Enabling Cyberattack-Resilient Load Forecasting through Adversarial Machine Learning

Zefan Tang, Jieying Jiao, Peng Zhang, Meng Yue, Chen Chen, Jun Yan

***Abstract*—Developing cyberattack-resilient load forecasting is critical for electric utilities in the face of increasingly broad cyberattack surfaces. It is, however, a challenging task due to the adversary's unknown behaviors. This paper bridges the gap by developing an adversarial machine learning (AML) approach for cyberattack-resilient load forecasting. The novelties of this paper include: 1) its analysis of cyber security issues for traditional artificial neural network (ANN) based load forecasting; 2) the ensemble adversarial training it establishes to tackle different attack scenarios; and 3) the selection of parameters for AML it evaluates to achieve desired performance. Test results validate the effectiveness and excellent performance of the presented method.**

***Index Terms*—Load forecasting, adversarial machine learning, ensemble adversarial training, cyber security, power systems**

I. Introduction

FORECASTING the electricity load for power grids under cyberattacks is an emerging but critically important research field. With the increasing deployment of smart grid technologies like sensing, digital control and communication infrastructure, the data needed as input to forecasting models can be compromised by an adversary through various means. For instance, real-time forecasting data significantly relies on power grids' communication, control, computing infrastructure and hardware facilities, all of which are vulnerable to attack [1], [2]. Compromising critical input forecasting data directly affects the real-time or near real-time operational planning of the grid.

Achieving cyberattack-resilient load forecasting poses a number of challenges. First, increasingly skillful and sophisticated cyber attackers may intrude into a system, make only slight changes in critical data without being detected, and cause significant errors in the forecasting results [3]. Second, accurate forecasting may require different types of data, e.g., historical load, historical and/or current meteorological variables [4], with different vulnerabilities and different impacts on the forecasting results. Moreover, with the fast development of smart grid technologies, the attack surface is becoming increasingly broad, which makes protecting the system from being compromised more difficult [5].

A common approach for cyberattack-resilient load forecasting is to adopt anomaly detection. For instance, by using predictive models, backcasting the input data, and comparing the predicted values with the original ones, the malicious data are likely to be detected [6], [7]. In addition to the model based methods (MBMs), various descriptive analytics based methods (DABMs) are also widely used for point anomalies or contextual anomalies such as abnormal patterns [8]. However, while anomaly detection plays a pivotal role in cleaning the input data, there are still some data not getting detected [9]–[11]. It is critical to develop robust versions of load forecasting to reduce the sensitivity to compromised input data, thus mitigating the impact of unidentified cyberattacks.

Many robust forecasting methods have been devised. For instance, the Huber regression method uses the Huber function to reduce the impact of bad data via the selection of parameters [12]. [13] develops robust versions of the exponential and Holt-Winters smoothing methods. The impact of compromised data is reduced not only in the selection of parameters, but also for the observed values. Moreover, some robust versions of the integration methods, e.g., robust functional principal component analysis [14], are also developed in an ensemble system to provide a more robust forecasting output. However, most of the existing robust methods are only concerned with the outliers, namely, the extremely high/low observations. Very little attention has been paid to other attack scenarios such as small errors on the input data [15].

This paper develops an adversarial machine learning (AML) method for cyberattack-resilient load forecasting. The traditional practice in training forecasting models is to use clean data only; therefore, the forecasting output will be erroneous if the input data become contaminated. In this study, an adversarial training is adopted to increase the robustness of an artificial neural network (ANN) based load forecasting model against cyberattacks. Through adversarial training, the model is trained with not only the clean data, but also malicious data generated by an adversary [16]. To tackle different cyberattack scenarios, ensemble adversarial training is established, and the parameters selection is evaluated for desired performance.

The rest of this paper is organized as follows: Section II describes some key cyber security issues, and this is followed by a description of AML for cyberattack-resilient load forecasting in Section III. The comparison results are provided in Section IV, and Section V concludes the paper.

This work was supported in part by the National Science Foundation under Grants ECCS-1611095, CNS-1647209 and ECCS-1831811, in part by the Department of Energy Cybersecurity for Energy Delivery Systems (CEDS), and in part by the Office of the Provost, University of Connecticut.

Z. Tang and P. Zhang are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269, USA (e-mail: peng.zhang@uconn.edu).

J. Jiao and J. Yan are with the Department of Statistics, University of Connecticut, Storrs, CT 06269, USA.

M. Yue is with Sustainable Energy Technologies Department, Brookhaven National Laboratory, Upton, NY 11973, USA.

C. Chen is with Energy Systems Division, Argonne National Laboratory, Lemont, IL 60439, USA.

978-1-7281-1981-6/19/$31.00 ©2019 IEEE

II. Cyber Security Issues

In this section, the cyber security issues on machine l
based load forecasting are described, and these inclu
following: 1) the forecasting model, 2) the attack mod
3) the impact analysis.

A. Forecasting Model

Achieving load forecasting through the use of n
learning is attracting great attention due to its fle
and suitability for complexity and non-linearity. In a
artificial neural network (ANN), each neuron receives n
inputs, processes them internally, and outputs a re
Different weights are allocated during the combination
inputs, and an activation function is subsequently app
the computed sum. A multilayered ANN consists of a
layer, one or more hidden layers, and an output lay
network propagates the values from the input layer t
the hidden layer(s) to the output layer, where a loss fun
applied. A widely used loss function is the quadratic f
of the output error, which is defined as

$$E_k = \frac{1}{2}(\hat{z}_k - z_k)^2, \tag{1}$$

where E_k is the loss function for the k^{th} output value $\hat{z}_k$, and z_k is the actual value.

The network is then updated iteratively by changing its weights until E_k is minimized or lower than a threshold. Many optimization methods have been devised to achieve this objective, and a famous one is back-propagation, which uses the stochastic gradient descent (SGD) method to estimate the gradients of E_k with respect to the parameters.

A data-based simulation is carried out to evaluate the performance of ANN-based load forecasting. The data are downloaded from the ISO New England website and consist of hourly loads (MW) from 20 power stations. As an example, the data from the first power station are utilized. Two years' data spanning 2004 and 2005 are used as the training data, and the one-year data throughout 2006 are used as the testing data. The ANN model has three layers with 50 neurons in the hidden layer. The hourly loads on each day are used as the input data to the ANN, and the output of the network is the average load on the next day. The relationship between the input and output is mapped, learned and stored into the weights via the two-year training dataset. The performance of the trained ANN is examined with the one-year testing data, as shown in Fig. 1.

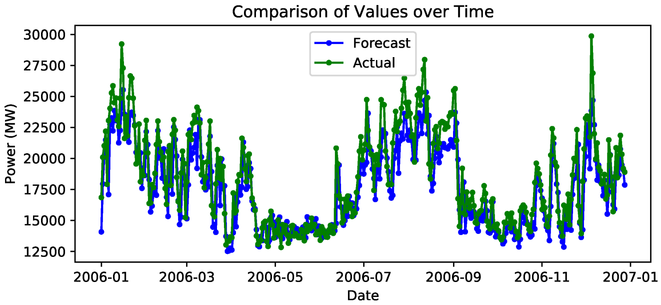


Fig. 1. Comparison results of ANN-based load forecasting and actual values on the testing data throughout 2006.

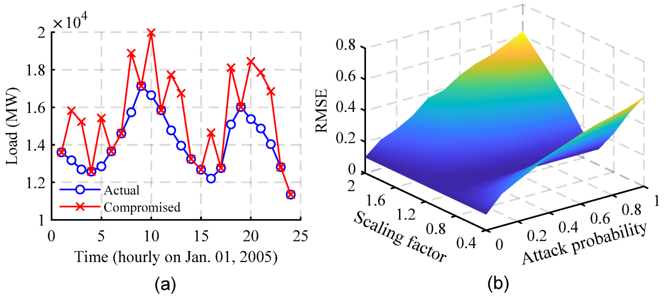


Fig. 2. Impact analysis. (a) An example of the attack model with the data from Jan. 1^{st} 2005, when p_{te} is 0.5 and λ_{te} is 1.2. (b) Illustration of cyberattack impact with different p_{te} and λ_{te}.

To evaluate the accuracy of the predicted results, the root mean square error (RMSE) is calculated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{m=1}^{N}(\hat{z}_m - z_m)^2}, \tag{2}$$

where $\hat{z}_m$ and z_m are the m^{th} predicted and actual values, respectively, and N is the total number of the predicted values.

B. Attack Model

The input data for the ANN model are appealing to malicious attacks. For each data point x_i, the attack can be modeled via two parameters: an attack probability p_{te} and a scaling factor λ_{te}. That is, with a probability p_{te}, each input data point x_i is modified via a scaling factor λ_{te} as follows:

$$x_i^* = \lambda_{te} \times x_i, \tag{3}$$

where x_i^* is the compromised data point.

An example of the attack model is given in Fig. 2 (a), where the data points are hourly and are obtained from Jan. 1^{st} 2005. For each data point, p_{te} is set at 0.5 and λ_{te} is 1.2. Note that 1) different combinations of p_{te} and λ_{te} have different effects on the data points; 2) even with a given pair of p_{te} and λ_{te}, the attack scenarios will be different at different simulation runs, so the simulation is repeated 100 times in this study and the average RMSE is calculated; 3) p_{te} and λ_{te} are only applied on the testing data to model the attack; and 4) the original data downloaded from the ISO New England website are assumed to be clean in this study, while the compromised data are generated through the attack model as shown in (3).

C. Impact Analysis

The impact of cyberattacks with different p_{te} and λ_{te} is illustrated in Fig. 2 (b). It can be seen that 1) when there is no attack, i.e., p_{te} is zero or λ_{te} is one, RMSE reaches the minimum; 2) for a given λ_{te}, the larger the p_{te}, the larger the RMSE; and 3) for a given p_{te}, the more discrepant of λ_{te} and one, the larger the RMSE. Note that, due to the unknown behaviors of the adversary, p_{te} and λ_{te} in the attack model are not known to the defenders.

III. Adversarial Machine Learning

In this section, the AML is presented to enhance the robustness of load forecasting against cyberattacks, which includes: 1) adversarial examples, 2) adversarial training, 3) simplified adversarial training, and 4) ensemble adversarial training.

A. Adversarial Examples

In a traditional ANN, the input training data to the model are clean, making the forecasting results sensitive to malicious data. To enhance the robustness against cyberattacks, the idea of AML is to develop an adversarial training such that the input training data are augmented with adversarial examples.

Most of the existing works on generating adversarial examples are focused on image recognition. The adversarial example x^{adv} is commonly defined as follows [16]:

$$f_W(x^{adv}) \neq z \wedge d(x, x^{adv}) \leq \epsilon, \tag{4}$$

where W is the weight matrix of the network, $f_W(\cdot)$ represents the output of the network, and z is the actual value. ϵ is a given value, and $d(\cdot,\cdot)$ represents the distance between two vectors, i.e., x and x^{adv} in (4). According to (4), the two inputs x and x^{adv} should have a small distance, while at the same time result in different outputs.

For load forecasting, however, different input vectors directly lead to different outputs. One modification is to change (4) as the following optimization problem:

$$\underset{x^{adv}}{\text{argmax}}\ L(f_W(x^{adv}), z) \tag{5}$$

$$\text{s.t. } d(x, x^{adv}) \leq \epsilon, \tag{6}$$

where $L(\cdot,\cdot)$ is the loss function between the prediction $f_W(x^{adv})$ and the actual value z.

B. Adversarial Training

Adversarial training aims to minimize (5) for all adversarial examples x^{adv} by optimizing the weight matrix W. Mathematically, the objective is expressed as

$$\underset{W}{\text{argmin}} \max_{(x,z)\in D} \max_{d(x,x^{adv})\leq\epsilon} L(f_W(x^{adv}), z), \tag{7}$$

where D is the dataset, e.g., the training dataset. The idea of adversarial training is to solve (7) by iteratively executing the following two steps [16]: 1) with all given x^{adv}, find the optimal W for the outer minimization problem; and 2) with the given W, find all the worst-case adversarial examples x^{adv} in the dataset D for the inner maximization problem.

The standard SGD method is used to train the network, namely, estimating the weight matrix W. Each weight w is updated as follows:

$$w_{j+1} = w_j - \eta \sum_{m=1}^{N} \nabla_w L(f_W(x_{mj}^{adv}), z_m), \tag{8}$$

where j denotes the j^{th} iteration, m represents the m^{th} input vector in the training dataset, and N is the total number of the input vectors in the training dataset. Therefore, x_{mj}^{adv} and z_m are the m^{th} adversarial example at the j^{th} iteration and the actual value, respectively. η is the learning rate which controls the speed of the training process.

C. Simplified Adversarial Training

According to (7) and (8), the adversarial examples are re-generated at each iteration in the training process, which inevitably makes the training process complicated and time-consuming. In this paper, a simplified adversarial training is presented. Instead of re-generating the adversarial examples at each iteration, it only generates the adversarial examples once (before the first iteration). Moreover, the generation of adversarial examples in (5) is simplified as follows:

$$x_i^{adv} = \lambda_{tr} * x_i, \tag{9}$$

where x_i^{adv} is the i^{th} data point within each compromised input vector x^{adv}, and x_i is the actual value. In (9), each data point x_i has a probability p_{tr} to be modified. The model in (9) is the same with the attack model in (3), except that (9) is for the training data while (3) is for the testing data.

With a given pair of p_{tr} and λ_{tr}, all the adversarial examples can be generated. The simplified adversarial training then estimates the weights iteratively according to (8) with constant adversarial examples x_m^{adv} replacing the varying x_{mj}^{adv}.

D. Ensemble Adversarial Training

An adversarial training commonly deals with a single attack, i.e., a constant pair of p_{te} and λ_{te} in (3). To tackle different attack scenarios, the paper presents an ensemble adversarial training. Instead of generating adversarial examples with a constant λ_{tr}, the ensemble adversarial training uses a varying λ_{tr} in (9), which is expressed as follows:

$$\begin{cases} \lambda_{tr} = \alpha + \beta \cdot r \\ \alpha = \lambda_{min} \\ \beta = \lambda_{max} - \lambda_{min} \end{cases} \tag{10}$$

where λ_{min} and λ_{max} are the minimum and maximum values of λ_{tr}, respectively. r is a random value with the range from 0 to 1. Each input training data point has a probability p_{tr} to be modified by the scaling factor λ_{tr}. Note that (10) is only applied to the inputs of the training data, and the outputs of the training data are still the true values.

From (10), it can be seen that λ_{tr} ranges from α to $\alpha + \beta$. The three parameters, α, β and p_{tr}, can be used to adjust the performance of load forecasting. Properly selecting α, β and p_{tr} is necessary to achieve desired performance.

IV. Results

In this section, the comparison results are provided, which include 1) ensemble adversarial training (EAdv.) with different combinations of α, β and p_{tr} under a single attack; 2) EAdv. with different combinations of α, β and p_{tr} when there is no attack; 3) effect of β on EAdv.; 4) effect of α on EAdv.; and 5) comparison of EAdv. and the traditional ANN.

A. EAdv. under A Single Attack

Fig. 3 gives the comparison results of EAdv. with differ combinations of α, β and p_{tr} when p_{te} is 0.5 and λ_{te} 1.6. Note that the value within the red circle in each subp represents the RMSE using the traditional ANN, i.e., α = and $\beta = 0$. It can be seen that 1) different combinations of β and p_{tr} have different performances under a single atta 2) compared with the traditional ANN, EAdv. with selec combinations of α, β and p_{tr} can achieve smaller RMSEs; the range from α to $\alpha + \beta$ should include λ_{te}, i.e., 1.6 in t case, to achieve a small RMSE; and 4) when $p_{tr} < p_{te}$, b

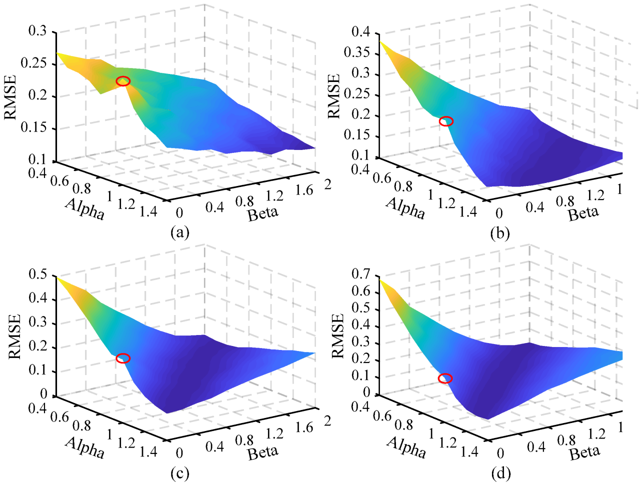


Fig. 3. Comparison results of EAdv. with different combinations of and p_{tr} when p_{te} is 0.5 and λ_{te} is 1.6. (a) $p_{tr} = 0.1$. (b) $p_{tr} = 0.3$ $p_{tr} = 0.5$. (d) $p_{tr} = 0.7$.

B. EAdv. without Attack

As the training data are augmented with adversarial ex ples in the training process, the accuracy of the forecasting results are likely to be decreased when there is no attack. Fig. 4 gives the comparison results of EAdv. with different combinations of α, β and p_{tr} when there is no attack. It can be seen that 1) a large p_{tr} tends to have a large RMSE, especially when $\alpha > 1$ or $\alpha + \beta < 1$; and 2) the range from α to $\alpha + \beta$ should include one to achieve a small RMSE.

C. Effect of β

Fig. 5 gives the comparison results of EAdv. with different β under different attack scenarios, i.e., different combinations of λ_{te} and p_{te}, when $\alpha = 0.6$ and $p_{tr} = 0.3$. It can be seen that different β have different impacts on the RMSE when λ_{te} ranges from 0.4 to 2 and p_{te} ranges from 0 to 1. With a constant α, a larger β tends to have a smaller RMSE when λ_{te} is large, but at the same time, it has a larger RMSE when λ_{te} is small. This is clearly illustrated in Fig. 6.

D. Effect of α

Fig. 7 gives the comparison results of EAdv. with different α under different attack scenarios when $\beta = 0.8$ and $p_{tr} = 0.3$.

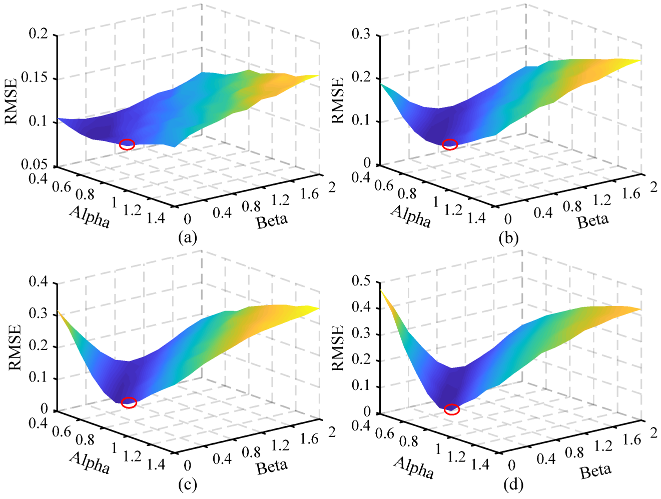


Fig. 4. Comparison results of EAdv. with different combinations of α, β and p_{tr} when there is no attack. (a) $p_{tr} = 0.1$. (b) $p_{tr} = 0.3$. (c) $p_{tr} = 0.5$. (d)

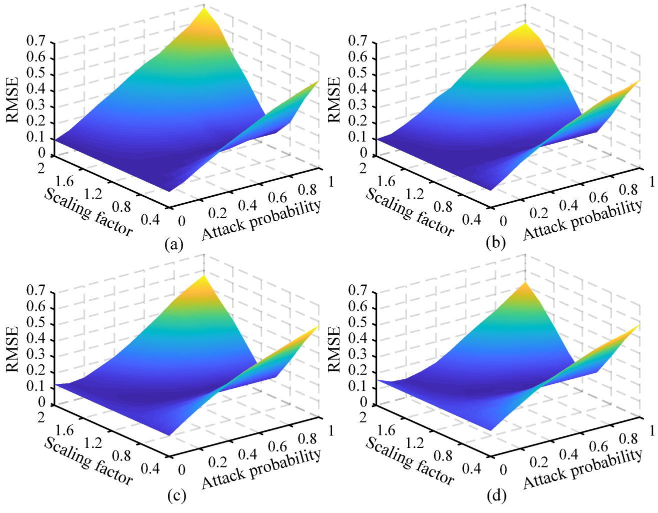
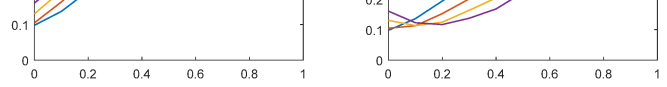


Fig. 5. Comparison results of EAdv. with different β under different attack scenarios when $\alpha = 0.6$ and $p_{tr} = 0.3$. (a) $\beta = 0.4$. (b) $\beta = 0.8$. (c) $\beta = 1.2$. (d) $\beta = 1.6$.

It can be seen that different α have different impacts on the RMSE when λ_{te} ranges from 0.4 to 2 and p_{te} ranges from 0 to 1. With a constant β, a larger α tends to have a smaller RMSE when λ_{te} is large, but have a larger RMSE when λ_{te} is small. It is also clearly illustrated in Fig. 8. Note that the results for α (as shown in Figs. 7 and 8) are similar with those for β (as shown in Figs. 5 and 6). It is reasonable, as the ranges from α to $\alpha + \beta$ in these two cases are similar. For instance, the range from α to $\alpha + \beta$ in Fig. 5 (a) is [0.6, 1], which is similar with the range of [0.4, 1.2] in Fig. 7 (a).

E. Comparison of EAdv. and Traditional ANN

Fig. 9 gives the comparison results of EAdv. and the traditional ANN under different attack scenarios. When λ_{te} is larger than one, i.e., 1.2, 1.4 or 1.6, the parameters for EAdv. are selected as $\alpha = 1$, $\beta = 0.8$, and $p_{tr} = 0.3$. When λ_{te} is smaller than one, i.e., 0.4, 0.6 or 0.8, the parameters for EAdv.



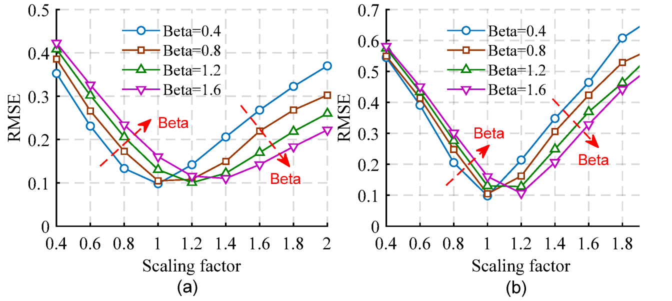
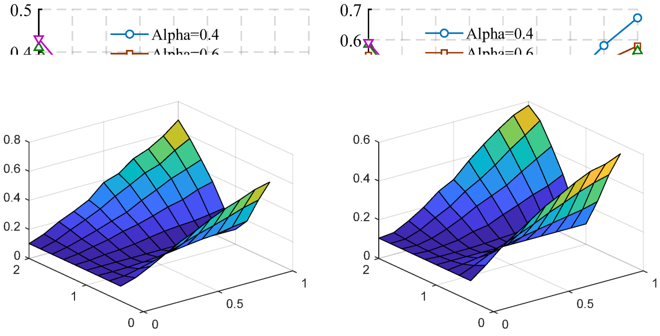


Fig. 6. Comparison results of EAdv. with different β under different att



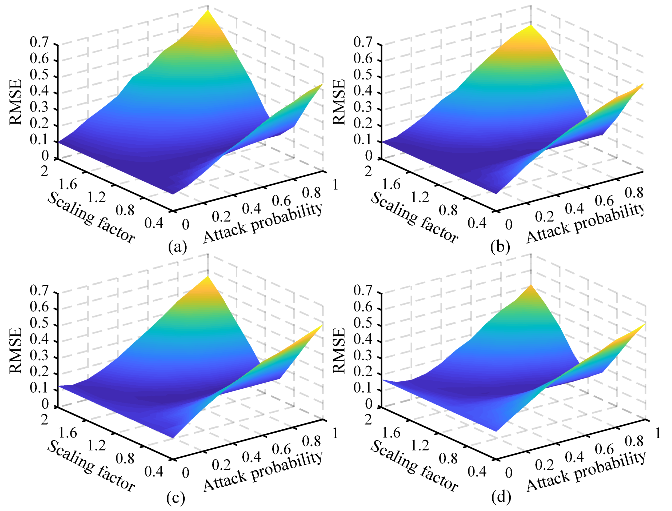


Fig. 7. Comparison results of EAdv. with different α under different attack scenarios when $\beta = 0.8$ and $p_{tr} = 0.3$. (a) $\alpha = 0.4$. (b) $\alpha = 0.6$. (c) $\alpha = 0.8$. (d) $\alpha = 1$.

are selected as $\alpha = 0.4$, $\beta = 0.8$, and $p_{tr} = 0.3$. From Fig. 9, it can be seen that compared with the traditional ANN, EAdv. generally reduces RMSEs under different attacks.

V. Conclusion

This paper develops an AML for cyberattack-resilient load forecasting. While most existing works fail to tackle the unknown behaviors of the adversary, the presented AML bridges this gap by developing an ensemble adversarial training, which can significantly enhance the robustness of the load forecasting against different attack scenarios. As an outcome of this research, this method is to be further developed as a powerful toolbox for system planning, operation and protection. Future works include improving this method for more robust performance and properly combining this method with other approaches such as the anomaly detection.

References

[1] S. Sridhar, A. Hahn, M. Govindarasu *et al.*, "Cyber-physical system security for the electric power grid," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 210–224, 2012.

[2] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Communications Surveys and tutorials*, vol. 14, no. 4, pp. 998–1010, 2012.

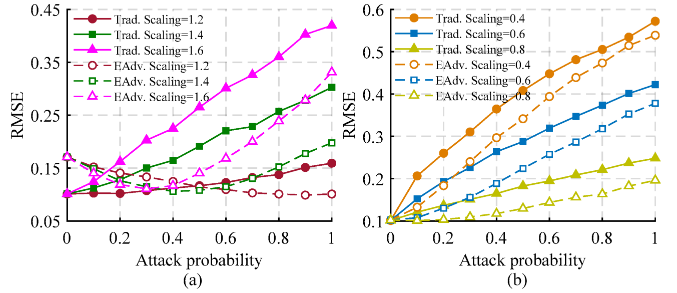


Fig. 9. Comparison results of EAdv. and the traditional ANN under different attack scenarios. (a) $\alpha = 1$, $\beta = 0.8$, and $p_{tr} = 0.3$. (b) $\alpha = 0.4$, $\beta = 0.8$, and $p_{tr} = 0.3$.

[3] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar, and K. Poolla, "Smart grid data integrity attacks," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1244–1253, 2013.

[4] R. Weron, *Modeling and forecasting electricity loads and prices: A statistical approach*. John Wiley & Sons, 2007, vol. 403.

[5] Z. El Mrabet, N. Kaabouch, H. El Ghazi, and H. El Ghazi, "Cyber-security in smart grid: Survey and challenges," *Computers & Electrical Engineering*, vol. 67, pp. 469–482, 2018.

[6] J. Xie and T. Hong, "GEFCom2014 probabilistic electric load forecasting: An integrated solution with forecast combination and residual simulation," *International Journal of Forecasting*, vol. 32, no. 3, pp. 1012–1016, 2016.

[7] L. Jian, H. Tao, and M. Yue, "Real-time anomaly detection for very short-term load forecasting," *Journal of Modern Power Systems and Clean Energy*, vol. 6, no. 2, pp. 235–243, 2018.

[8] Z. Guo, W. Li, A. Lau, T. Inga-Rojas, and K. Wang, "Detecting X-outliers in load curve data in power systems," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 875–884, 2012.

[9] K. G. Boroojeni, M. H. Amini, and S. Iyengar, "Bad data detection," in *Smart Grids: Security and Privacy Issues*. Springer, 2017, pp. 53–68.

[10] C. Tu, X. He, Z. Shuai, and F. Jiang, "Big data issues in smart grid–A review," *Renewable and Sustainable Energy Reviews*, vol. 79, pp. 1099–1107, 2017.

[11] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Transactions on Smart Grid*, 2018.

[12] P. J. Huber *et al.*, "Robust regression: asymptotics, conjectures and Monte Carlo," *The Annals of Statistics*, vol. 1, no. 5, pp. 799–821, 1973.

[13] S. Gelper, R. Fried, and C. Croux, "Robust forecasting with exponential and Holt–Winters smoothing," *Journal of forecasting*, vol. 29, no. 3, pp. 285–300, 2010.

[14] R. J. Hyndman and M. S. Ullah, "Robust forecasting of mortality and fertility rates: a functional data approach," *Computational Statistics & Data Analysis*, vol. 51, no. 10, pp. 4942–4956, 2007.

[15] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 1939–1947.

[16] Q. Cai, M. Du, C. Liu, and D. Song, "Curriculum adversarial training," *arXiv preprint arXiv:1805.04807*, 2018.