73	0.41	0.15	0.01	0.0
adv0.1	0.92	0.82	0.77	0.75	0.59	0.02	0.0
adv0.2	0.91	0.72	0.64	0.62	0.54	0.12	0.01
adv0.3	0.92	0.7	0.48	0.43	0.37	0.16	0.01

Table 7: ACC CNN 100-PGD

noise level	0	0.01	0.03	0.05	0.1	0.2	0.3
ce	0.94	0.76	0.18	0.02	0.0	0.0	0.0
jacob	0.87	0.85	0.81	0.76	0.54	0.06	0.0
mse	0.92	0.71	0.23	0.07	0.02	0.01	0.01
mseMargin	0.93	0.79	0.37	0.14	0.03	0.01	0.0
loss2	0.93	0.9	0.88	0.85	0.71	0.37	0.22
loss1	0.93	0.89	0.73	0.54	0.36	0.23	0.12
adv0.1	0.91	0.9	0.86	0.83	0.65	0.01	0.0
adv0.2	0.92	0.74	0.65	0.59	0.46	0.06	0.0
adv0.3	0.93	0.82	0.33	0.14	0.13	0.03	0.0

Table 8: PREC CNN 100-PGD

noise level	0	0.01	0.03	0.05	0.1	0.2	0.3
ce	0.94	0.78	0.22	0.04	0.0	0.0	0.0
jacob	0.87	0.86	0.81	0.76	0.55	0.08	0.0
mse	0.93	0.78	0.31	0.08	0.01	0.0	0.0
mseMargin	0.93	0.82	0.4	0.17	0.01	0.0	0.0
loss2	0.93	0.91	0.89	0.87	0.78	0.51	0.28
loss1	0.94	0.91	0.77	0.59	0.34	0.21	0.12
adv0.1	0.91	0.9	0.87	0.83	0.66	0.01	0.0
adv0.2	0.92	0.77	0.68	0.62	0.49	0.08	0.0
adv0.3	0.94	0.84	0.34	0.16	0.16	0.03	0.0

Table 9: ACC MLP SPSA

noise level	0	0.01	0.03	0.05	0.1	0.2	0.3
ce	0.93	0.67	0.17	0.02	0.0	0.0	0.0
jacob	0.85	0.82	0.76	0.69	0.42	0.04	0.0
mse	0.91	0.71	0.31	0.19	0.06	0.04	0.03
mseMargin	0.91	0.77	0.45	0.32	0.12	0.02	0.0
loss2	0.9	0.82	0.73	0.67	0.51	0.32	0.2
loss1	0.92	0.87	0.69	0.39	0.08	0.02	0.0
adv0.1	0.91	0.77	0.69	0.69	0.58	0.04	0.0
adv0.2	0.89	0.6	0.52	0.52	0.52	0.23	0.04
adv0.3	0.91	0.59	0.32	0.31	0.32	0.25	0.09

Table 10: PREC MLP SPSA

noise level	0	0.01	0.03	0.05	0.1	0.2	0.3
ce	0.93	0.68	0.2	0.05	0.0	0.0	0.0
jacob	0.86	0.83	0.77	0.7	0.46	0.06	0.0
mse	0.93	0.78	0.35	0.22	0.05	0.02	0.01
mseMargin	0.93	0.84	0.54	0.38	0.19	0.02	0.01
loss2	0.92	0.86	0.79	0.73	0.56	0.3	0.19
loss1	0.92	0.88	0.72	0.44	0.08	0.03	0.01
adv0.1	0.92	0.78	0.7	0.71	0.59	0.05	0.0
adv0.2	0.91	0.62	0.54	0.55	0.55	0.27	0.06
adv0.3	0.92	0.62	0.36	0.34	0.36	0.29	0.13

Table 11: ACC CNN SPSA

noise level	0	0.01	0.03	0.05	0.1	0.2	0.3
ce	0.94	0.74	0.14	0.01	0.0	0.0	0.0
jacob	0.85	0.83	0.79	0.75	0.52	0.07	0.0
mse	0.92	0.63	0.18	0.06	0.02	0.01	0.0
mseMargin	0.92	0.75	0.31	0.09	0.01	0.0	0.0
loss2	0.93	0.89	0.81	0.77	0.58	0.22	0.15
loss1	0.93	0.9	0.72	0.52	0.31	0.16	0.05
adv0.1	0.91	0.89	0.86	0.82	0.66	0.01	0.0
adv0.2	0.92	0.73	0.63	0.6	0.6	0.28	0.03
adv0.3	0.94	0.81	0.32	0.15	0.14	0.17	0.15

Table 12: PREC CNN SPSA

noise level	0	0.01	0.03	0.05	0.1	0.2	0.3
ce	0.95	0.76	0.19	0.02	0.0	0.0	0.0
jacob	0.86	0.84	0.8	0.75	0.53	0.1	0.0
mse	0.93	0.72	0.24	0.06	0.02	0.0	0.0
mseMargin	0.93	0.8	0.37	0.11	0.0	0.0	0.0
loss2	0.94	0.91	0.86	0.83	0.74	0.2	0.07
loss1	0.93	0.91	0.75	0.54	0.29	0.16	0.05
adv0.1	0.91	0.9	0.87	0.83	0.68	0.03	0.0
adv0.2	0.92	0.76	0.66	0.63	0.63	0.32	0.02
adv0.3	0.94	0.83	0.32	0.16	0.16	0.21	0.17

E APPENDIX

In this section, ECG signals with different levels of noise are shown. The attack is PGD-100 and the target network is MLP, which has been trained with Cross-entropy loss for 50 epochs. As we can see, when the noise level reaches 0.1 (shown in Figure 8 (f)), the ECG signal is even hardly recognizable by human eye.

F APPENDIX

In the experiment, loss funtions $loss_1$ and $loss_2$ run in mini-batches. Assume there are k classes,$loss_1$ should be:

$$R_1 = \sum_i ||w_i - \gamma \frac{X}{X^T X}||_2^2 \tag{13}$$

We combine the regularization term R_1 with mean square error (MSE) loss and margin loss for classification, given by

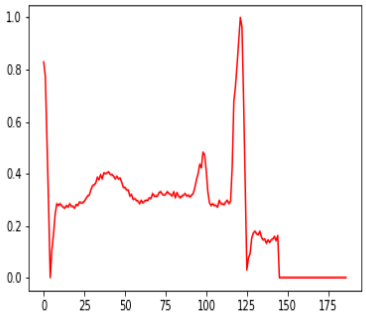
$$loss_1(X, Y) = \sum (Z - Y)^2 + multi\_margin\_loss(Y, Z) + \beta_1 R_1 \tag{14}$$

$loss_2$ should be:

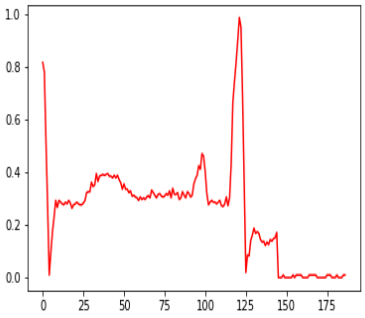
$$R_2 = \sum_i^k \frac{||w_i||_1 . \epsilon_{max}}{|z_i|} \tag{15}$$

We combine the regularization term R_2 with mean square error (MSE) loss and margin loss for classification, given by

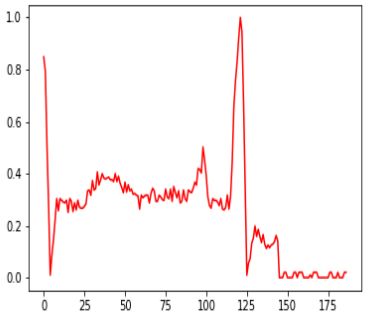
$$loss_2(X, Y) = (Z - Y)^2 + multi\_margin\_loss(Y, Z) + \beta_2 log(1 + R_2) \tag{16}$$



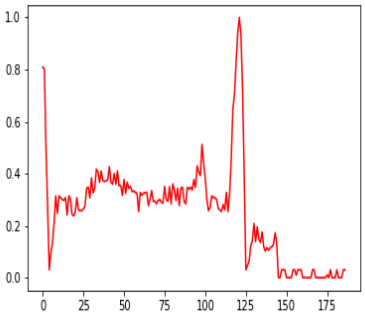
(a) clean ECG



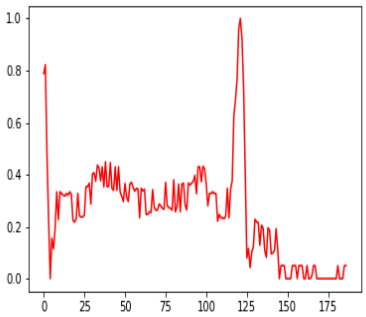
(b) ECG on noise level 0.01



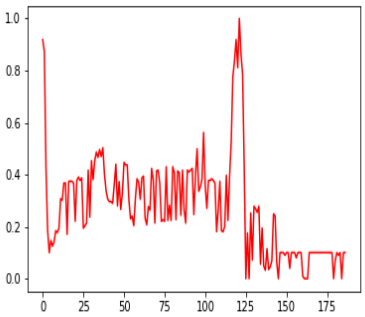
(c) ECG on noise level 0.02



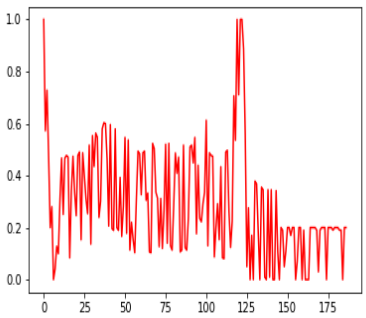
(d) ECG on noise level 0.03



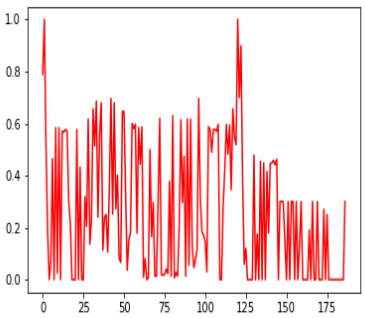
(e) ECG on noise level 0.05



(f) ECG on noise level 0.1



(g) ECG on noise level 0.2



(h) ECG on noise level 0.3

Figure 8: ECG signals on different noise levels