Overcoming Stealthy Adversarial Attacks on Power Grid Load Predictions Through Dynamic Data Repair

Xingyu Zhou(✉), Robert Canady(✉), Yi Li(✉), Xenofon Koutsoukos(✉), and Aniruddha Gokhale(✉)

Department of EECS, Vanderbilt University, Nashville, TN 37235, USA
{xingyu.zhou,robert.e.canady,yi.li,xenofon.koutsoukos,
a.gokhale}@vanderbilt.edu

Abstract. For power distribution networks with connected smart meters, current advances in machine learning enable the service provider to utilize data flows from smart meters for load forecasting using deep neural networks. However, recent research shows that current machine learning algorithms for power systems can be vulnerable to adversarial attacks, which are small designed perturbations crafted on normal inputs that can greatly affect the overall performance of the predictor. Even with only a partial compromise of the network, an attacker could intercept and adversarially modify data from some smart meters in a limited range to make the load predictor deviate from normal prediction results. In this paper, we leverage the dynamic data-driven applications systems (DDDAS) paradigm and propose a novel data repair framework to defend against these kinds of adversarial attacks. This framework complements the predictor with a self-representative auto-encoder and works in an iterative manner. The auto-encoder is used to detect and reconstruct the likely adversarial part in the input data. Different reconstruction results come up given different sensitivity levels in detection. As new data flows in each iterative time step, the service provider continuously checks the error of the previous prediction step and dynamically trades off between different detection sensitivity levels to seek an overall stable data reconstruction. Case studies on power network load forecast regression demonstrate the vulnerability of current machine learning algorithms and correspondingly the effectiveness of our defense framework.

Keywords: Power systems · Adversarial attacks · Load forecasting · Dynamic data repair

This work is supported in part by AFOSR DDDAS FA9550-18-1-0126 program. Any opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the sponsor.

F. Darema et al. (Eds.): DDDAS 2020, LNCS 12312, pp. 102–109, 2020.
https://doi.org/10.1007/978-3-030-61725-7_14

1 Introduction

In modern smart grids, accurate load forecasting is critical for managing the infrastructure through targeted pricing and predictive maintenance. Advances in machine learning enable the service provider to utilize data flows from smart meters to perform load forecasting [9] using a deep learning model. However, recent research [6] reveals that current machine learning algorithms proposed for power system application scenarios can be vulnerable to adversarial attacks [11], which are inputs with small designed perturbations added to normal ones that can adversely affect the overall performance of the predictor [7,10]. In partially compromised hierarchical power networks, an attacker could intercept and maliciously modify data from some smart meters with small perturbations that can still make the load predictor deviate from normal prediction results.

To address these issues, we adopt the dynamic data-driven applications systems (DDDAS) paradigm [3] in providing a novel data repair framework to defend against such kind of adversarial attacks as shown in Fig. 1. This framework extends our prior work [13] of a cloud-supported platform for sensor networks (e.g., smart grid networks) to formalize general resilience testing procedures under adversarial settings using the model-driven approach [4]. To the best of our knowledge, this work is the first to introduce such a kind of dynamic data repair against adversarial attacks [5], and make the following contributions in this paper.

- We present a framework that can formalize the security and resilience testing in distributed sensor networks under adversarial settings;
- We design an iterative dynamic data repair scheme of Dropout-Detect-Reconstruct-Tradeoff to boost the robustness of data using the DDDAS paradigm for ongoing predictions; and
- We conduct a case study for distributed power network load forecasting to demonstrate potential risks for machine learning predictors and the efficiency of our defensive data repair framework.

The rest of the paper is organized as follows. Section 2 illustrates the theoretical background of our adversarial attack setting and dynamic data repair framework in a step-by-step manner. Section 3 presents a case study to demonstrate the capabilities of our framework on a power distribution network. Finally, Sect. 4 concludes the paper and presents opportunities for future research.

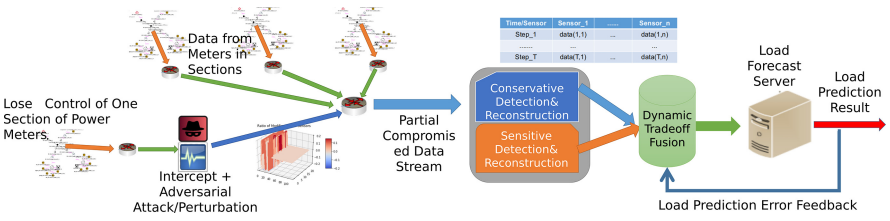


Fig. 1. Overall workflow for dynamic data repair under adversarial attack

2 Methodology

In this section we provide details of our approach. The techniques will be introduced following the execution order of attack and defense. Our predictor is based on deep learning. Specifically, the model absorbs data from distributed sensors and fetches their values from current and some time steps back to predict the total system load for the next time step.

2.1 Model of Stealthy Adversarial Attacks

To compromise the prediction system, an attacker intercepts and adds designed perturbations to the normal data flow. Without loss of generality, we assume that the attacker's goal is to maximize the load prediction deviation. For this scenario, larger ranges of input and output numerical value data space as well as the adoption of anomaly detectors leads to higher complexity in attack settings. To illustrate the vulnerabilities of the prediction system, we propose an attack method adapted from the most popular adversarial attack called FGSM (Fast Gradient Sign Method) [7], which generates adversarial perturbations using only one single equation: $\eta = \epsilon \cdot sign(\nabla_x J(\theta, x, y))$. Here θ represents the parameters of the model, x represents inputs to the model, y refers to the targets associated with x (for tasks with targets) and $J(\theta, x, y)$ is the goal loss function for deviating the neural network. The magnitude constraint added to the original sample is represented by ϵ.

With the presence of an anomaly detector, we reformulate an adversarial attack [13] as an optimization problem which attempts to find the best synthetic perturbations that maximize the prediction loss while keeping the modification magnitude at a small enough level so as to go undetected. Compared to the FGSM attack, we implement an iterative attack that allows each meter (value in input data array) to have its unique modification value because the input range may not be fixed. Our approach performs a number of iterations with small step ratio and updates the gradient sign method from the output of the previous iteration. Intermediate results are first checked with the detector to remove exposed parts and then sent into the next iteration for further exploration. This procedure eventually generates an adversarial but undetected data sample.

2.2 Resilient Detection and Reconstruction

To detect compromised sensors, we use an auto-encoder as the self-representation to build an anomaly detector. Auto-encoder models learn internal representations with the objective $AE(x) = x$ mapping to the input distribution itself. For the sensor network in our case study, we set individual detection thresholds for each meter reading. After training the auto-encoder using the training data, we use the training data to compute the fitting error (l_2 Norm) for all sensors and using maximum fitting deviation of each sensor as the error threshold for anomaly detection. During the prediction phase, the auto-encoder takes inputs and compares output residuals with the pre-computed thresholds and generates

a list of sensors with the potential for adversarial attacks. In this way the detector judges whether specific sensors in the network are likely to be compromised.

Such a static detection is still vulnerable to stealthy attacks and can be made resilient when the input test sample first goes through a randomized dropout step [2]. The detection runs with controllable sensitivity levels. With $dctIter$ dropout iterations, if no less than $dctThres$ times the sensor has been marked as anomaly if would be returned as a high likely adversarial sensor. Different reconstruction results come up given different sensitivity levels in such a detection phase. In each detection iteration, a portion of the input data is randomly dropped out and a reconstruction is conducted using the remaining data. The residual between the original and the reconstructed data can be used to detect the likely adversarial part of data. Based on the detection results, the likely adversarial data part can be erased and reconstructed using the auto-encoder.

2.3 Iterative Dynamic Repair

The resilient detection and reconstruction procedure is configurable and sensitive to measurements. One key property for prediction tasks like load forecasting is that as new data flows in continuously, the system can utilize new data to validate the quality of previous predictions for which the DDDAS paradigm [3] is best suited to provide adaptive data repair against adversarial attacks as shown in Algorithm 1.

For the resilient detection and reconstruction, given a fixed dropout rate, the sensitivity can be adjusted with the number of detection iteration ($dctIter$) and the detection iteration threshold ($dctThres$). Given the infinite number of combination settings for the resilient detection, we consider three settings with the least computation burden (sensitivity from high to low): (1) $x1in2t \leftarrow resCor(x, dctIter = 2, dctThres = 1)$ and (2) $x1in1t \leftarrow resCor(x, dctIter = 1, dctThres = 1)$ and (3) $x2in2t \leftarrow resCor(x, dctIter = 2, dctThres = 2)$. We implement adjustments in iterative time steps to seek a balanced trade-off between sensitivity levels. The overall prediction result with dynamic repair is computed as a weighted sum of these three resilient reconstructions [12]. For each time step, the system checks the previous prediction deviations from these three levels and allocates higher weights for the least deviated reconstruction level.

3 Empirical Validation of the Claims

3.1 Power System Setting

For data collection, we conduct a detailed simulation of an electric distribution system using GridLAB-D provided by the Pacific Northwest National Laboratory (PNNL) [8]. We selected the prototypical feeder of a moderately populated area *R1-12.47-3*, and included representative residential loads like heating, ventilation and air conditioning (HVAC) systems to the distribution network model [1]. In

Algorithm 1. Dynamic Repair (*dynRepair*)

Require: $\boldsymbol{x}$: original observation data flow; f: predictor; $NumTime$: number of execution time steps; $resCor$: resilient correction function; $ErrThres$: ideal prediction error threshold; $return$: return function for each time step; y: ground truth value.

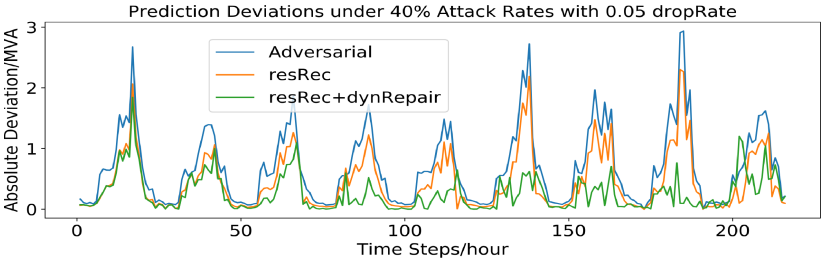
$\alpha = [1.0, 0.0, 0.0]$, $\alpha_{bias} = 0.05$, $x \leftarrow \boldsymbol{x}[0]$,$t \leftarrow 1$
$pred, pred1in1, pred1in2, pred2in2 \leftarrow EmptyList$
$x1in1t \leftarrow resCor(x, dctIter = 1, dctThres = 1)$, $pred1in1.append(f(x1in1t))$
$x1in2t \leftarrow resCor(x, dctIter = 2, dctThres = 1)$, $pred1in2.append(f(x1in2t))$
$x2in2t \leftarrow resCor(x, dctIter = 2, dctThres = 2)$, $pred2in2.append(f(x2in2t))$
$pred[0] \leftarrow pred1in1t * \alpha[0] + pred1in2t * \alpha[1] + pred2in2t * \alpha[2]$
while $t < NumTime$ **do**
  $resPre1 \leftarrow abs(pred[t-1] - y[t-1])$, $resPre2 \leftarrow abs(pred[t-2] - y[t-2]$
  if $t > 1$ **and** $resPre1 > ErrThres$ **and** $resPre1 > resPre2$ **then**
    $res1in1 \leftarrow abs(pred1in1[t-1] - y[t-1])$
    $res1in2 \leftarrow abs(pred1in2[t-1] - y[t-1])$
    $res2in2 \leftarrow abs(pred2in2[t-1] - y[t-1])$
    $idx = argmin([res1in1, res1in2, res2in2])$
    $\alpha \leftarrow \alpha - \alpha_{bias}$, $\alpha[idx] \leftarrow \alpha[idx] + 3 * \alpha_{bias}$
  end if
  $x \leftarrow \boldsymbol{x}[t]$
  $x1in1t \leftarrow resCor(x, dctIter = 1, dctThres = 1)$, $pred1in1.append(f(x1in1t))$
  $x1in2t \leftarrow resCor(x, dctIter = 2, dctThres = 1)$, $pred1in2.append(f(x1in2t))$
  $x2in2t \leftarrow resCor(x, dctIter = 2, dctThres = 2)$, $pred2in2.append(f(x2in2t))$
  $pred[t] \leftarrow f(x1in1t) * \alpha[0] + f(x1in2t) * \alpha[1] + f(x2in2t) * \alpha[2]$,
  $return(pred[t])$,$t \leftarrow t + 1$
end while

summary, our distribution model has a total of 109 commercial and residential user loads. Smart meters are connected to end users and their usage data reports are transmitted to the upper-level control center in a hierarchical manner. For each hourly time step, the prediction model takes load data from distributed meter readings in the past 24 h and also takes into account the temperature data for the same period of time. We build a load forecasting model for this power distribution network using a relatively large LSTM deep neural network (with 3 LSTM layers of 150 units and 2 fully-connected layers of 200 units). The predictor on the clean data generates a mean squared error (MSE) of 0.1255 (Mega Volt Amp) on the test data set for a total of 216 time steps.

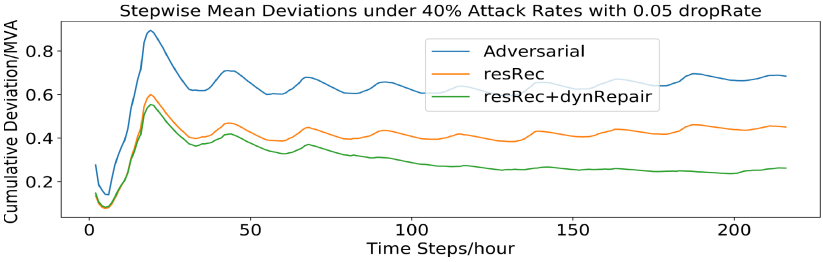
The attack scenario is a manipulation of sensor data under reasonable constraints with full knowledge of the prediction and detection model. In each time step, the attacker can manipulate a fixed number of meters in the network (10%–50% in our experiments). Moreover, for each meter, the attacker is allowed to deviate the meter reading by a limited level of 20%. Under these constraints, we generate stealthy adversarial examples using the iterative attack method.

3.2 Evaluating Reconstruction and Repair

We evaluate our dynamic data repair framework on various settings under strong attacks with a maximum modification ratio of 20% for compromised sensors. Figure 2 shows the prediction results when 40% of sensors in the network are compromised in two ways: (a) shows absolute prediction deviations from normal prediction results, and (b) shows mean absolute prediction deviations from normal prediction results of current prediction and all the ones prior to the current time step. Even with a large portion of 40% sensors compromised, the adversarial impact can still be mitigated to an overall practical level of 0.3 (Mega Volt Amp).



(a) Absolute Prediction Deviation from Original Prediction



(b) Cumulative Mean Absolute Deviation from Original Prediction

Fig. 2. Predictions under 40% compromise and 5% detection dropout rate

We present experimental results under more flexible settings in Table 1, which shows results under four levels of detection dropout rate: $5\%, 10\%, 20\%, 30\%$ with $20, 40, 60, 80$ reconstruction cycles. The error metric we chose is the most commonly used mean squared error (MSE) over the test dataset. For different attack rates, the best defense settings are marked in dark black. We can see that low detection dropout rates with more detection cycles usually show more stable prediction performances. From the figures we can also see that adversarial impacts in this load forecast case usually occurs at peak points. Further, the data repair framework successfully decreases prediction deviations at these vulnerable points without much impact on other locations.

The experimental results also clearly show the trade-off caused by the iterative data repair. With a large number of detection iterations, the chance of being totally stealthy for an adversarial sensor is reduced to a negligible level. Meanwhile, low threshold settings lead to obvious negative impacts caused by false alarms. From our experiments, the upper bound of this repair is determined by the performance of this self-representation model (auto-encoder here) and therefore we can see that dynamic repair does not always show best performance when the compromised sensor ratio is relatively low. This sensitive repair might lead to an unstable prediction performance over detection iterations in each time step. As shown in our experiments, this potential risk is most obvious when the detection dropout rate is high. As a result, the combination of a relative low detection dropout rate along with more iterations would usually lead to smoother and more stable performance.

Table 1. Prediction Mean Squared Error (MSE) under different settings

Drop/%	adv/%	natErr	advErr	resRec/numCycle				resRec+dynRepair/numCycle			
				20	40	60	80	20	40	60	80
5	10	0.126	0.173	0.152	0.144	0.142	**0.141**	**0.146**	0.147	0.150	0.149
	20	0.126	0.311	0.211	0.163	0.148	**0.143**	0.170	0.159	0.155	**0.149**
	30	0.126	0.538	0.380	0.300	0.257	**0.232**	0.281	0.216	0.197	**0.188**
	40	0.126	0.921	0.729	0.626	0.559	**0.523**	0.566	0.442	0.345	**0.301**
	50	0.126	1.329	1.090	0.979	0.909	**0.862**	0.876	0.719	0.581	**0.500**
10	10	0.126	0.171	0.139	**0.137**	0.138	0.139	0.152	0.148	**0.146**	0.147
	20	0.126	0.311	0.174	0.144	**0.139**	0.139	0.158	**0.150**	0.152	0.168
	30	0.126	0.538	0.310	0.236	0.211	**0.200**	0.224	0.183	**0.179**	0.179
	40	0.126	0.921	0.632	0.524	0.481	**0.464**	0.406	0.293	0.270	**0.267**
	50	0.126	1.329	0.984	0.859	0.808	**0.784**	0.688	0.526	0.477	**0.455**
20	10	0.126	0.173	**0.139**	0.146	0.145	0.145	**0.140**	0.153	0.150	0.151
	20	0.126	0.311	0.142	0.139	**0.138**	0.171	**0.160**	0.300	0.308	0.286
	30	0.126	0.538	0.229	**0.201**	0.216	0.218	**0.182**	0.192	0.229	0.273
	40	0.126	0.921	0.541	0.473	0.459	**0.457**	0.912	0.994	0.803	**0.760**
	50	0.126	1.329	0.850	0.777	0.767	**0.759**	0.567	**0.527**	0.559	0.579
30	10	0.126	0.173	**0.139**	0.140	0.139	0.139	0.147	0.140	**0.138**	0.142
	20	0.126	0.311	**0.138**	0.138	0.139	0.139	0.150	0.148	0.139	**0.139**
	30	0.126	0.538	0.219	0.201	0.202	**0.201**	0.197	**0.184**	0.197	0.213
	40	0.126	0.921	0.521	0.491	0.488	**0.485**	**0.378**	0.388	0.429	0.462
	50	0.126	1.329	0.876	0.842	**0.840**	0.838	**0.598**	0.644	0.699	0.786

An important property of our approach is that it takes advantage of existing pre-trained models in a resilient way, which means it can be combined with other defense techniques with no constraints. It is a generalized model deployment strategy to improve robustness that is easily transferable to other learning settings.

4 Conclusion

This paper demonstrated how to analyze and improve the robustness of learning-based prediction models in power distribution networks using the DDDAS paradigm. Given the existence of threats from stealthy adversarial attacks, we first designed a resilient detection and reconstruction strategy using randomization elements. We then proposed a practical, iterative dynamic data repair strategy to seek an optimal trade-off between reconstruction results from different sensitivity levels. Our work not only shows the importance of introducing randomization elements to increase robustness in learning-based systems but also the effectiveness of deviation feedback for predictions on-the-fly. Even though our defense framework has shown promising results, the computation cost for an optimal defense efficiency can be very high thereby requiring new approaches to simplify and accelerate computations for real time applications.

References

1. https://github.com/gridlab-d/Taxonomy_Feeders (2015). Accessed October 2019
2. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420 (2018)
3. Blasch, E., Bernstein, D., Rangaswamy, M.: Introduction to dynamic data driven applications systems. In: Blasch, E., Ravela, S., Aved, A. (eds.) Handbook of Dynamic Data Driven Applications Systems, pp. 1–25. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95504-9_1
4. Broll, B., Whitaker, J.: DeepForge: an open source, collaborative environment for reproducible deep learning (2017)
5. Carlini, N., Wagner, D.: Adversarial examples are not easily detected: bypassing ten detection methods. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 3–14. ACM (2017)
6. Chen, Y., Tan, Y., Zhang, B.: Exploiting vulnerabilities of load forecasting through adversarial attacks. In: Proceedings of the Tenth ACM International Conference on Future Energy Systems, pp. 1–11. ACM (2019)
7. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
8. Schneider, K.P., Chen, Y., Chassin, D.P., Pratt, R.G., Engel, D.W., Thompson, S.E.: Modern grid initiative distribution taxonomy final report. Technical report. Pacific Northwest National Laboratory (2008)
9. Sevlian, R., Rajagopal, R.: A scaling law for short term load forecasting on varying levels of aggregation. Int. J. Electr. Power Energy Syst. **98**, 350–361 (2018)
10. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
11. Vorobeychik, Y., Kantarcioglu, M.: Adversarial machine learning. Synth. Lect. Artif. Intell. Mach. Learn. **12**(3), 1–169 (2018)
12. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. arXiv preprint arXiv:1901.08573 (2019)
13. Zhou, X., et al.: Evaluating resilience of grid load predictions under stealthy adversarial attacks. In: 2019 Resilience Week (RWS), vol. 1, pp. 206–212. IEEE (2019)