DENSE-DEFENSE: Diversity Promoting Ensemble Adversarial Training Towards Effective Defense

Onat Gungor[1,3], Tajana Rosing[1,2], and Baris Aksanli[3]

[1]Department of Electrical and Computer Engineering, University of California San Diego
[2]Department of Computer Science and Engineering, University of California San Diego
[3]Department of Electrical and Computer Engineering, San Diego State University
ogungor@ucsd.edu, tajana@ucsd.edu, baksanli@sdsu.edu

***Abstract*—Data-driven predictive maintenance utilizes machine learning (ML) to map input sensor data to a desired output. However, security of ML models is vulnerable to adversarial examples which can impact their prediction performance significantly. Ensemble adversarial training (EAT) is one approach to defending ML models against adversarial examples where training data is augmented with perturbations transferred from different pre-trained methods. Diversity among the ensemble learners plays a crucial role in reaching overall ensemble robustness. In this work, we propose a diversity promoting ensemble adversarial training approach as a defense mechanism for data-driven predictive maintenance applications. In our framework, we first measure the loss gradient similarity among pre-trained ML models and select the most dissimilar ones to promote diversity. Then, we create perturbed training instances using the selected diverse base learners and augment those examples into our training data. In testing, we measure the performance change after adversarial attacks are introduced. Our experiments on NASA C-MAPSS dataset show that we can improve the resiliency by up to 97% (43% on average) compared to state-of-the-art training settings.**

***Index Terms*—sensor data processing, data-driven predictive maintenance, secure machine learning, adversarial training**

I. Introduction and Related Work

Remaining useful life (RUL) is defined as the remaining time of a machine to perform its functions until it fails [1]. RUL estimation is a crucial predictive maintenance application to schedule an optimal maintenance [2]. Data-driven RUL estimation utilizes sensor data to build machine learning (ML) models. Recently, this approach became popular with abundance of available sensor data where sensor data collection and processing plays a crucial role to achieving good prediction performance [3], [4]. Furthermore, performance of ML methods relies heavily on input sensor data quality. Thus, these methods are vulnerable to adversarial attacks where an attacker can modify input data or model parameters worsening ML prediction performance significantly [5]. Since ML is in the center of data-driven RUL prediction, these attacks may lead to wrong maintenance decisions causing undetected failures in a system [6]. Hence, there is a need for effective defense mechanisms that can minimize the impact of adversarial attacks in RUL prediction domain.

Adversarial training is one of the most effective defense approaches against adversarial attacks [7]. It augments training data with adversarial examples in each training iteration. However, this approach converges to a degenerate global minimum [8]. To solve this problem, ensemble adversarial training (EAT) is introduced by Tramer et al. [8] where training data is augmented with adversarial examples generated from different target models. EAT provides a better defense mechanism since it is harder for the attacker to trick multiple models in the ensemble instead of just a single model. To obtain ensemble robustness against adversarial attacks, the base learners should be diverse [9]. There are different methods proposed in the literature that promote diversity in ensemble adversarial training [9]–[12]. Yang et al. [12] theoretically show that promoting the orthogonality between gradients of base models leads to higher robustness. Inspired by this work, we promote diversity based on loss gradient similarity among base learners.

This paper proposes diversity promoting ensemble adversarial training framework as a defense mechanism. To the best of our knowledge, our work is the first that proposes ensemble adversarial training towards more resilient data-driven predictive maintenance. Given 10 different pre-trained deep learning (DL) methods, we first calculate pairwise loss gradient similarity. Based on the similarity values, we select the most dissimilar subset of methods. We then create perturbed training examples based on the selected methods where we use Fast Gradient Sign Method [13]. These crafted examples are augmented to the regular (i.e., non-perturbed) training data. Given augmented training data, we train a convolutional neural network (CNN) [14] because of its high accuracy in RUL prediction. In testing, we first create perturbed test instances using our trained CNN based on different adversarial attacks. These instances are then transferred to pre-trained DL methods to measure the performance change after adversarial attacks which we refer as resiliency. The less the performance change is the more resilient a method is. We compare our approach with two non-adversarial state-of-the-art training settings. Our experiments on NASA C-MAPSS dataset [15] show that the proposed ensemble training approach can improve the learner resiliency by up to 97% (43% on average).

II. Proposed Framework

Fig. 1 depicts our proposed ensemble adversarial training framework. Given pre-trained deep learning models, we first

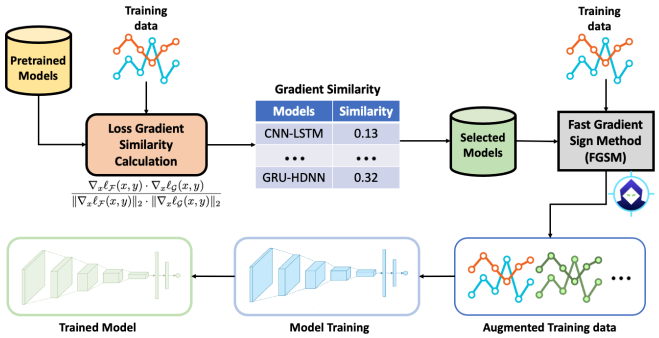

Fig. 1: Our Proposed Framework (*DENSE-DEFENSE*)

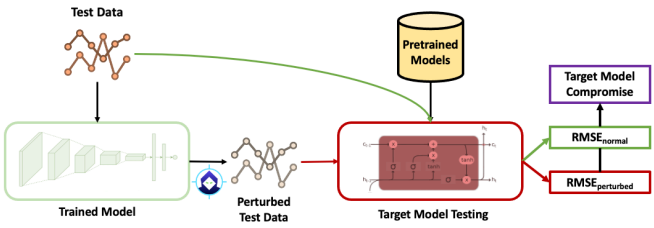

Fig. 2: Testing Framework

calculate the loss gradient similarity among learners and select the most dissimilar ones. Using the selected learners and fast gradient sign method (FGSM), perturbed training examples are generated and augmented to the training data. Then, we train a convolutional neural network (CNN) [14] using the augmented training data. As the output of this framework, we obtain the trained CNN model. Next, we explain the steps of our framework in detail:

Pre-trained Models: We used 10 different pre-trained deep learning models from our previous study [5]: Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Bi-directional LSTM (BLSTM), Gated Recurrent Unit (GRU), Bi-directional GRU (BGRU), Convolutional Neural Network (CNN), Wavenet (WAVE), CNN-LSTM (CLSTM), CNN-GRU (CGRU), GRU-LSTM (GLSTM). We cover a good range of DL methods, increasing the generalizability of our study.

Loss Gradient Similarity Calculation: To introduce diversity into ensemble adversarial training, we measure pairwise loss gradient similarity among two different pre-trained models ($\mathcal{F}$ and $\mathcal{G}$) based on the following formulation [12]:

$$\left| \frac{(\nabla_x \mathcal{L}_\mathcal{F})^T (\nabla_x \mathcal{L}_\mathcal{G})}{\|(\nabla_x \mathcal{L}_\mathcal{F})\|_2 \cdot \|(\nabla_x \mathcal{L}_\mathcal{G})\|_2} \right| \tag{1}$$

where $\nabla_x \mathcal{L}_\mathcal{F}$ and $\nabla_x \mathcal{L}_\mathcal{G}$ denote the loss gradient vectors of base models $\mathcal{F}$ and $\mathcal{G}$ on input x. Note that Equation 1 is the absolute value of cosine similarity between the gradients of the two loss functions. As a result of this step, we obtain the gradient similarity table as illustrated in Fig. 1. The smaller the similarity is, the more diverse the two models are. We then select the models which are least similar to our pre-trained CNN since this model structure will be used in adversarial training. We increment the number of models (thus augmented data size) until no further resiliency improvement is observed.

Augmented Training Data Generation and Training: Based on the selected models from the previous step, we generate perturbed training examples using fast gradient sign method (FGSM) [13]. FGSM first calculates the gradient of the cost function with respect to the input of the neural network. Adversarial examples are then created based on the gradient direction: $\ddot{x} = x + \epsilon * sign(\nabla_x \mathcal{L}(\theta, x, y))$ where $\ddot{x}$ represents the crafted adversarial examples and ϵ denotes the amount of the perturbation. We select FGSM since it can create adversarial examples efficiently [16]. These crafted examples are then augmented to the training data. Given augmented training data, we train convolutional neural network (CNN) [14] due to its high prediction accuracy in RUL prediction. As an output, we obtain the trained CNN model.

Testing Framework: Fig. 2 shows our testing framework where we adapt a transferable black-box attack strategy [17], [18]. Given test data, we first create perturbed test data using our trained model (CNN) based on three different adversarial attacks: fast gradient sign method (FGSM) [13], basic iterative method (BIM) [19], and momentum iterative method (MIM) [20]. We then transfer these instances to our pre-trained models. We measure pre-trained models' prediction performance before ($RMSE_{normal}$) and after ($RMSE_{perturbed}$) the attacks where $RMSE$ refers to root mean squared error. To measure the prediction performance change, we define a metric called *mean compromise* formulated as:

$$Compromise_{mean} = \left(\sum_{i=1}^{M} \frac{RMSE^i_{perturbed}}{RMSEnormal} \right) / M \tag{2}$$

where $Compromise_{mean} > 1$ (with the assumption that attacks lead to worse prediction performance) and M denotes the number of adversarial attacks (i.e., $M = 3$). The **smaller** the *mean compromise* is, the **more resilient** the model becomes against adversarial attacks.

Compared Training Settings: We compare our proposed method's resiliency with two different non-adversarial training settings which directly use the pre-trained models: (i) **white-box setting** creates perturbed test instances using a pre-trained model and use these in the same model's testing (i.e., no test example transfer across different models). For instance, only pre-trained LSTM creates perturbed test instances to be used in LSTM resiliency measurement, (ii) **black-box setting** creates perturbed test examples only using pre-trained CNN model and transfers the examples to other pre-trained models in testing time. This setting is similar to our testing strategy, yet it does not include adversarial training.

III. Experimental Analysis

Dataset Description: We use NASA C-MAPSS [15] which is a benchmark dataset for RUL estimation. This dataset includes multiple aircraft engines simulated under different operating and fault conditions. Fig. 3 depicts the simplified version of simulated engine diagram. The data is collected

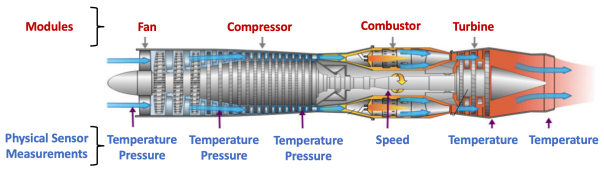

Fig. 3: Engine Diagram Simulated in C-MAPSS [15]

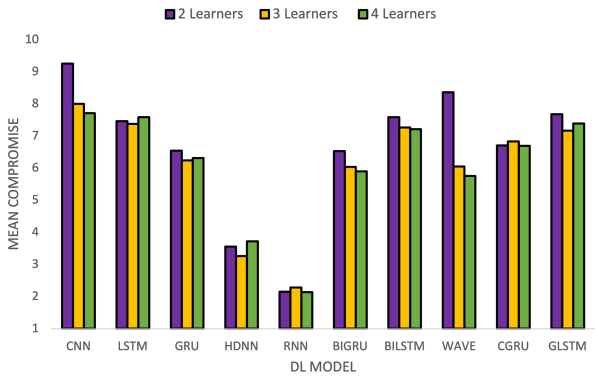

Fig. 4: Impact of Number of Base Learners in Resiliency

TABLE I: *Mean Compromise* Comparison

DL Model / Approach	White-box	Black-box	*DENSE-DEFENSE*
CNN	72.31	72.31	**7.71**
LSTM	7.83	8.46	**7.59**
GRU	7.50	7.87	**6.31**
HDNN	122.50	87.25	**3.72**
RNN	4.36	3.77	**2.13**
BIGRU	7.22	6.67	**5.90**
BILSTM	8.30	8.76	**7.21**
WAVE	25.59	24.45	**5.75**
CGRU	13.19	11.79	**6.69**
GLSTM	8.40	9.17	**7.39**

TABLE II: *DENSE-DEFENSE* Resiliency Improvement

DL Model / Improvement (%)	White-box	Black-box
CNN	89.34	89.34
LSTM	3.13	10.28
GRU	15.83	19.72
HDNN	96.96	95.74
RNN	51.09	43.42
BIGRU	18.38	11.56
BILSTM	13.11	17.66
WAVE	77.52	76.47
CGRU	49.25	43.24
GLSTM	11.98	19.42
Average	**42.66**	**42.69**
Maximum	**96.96**	**95.74**

using temperature, pressure, and speed sensors. We select the FD002 dataset which is one of the most complicated (i.e. the highest number of operating and fault conditions) datasets in C-MAPSS. We have separate training and test data where the goal is to predict RUL for the test data. Our feature columns include the engine ID, cycle index, three operational settings, and 21 different sensor measurements.

Experimental Setup: We use the following parameters for the selected adversarial methods [20], [21]: amount of perturbation (ϵ) = 0.1, step size (α) = 0.001, number of iterations (I) = 100, decay factor (μ) = 1. For the DL model training, we use *Adam* optimizer with learning rate 0.001, *elu* activation function, batch size of 128, and a max number of epochs of 150, and sliding time window size of 80. We repeat each experiment 10 times and report average values. All experiments are run on a PC with 16 GB RAM and an 8-core 2.3 GHz Intel Core i9 processor.

Impact of Number of Base Learners in Resiliency: For our proposed method, we experiment with different number of diverse base learners and measure their *mean compromise* across all pre-trained DL models to determine *DENSE-DEFENSE* optimal configuration. Fig. 4 shows mean compromise values (y-axis) across each DL method (x-axis). We can observe that switching from 2 to 3 learners increase the resiliency of the proposed method. However, adding more models after 3 learner ensemble does not bring a significant resiliency benefit (for some models, it even decreases the resiliency). Specifically, these ensemble configurations (2 learners, 3 learners, and 4 learners) have 6.58, 6.05, and 6.04 average compromise values across all models.

Mean Compromise Comparison: After we selected the optimal configuration for *DENSE-DEFENSE*, we compare our approach with two other training settings (white-box and black-box) explained in Section II-Compared Approaches. Table I shows the mean compromise values for the 3 selected settings: white-box, black-box and our method *DENSE-DEFENSE*. We can observe that for all DL methods, our approach provides the highest resiliency.

Resiliency Improvement: Based on the values in Table I, we also calculate our method's improvement over the white-box and black-box settings. Table II presents the *DENSE-DEFENSE* resiliency improvement over the selected approaches. Compared to white-box and black-box settings, our method improves the resiliency by up-to 96.9% and 95.7% respectively. For both approaches, we obtain 43% average resiliency improvement. The results show that our method provides a more resilient learning solution. Hence, our ensemble training approach is an efficient defense mechanism against different adversarial attacks.

IV. Conclusion

Data-driven remaining useful life (RUL) estimation methods utilize machine learning (ML) in order to map input sensor data to real RUL values. However, ML methods are impacted significantly by small perturbations in input data. Hence, adversarial attacks against ML methods can lead to bad outcomes for predictive maintenance applications. To provide one possible defense against those attacks, in this work we propose diversity promoting ensemble adversarial training where selected diverse base learners' perturbed instances are included in the training process. Our experiments show that our method can be a really efficient defense mechanism against different adversarial attacks where we improve the resiliency by up to 97% (43% on average) compared to state-of-the-art training approaches.

REFERENCES

[1] O. Gungor, T. S. Rosing, and B. Aksanli, "Opelrul: Optimally weighted ensemble learner for remaining useful life prediction," in *2021 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pp. 1–8, IEEE, 2021.

[2] M. S. K. Kopuru, S. Rahimi, and K. Baghaei, "Recent approaches in prognostics: State of the art," in *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, pp. 358–365, The Steering Committee of The World Congress in Computer Science, Computer . . . , 2019.

[3] O. Gungor, T. S. Rosing, and B. Aksanli, "Dowell: diversity-induced optimally weighted ensemble learner for predictive maintenance of industrial internet of things devices," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 3125–3134, 2021.

[4] O. Gungor, T. Rosing, and B. Aksanli, "Enfes: Ensemble few-shot learning for intelligent fault diagnosis with limited data," in *2021 IEEE Sensors*, pp. 1–4, IEEE, 2021.

[5] O. Gungor, T. Rosing, and B. Aksanli, "Stewart: Stacking ensemble for white-box adversarial attacks towards more resilient data-driven predictive maintenance," *Computers in Industry*, vol. 140, p. 103660, 2022.

[6] G. R. Mode and K. A. Hoque, "Crafting adversarial examples for deep learning based prognostics," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 467–472, IEEE, 2020.

[7] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent advances in adversarial training for adversarial robustness," *arXiv preprint arXiv:2102.01356*, 2021.

[8] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*, 2017.

[9] H. Yang, J. Zhang, H. Dong, N. Inkawhich, A. Gardner, A. Touchet, W. Wilkes, H. Berry, and H. Li, "Dverge: diversifying vulnerabilities for enhanced robust generation of ensembles," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5505–5515, 2020.

[10] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu, "Improving adversarial robustness via promoting ensemble diversity," in *International Conference on Machine Learning*, pp. 4970–4979, PMLR, 2019.

[11] S. Kariyappa and M. K. Qureshi, "Improving adversarial robustness of ensembles with diversity training," *arXiv preprint arXiv:1901.09981*, 2019.

[12] Z. Yang, L. Li, X. Xu, S. Zuo, Q. Chen, P. Zhou, B. Rubinstein, C. Zhang, and B. Li, "Trs: Transferability reduced ensemble via promoting gradient diversity and model smoothness," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17642–17655, 2021.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[14] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.

[15] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *2008 international conference on prognostics and health management*, pp. 1–9, IEEE, 2008.

[16] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," *arXiv preprint arXiv:2001.03994*, 2020.

[17] S. Bhambri *et al.*, "A survey of black-box adversarial attacks on computer vision models," *arXiv preprint arXiv:1912.01667*, 2019.

[18] O. Gungor, T. Rosing, and B. Aksanli, "Res-hd: Resilient intelligent fault diagnosis against adversarial attacks using hyper-dimensional computing," *arXiv preprint arXiv:2203.08148*, 2022.

[19] 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.

[20] 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.

[21] 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.