Evaluating Resilience of Grid Load Predictions under Stealthy Adversarial Attacks

Xingyu Zhou, Yi Li, Carlos A. Barreto, Jiani Li, Peter Volgyesi, Himanshu Neema, Xenofon Koutsoukos
Institute for Software Integrated Systems
Vanderbilt University
Nashville, TN 37235

***Abstract*—Recent advances in machine learning enable wider applications of prediction models in cyber-physical systems. Smart grids are increasingly using distributed sensor settings for distributed sensor fusion and information processing. Load forecasting systems use these sensors to predict future loads to incorporate into dynamic pricing of power and grid maintenance. However, these inference predictors are highly complex and thus vulnerable to adversarial attacks. Moreover, the adversarial attacks are synthetic norm-bounded modifications to a limited number of sensors that can greatly affect the accuracy of the overall predictor. It can be much cheaper and effective to incorporate elements of security and resilience at the earliest stages of design. In this paper, we demonstrate how to analyze the security and resilience of learning-based prediction models in power distribution networks by utilizing a domain-specific deep-learning and testing framework. This framework is developed using DeepForge and enables rapid design and analysis of attack scenarios against distributed smart meters in a power distribution network. It runs the attack simulations in the cloud backend. In addition to the predictor model, we have integrated an anomaly detector to detect adversarial attacks targeting the predictor. We formulate the stealthy adversarial attacks as an optimization problem to maximize prediction loss while minimizing the required perturbations. Under the worst-case setting, where the attacker has full knowledge of both the predictor and the detector, an iterative attack method has been developed to solve for the adversarial perturbation. We demonstrate the framework capabilities using a GridLAB-D based power distribution network model and show how stealthy adversarial attacks can affect smart grid prediction systems even with a partial control of network.**

***Index Terms*—power systems, adversarial attacks, load forecasting, model-based design, testbed**

I. Introduction

For electricity markets, the profit and safety is based on a dynamic balance between supply and demand. As a result, an accurate demand prediction system is essential for the pricing strategy to maintain infrastructures as well as to maximize the profit. For smart grids, service providers may try to reduce the uncertainties by forecasting the demand of their customers. In particular, the accuracy of the forecasts improves aggregating the demand of large groups of customers [1].

With recent developments of machine learning techniques especially deep learning, neural network prediction models can be constructed to gain good results. Time-series network models from distributed meter readings of different areas in the smart grid is useful in predicting the future load demands.

However, the complexity of current inference systems leads to vulnerabilities that can be exploited by adversarial attacks. In particular, deep neural networks are susceptible to adversarial examples, which poses a great risk in current machine learning systems. Adversarial examples are synthetic modifications added to original samples in a way that can misguide the neural network prediction systems. These are inputs to machine learning models that an attacker has intentionally intercepted and disguised to cause the prediction model to make mistakes [2].

Even though the field of adversarial machine learning has been active for more than a decade [3], most work has been conducted on classification problems. Adversarial regression, which is widely seen in cyber-physical system (CPS) settings, is still a relatively new topic. Moreover, since the discovery of adversarial examples in current deep neural networks in 2013 [4], security and robustness in state-of-art deep machine learning models has become a hot topic [5] and aroused significant concern. Consequently, even though a power load prediction model can be built easily using state-of-art deep learning techniques, a more cautious view still needs to be taken due to security issues that affect cyber-physical systems with integrated learning-based components [6].

In a nutshell, accurate load forecasting in smart grids is critical for managing infrastructure through targeted pricing and for maximizing profit. However, the hierarchical topology of power networks could lead to partial compromise of the network. Testing and realizing these kinds of security risks becomes difficult due to reliance on domain-specific knowledge for customizing the state-of-art machine learning techniques. To address these issues in the power system CPS domain we followed a model-based design approach [7] [8] [9] and developed our forecasting method and security testing framework using DeepForge [10].

In the following sections we attempt to put attention on the following objectives:

- Develop a model-based and cloud supported platform for rapidly designing and evaluating resilient learning/testing settings for sensor network architectures.
- Present a general step-by-step framework that can formalize the security and resilience testing in distributed sensor networks under adversarial settings.
- Design a generic procedure that can be utilized to implement stealthy adversarial attacks on machine learning

978-1-7281-2135-2/19/$31.00 ©2019 IEEE

predictors with the presence of self-checking detectors in sensor networks.
- Develop a case study for distributed power network load forecasting and demonstrate potential risks even with prediction procedures that incorporate anomaly detection algorithms.

The rest of the paper is organized as follows. Section II provides the motivation for evaluating security risks in the power system. Section III illustrates the theoretical background of our adversarial attack evaluation platform. Section IV presents a case study to demonstrate the capabilities of our platform by utilizing a power distribution network and simulating it with GridLAB-D. Finally, Section V concludes the paper and draws remarks for future directions.

II. Motivation

A cyber-physical system (CPS) is an intersection of computers and the physical world [11]. The two domains are connected with a multitude of sensors and/or actuators with dynamic system characteristics. In general, sensor networks are tightly integrated and utilized in a CPS for dynamic control and decision making. On the other hand, potential security risks in sensor networks can often be easily generalized and transferred to specific application scenarios. In particular, for critical large-scale infrastructures like smart grids, a large network of sensors are required to work together to support high-level decisions.

Machine learning techniques are required for smart CPSs to make decisions automatically and more adaptive to the environment. For a smart grid power network, load forecasting models are learned and applied to make predictions on future loads for efficient and profitable power system operations [12]. Research on load forecasting have been conducted over thirty years [13] and the critical role of load forecasting in smart grids have been demonstrated from various aspects [14].

In this paper, we attempt to explore the vulnerability of machine learning models in distributed sensor network settings. We consider how smart meter readings from different sources in a power distribution system are used for forecasting load using a pre-trained machine learning model. For a more realistic setting, we also use an anomaly detector (trained from the same dataset as the prediction model) to see whether input data deviates significantly from the nominal values.

In order to negatively impact the power system, the attacker can manipulate input data [15] to mislead the predictor. We set *reasonable* flexible constraint settings on the attacker due to the physical extent of the network and the cost of compromising individual sensors. In particular, we allow the attacker to modify a limited number of meter readings with an upper bound on the modification ratio of each meter value. These kinds of constraints enable the attack against the predictor to be stealthy enough to remain hidden from the anomaly detector that is used along with the predictor. In the following sections, we present an architecture for inference model learning and testing for sensor network settings, and demonstrate sensor network level adversarial attacks using a power distribution network load forecasting case.

III. Methodology

In this section we provide more details on the underlying methodologies of our framework. We use state-of-art tools of TensorFlow/Keras to build generic executable pipelines to show the prediction and attack tests under flexible settings. This approach enables easier collaboration and potentially more effective dissemination of testing results.

A. Model-Based Framework

To bridge the gap between domain specific application data and machine learning, our framework builds on DeepForge [10]. DeepForge is a model-based and cloud-based collaborative development environment with deeply integrated domain specific modeling features created using WebGME [16]. It combines model integrated computing with rapid prototypical development of machine learning models.

DeepForge presents machine learning models with four hierarchical concepts: Pipelines, Operations, Executions and Jobs. *Operations* are atomic functions which accept inputs and return outputs. A *Pipeline* refers to a stack of tasks, such as data pre-processing, training or testing. Executing pipelines results in the creation of *Executions*. A *Job* corresponds to the selected operation along with its run status and metadata associated with its execution. DeepForge provides built-in extensions to the state-of-art Keras/TensorFlow machine learning framework. It enables real-time collaboration between modelers and analysts and uses strict version control for reproducible experiments.

To unify the data representation,we use a generalized input and processing data format for networked sensor data. Each data input is a two-dimensional matrix consisting of data from n sensors involving T time steps. As this is a simple notation, there may be optional pre-processing steps required to transform raw input data into this format. For the grid load forecast case, the data of power meter sensor readings are numerical values and our testbed allows normalization on input data to ease sequential processing steps.

To support the power domain we developed several reusable *atomic* operations such as data gathering using GridLAB-D simulator, pre-processing data, model training, and testing. We developed hybrid adversarial attack methods ranging from single-step FGSM [5] to more complex adversarial attack settings. Figure 1 shows the key atomic operations (operation screenshot truncated) provided by our framework and a sample adversarial attack pipeline constructed using these components. For testing purposes, we also embed a pre-trained prediction model as well as a detection model for the load forecasting case study.

Ten step-by-step pipelines ranging from simple train/test workflows to more complex attack/detection evaluations are incorporated. All of these pipelines can be utilized and modified easily. This effectively provides a highly user-friendly abstraction for domain-specific deep-learning applications.

We formalize the application and security testing procedure pipelines for sensor networks commonly used in CPS and divide them into the following five major parts and provide

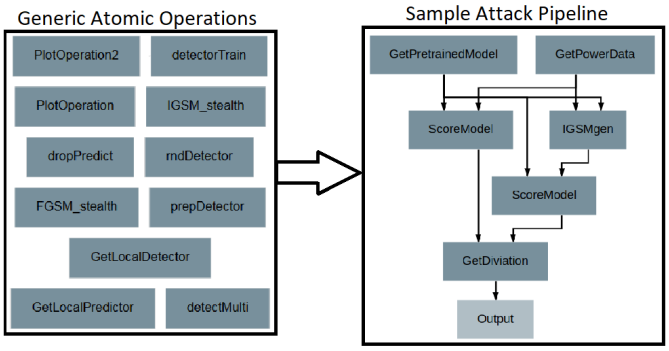

Figure 1: Creating Attack Pipelines using Generic Atomic Operations

targeted executable pipelines using Tensorflow and Keras to generalize them for different scenarios:

- **Basic predictor training/testing**: This is the overall goal and a crucial step in the machine learning model for the sensor network data implementation. A predictor training and evaluation pipeline is provided for generic use.
- **Adversarial attack on the prediction model**: This involves a worst white-box attack setting that allows users to test the robustness of the predictor. A few different adversarial settings are pre-built as basic pipelines targeting maximizing prediction deviations from original predictions or simply maximizing or minimizing prediction results.
- **Anomaly detector for the system**: In practical uses, the user also holds an auto-encoder model to detect anomalous sensors or to denoise the noisy sensor data. For the sensor network, we propose an LSTM auto-encoder concerning the general requirement for multiple sensors involving multiple time steps.
- **Stealthy adversarial attacks**: Considering the existence of an anomaly detector, the attacker can currently conduct stealthy adversarial attacks to deviate the predictor as well as evading the anomaly detector. Stealthy attacks need to consider both maximizing prediction loss as well as evading the detector.
- **Evaluation of attacks and defenses**: Allow the user to test different system settings to explore the robustness of the detection and prediction system. A general randomization-based exploration tool is provided to experiment on potential strategies.

B. Predictor Model

Predicting the total demand of the distribution system is essentially a multi-variate time-series regression task. We solve the prediction problem using state-of-art methods based on *Recurrent Neural Networks (RNNs)*. In particular, we use a Long-Short-Term Memory (LSTM) network [17], which accepts data sequences as inputs and returns a sequence of predictions. There is no universal standard for a good prediction model in realistic tasks. However, for regression problems, typically the statistical metric of mean squared error (MSE) is used. We use MSE for evaluating our models.

C. Anomaly Detector

In our literature survey, we did not find a detailed explanation for the origin of adversarial examples. However, we could gather approximate reasoning of adversarial examples from 'unexplored spaces' or 'over-explored spaces' from higher dimensions [18]. This enabled us to determine key characteristics of adversarial examples from a statistical point of view.

In this way, we can view adversarial examples as some kind of anomaly that needs to be detected. Previous research conducted in detecting adversarial examples [19] [20] mostly focused on image classification tasks, such as *MNIST* digit recognition or *Cifar-10* object classification. A recent trend of detection seeks to make this detector construction process more automatic via generative models [21] [22]. The basic intuition behind these detection techniques is to judge whether the input sample is likely to be from the normal distribution of the sample data [23].

Using machine learning models for anomaly detection in CPS has become more popular recently due to improvements in both deep-learning techniques and computation power. Researchers have experimented with auto-encoders [24] [25] [26] and generative adversarial networks [27]. It is worth pointing out that in CPS, the input data formats vary significantly, which limits creation of a universal detection framework. Therefore, we can only seek domain-specific or even case-specific solutions [26]. Here, for numerical data coming from distributed sensors, we use an auto-encoder to build an anomaly detector.

Auto-encoder models learn internal representations with the objective $f(x) = x$ mapping to the input itself. In order to detect anomaly using an auto-encoder, a common procedure is to set statistical thresholds for the residual between the original input and the reconstructed input. Most detection cases only need to give binary outputs for the whole input sample. For the sensor network in our case study, we expect the detector to find whether specific sensors in the network are likely to be compromised [28].

We set individual detection thresholds for each sensor meter in the network using a simple procedure. After training the auto-encoder using the training data, we use the training data to compute the fitting error (MSE) for all sensors and using maximum MSE 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 potential for adversarial attacks.

D. Stealthy Adversarial Attack

There are two important premises for a successful adversarial attack. First, the attack should not be detected by the machine learning system it is attacking. Secondly, the attack should cause worse performance of the machine learning prediction system. Based on these two general requirements, an adversarial

attack can be formulated 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.

1) l_0-FGSM Attack: Among the various attack methods developed so far, one of the most well-known and popular is the FGSM (Fast Gradient Sign Method) [5] which formulates the optimization using only a 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 cost function used to train the neural network. The magnitude constraint added to the original sample is represented by ϵ. This method is quite simple and intuitive. The attacker makes modifications to maximize the loss function, meanwhile as the modification is small enough it actually preserves the original information structure. It is worth pointing out this attack method regard the requirement of being stealthy as self-evident under the magnitude constraint of ϵ.

We adapt our algorithm to CPS, which are highly complex with a data input space that is potentially bigger than a fixed range. *Algorithm 1* shows a single attack step to deviate this predictor. Note that each meter (value in our input vector) has its unique regression case because the input range may not be in a fixed area. Therefore, the deviation is denoted as a ratio rather than a fixed value. And the number of meters allowed to be modified is also limited.

Algorithm 1 l_0-FGSM Attack

Require: $\boldsymbol{x}_0$: original observation; f: predictor; $J(f, x)$: cost function of predictor f according to input data x; D: l_∞ max deviation ratio; N: set of meters can be modified; $allowModN$: data selector to get values from a set of meters.

$\Delta \boldsymbol{x} \leftarrow \mathbf{0}$, $i \leftarrow 0$
$grad \leftarrow \nabla J(f, \boldsymbol{x}_0)$
$grad \leftarrow allowModN(grad, N)$
$\Delta \boldsymbol{x} \leftarrow D * \boldsymbol{x} * sign(grad)$
return $\boldsymbol{x} + \Delta \boldsymbol{x}$

2) Stealthy l_0-IGSM Attack: Algorithm 1 generates simple single-step attacks, which suffer from two limitations. First, the non-linearity of the predictor itself makes it almost impossible for a single-step to reach the optimal loss increase. On the other hand, an out-of-distribution detector can detect those sensors which are modified large enough and thus nullify the adversarial effects. For certain cases with detectors, the attacker can reformulate the optimization problem into a joint form [29] to solve for the joint adversarial perturbation. However, the detector we are using provides thresholds for individual sensors. which makes it difficult for such piecewise functions to be formed together.

To address these two limitations, we apply an iterative approach to reach for a more optimal adversarial perturbation. In practice, for each $\boldsymbol{x}_0$, our approach performs $NumIter$ number of iterations with maximum deviation ratio D, takes steps of step length ratio $(1 + D)^{(1/NumIter)}$, and computes $\Delta \boldsymbol{x}$ using the gradient sign method starting from the result of previous iteration. With the existence of a detector, intermediate results are first checked with the detector to remove exposed parts and then sent into the next iteration for further exploration. *Algorithm 2* shows how we generate the stealthy attack.

Algorithm 2 Iterative Stealthy Attack

Require: $\boldsymbol{x}_0$: original observation; f: predictor; d: detector; $J(f, x)$: cost function of predictor f according to input data x; D: max. deviation ratio; N: number of sensors allowed modifying; $allowModN$: data selector to get values from a set of meters; $NumIter$: number of iterations.

$\Delta \boldsymbol{x} \leftarrow \mathbf{0}$, $i \leftarrow 0$
$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$
$\alpha \leftarrow (1 + D)^{(1/NumIter)}$
while $i < NumIter$ **do**
  $grad0 \leftarrow \nabla J(f, \boldsymbol{x})$
  $grad \leftarrow allowModN(grad0, N)$
  $\Delta \boldsymbol{x} \leftarrow \alpha * \boldsymbol{x} * sign(grad)$
  $\Delta N \leftarrow d(\boldsymbol{x} + \Delta \boldsymbol{x})$
  $grad \leftarrow allowModN(grad0, N - \Delta N)$
  $\Delta \boldsymbol{x} \leftarrow \alpha * \boldsymbol{x} * sign(grad)$
  $\boldsymbol{x} \leftarrow \boldsymbol{x} + \Delta \boldsymbol{x}$
end while
return $\boldsymbol{x}$

Essentially, Algorithm 2 is a hybrid attack combining L_0 and L_∞ constraints and the detection threshold constraints. In practice, an attacker is allowed to modify Num meter readings with a maximum deviation ratio D. For each input $\boldsymbol{x}_0$, we choose the most sensitive, but undetected, meters using their gradient values for each iteration. Figure 2 shows a high-level execution pipeline as an example for a stealthy attack we have implemented. In Figure 2, input artifacts like data and pre-trained models are shown in light blue and operations in dark blue. An attacker fetches the pre-trained prediction models and data to attack and uses the 'IGSM_stealth' operation to generate adversarial perturbations. A generic prediction evaluation operation of 'ScoreModel' is used on both original and adversarial samples. The adversarial impact can be computed later based on the deviation from these two metric results shown as the overall output. The integrity of model-based tools provides a simple framework combining different types of practical constraints and a low computation complexity for fast prototypical tests.

IV. Experiment Results

For demonstration purposes, we utilize a case study based on a medium-scale power distribution network over time. It is worth noting that the prediction and attack evaluation strategies

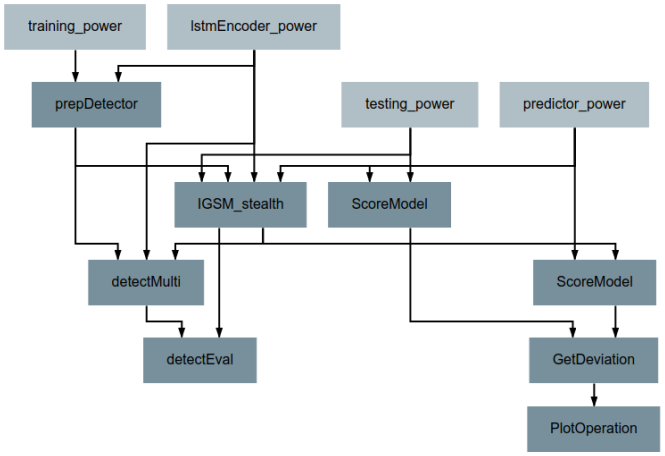

Figure 2: Stealthy Adversarial Attack Pipeline

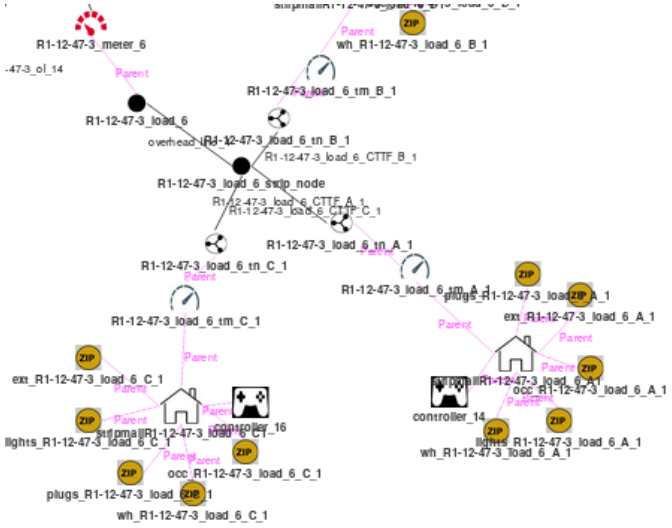

Figure 3: Part of the Power Distribution Network

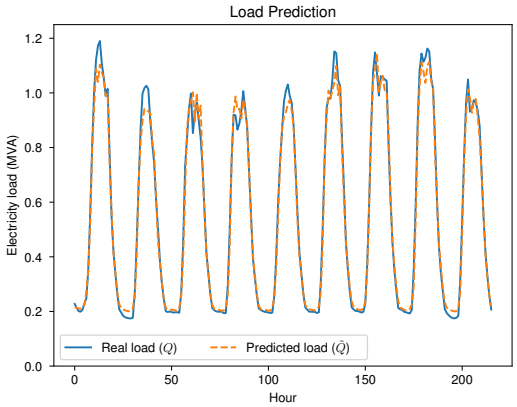

Figure 4: Load Prediction Illustration

can also be generalized to other distributed sensor network settings with no barrier.

A. *Power System Setting*

For this case study, we make a detailed simulation of an electric distribution system using GridLAB-D and the prototypical distribution feeder model provided by the Pacific Northwest National Laboratory (PNNL) [30]. The distribution model captures the fundamental characteristics of distribution utilities in the US. Figure 3 shows the topological structure of the power distribution network. The figure shows the load data collection mechanism for the distributed smart meter setting. Meters are connected to the user households and their usage data reports are transmitted to the control center in a hierarchical manner. In this case, we use the prototypical feeder *R1-12.47-3*, which represents a moderately populated area. Furthermore, we added representative residential loads to the distribution model using the script in [31]. In summary, our distribution model has 109 commercial and residential loads, which in-turn include appliances such as heating, ventilation, and air conditioning (HVAC) systems, water heaters, and pool pumps. GridLAB-D allows us to model the response of the loads to weather and market's prices, giving realism to the simulations.

For each hourly time step, the predictor takes loads from meter readings in the past 24 hours and also takes into account the temperature data during the same period of time. After absorbing and formatting these inputs, the predictor gives out the future total network load for the next hour. This prediction is conducted iteratively to provide references for dynamic pricing or facility maintenance.

B. *Results*

We use DeepForge to build a load forecasting model for this power distribution network using techniques of LSTM and deep neural networks. We use the historical data from 109 loads (both commercial and residential) and the temperature during summer time (June to August) as inputs. In this case, we use the measurements of the past 24 hours to predict the total load during the next hour. We make predictions every hour. Owing to the large-scale dataset generated for our problem, we chose a relatively large-scale neural network that includes three LSTM layers (with 150 units) and two fully connected layers (with 500 units). Figure 4 shows an example of the load prediction. The predictor is trained from 90% of the total data and leads to a mean squared error (MSE) of 0.1255 (*units:* Mega Volt-Amp) on the test dataset.

Along with the predictor, the power management system also holds an anomaly detector to detect whether sensors are working normally. We use a hard-decision detector which only outputs the binary classification results for sensors. The output for this detector is a vector with binary values indicating whether each meter reading in the network is likely to be an anomaly. Figure 6 shows the confusion matrix for the anomaly detector when 30% of meters in this network are added with a 20% level of Gaussian noise. The detector in our experiments shows an overall detection accuracy of 98.1 under random

Gaussian noise. However, as we show later, this detector is still vulnerable to carefully designed adversarial attacks.

In our attack scenario, the sensor data is manipulated under constraints. At every time step, the attacker can manipulate a fixed number of meters in the network (30% in our experiments). Moreover, for each meter, the attacker is allowed to deviate the meter reading by a constraint level of 20%. This is a modification setting that is equivalent to the noise level in the original detection test.

Figure 5 shows the modification ratio of one sample for the IterativeGSM attack. The horizontal axis denotes the meter index and vertical index indicates multiple data for one meter sample. We can see that in a total of 109 meters, 32 ($rate = 30\%$) meters can be modified by a ratio no more than 20%.

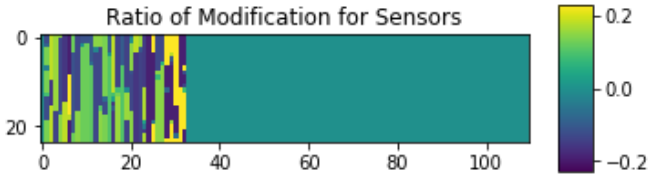

Figure 5: Modification Ratio for IGSM

Under same level of constraints, we explore four different adversarial attack settings (with and without detector):

- FGSM. Single step attack to maximize the deviation from the prediction
- IGSM. Iterative attack to maximize the deviation from the prediction
- DirectedGSM ($reverse = 1$). Iterative attack to minimize the predicted values
- DirectedGSM ($reverse = -1$). Iterative attack to maximize the predicted values

Figure 7 shows the confusion matrix for the anomaly detector when 30% of meters in this network are added with a level of 20% adversarial perturbation. The most critical metric here is the false positive (FP) part of the detection under adversarial perturbation. In contrast to the original detection results (see Figure 6), the ratio of undetected adversarial meters shows an increase from 2% to 98%. This clearly demonstrates that the generic stealthy attack procedure described above can evade static anomaly detectors with a very high success rate.

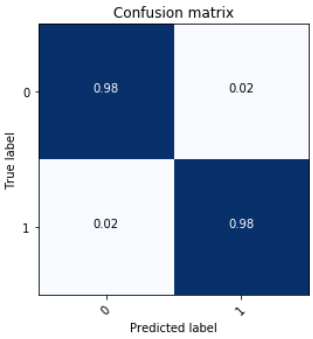

Figure 6: Detector Results using 30% Meters with 20% Level Gaussian Noise

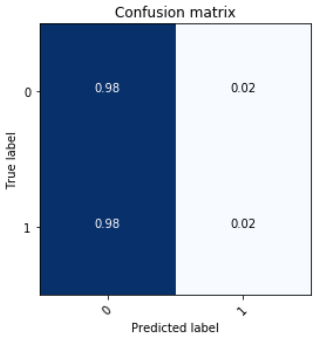

Figure 7: Detector Results using 30% Meters with 20% Level adversarial perturbation

To better evaluate the impact of adversarial attacks under detection settings, we updated the prediction step using a straightforward anomaly removal strategy. After the anomaly detection step, the sensor detected as abnormal from the input data space would be set as zero values. Thus, for the prediction, the input data is only relied on the remaining (most likely normal) sensors. In this way, the attacker only puts attention on the undetected part and tries to maximize the adversarial impact while keep the perturbation stealthy.

Table I shows experiment results under different attack and defense settings. The detection and anomaly removal strategy itself only brings in negligible deviations from the original predictions on normal data. Unfortunately, adversarial attacks with full knowledge of the detectors can find out the worst case for the predictor while still evading the detector. Our experiments show that the static detection and prediction mechanism remains vulnerable under adversarial settings even when only a small portion of sensors in the network are intentionally modified. A previous research [6] also explores the vulnerability of load forecast system under adversarial attack on weather data, where the significance of the single variable for weather is shown for the forecast design.

V. Conclusion and Future Work

In this paper, we demonstrated how to evaluate security and resilience of load forecasting predictors for a power distribution network. To enable rapid prototyping and evaluation, we introduced domain-specific abstractions in the model-based platform of DeepForge to construct a testing framework. To illustrate the capabilities of this framework, we used GridLAB-D for power network simulation, and provided various configurable and flexible security test settings in different forms. Our experiments showed that CPS that use machine learning techniques for load predictions can suffer from a worst-case attack even under a partial network compromise.

Our future work is focused on the following two aspects. First, our current investigation of the resilience of the power distribution network uses a static dataset generated from GridLAB-D. We plan to incorporate GridLAB-D as a more flexible DeepForge extension. This will enable GridLAB-D to work in a simulation-in-the-loop manner with simultaneous learning. This means users will be able to define simulations in a more flexible way that incorporates more kinds of user-defined uncertainties such as weather anomalies and dynamic network changes. Secondly, we also plan to explore the origins of the vulnerabilities in the prediction system under test and propose potential solutions for practical adversarial settings. We plan to integrate a randomization-based defense strategy exploration tool that can help generate resilient detection or mitigation defense mechanisms for various scenarios.

Table I: Prediction Results (MSE) with Different Prediction Deployment Settings

Attack/Detection Settings	Original/NoAttack	Adversarial/NoDetect	Original/StaticDetect	Adversarial/StaticDetect
Fast-GSM (rate=0.3,step_len=0.2)	0.1255	0.5375	0.1287	0.5322
Iterative-GSM (rate=0.3, step_ len=0.01,step_num=20)	0.1255	0.7801	0.1287	0.7606
DirectedGSM (rate=0.3, step_len=0.01 ,step_num=20, reverse=1)	0.1255	0.4785	0.1287	0.4913
DirectedGSM (rate=0.3, step_len=0.01 ,step_num=20, reverse=-1)	0.1255	1.025	0.1287	0.9899

VI. Acknowledgments

This work is supported in part by the NSF PIRE and CPS program under award #1521617, by NIST under awards #70NANB18H269 and #70NANB17H266, and by NSA SoS CPS Lablet award H98230-18-D-0010.

References

[1] R. Sevlian and R. Rajagopal, "A scaling law for short term load forecasting on varying levels of aggregation," *International Journal of Electrical Power & Energy Systems*, vol. 98, pp. 350–361, 2018.

[2] Y. Vorobeychik and M. Kantarcioglu, "Adversarial machine learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 12, no. 3, pp. 1–169, 2018.

[3] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

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

[6] Y. Chen, Y. Tan, and B. Zhang, "Exploiting vulnerabilities of load forecasting through adversarial attacks," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*. ACM, 2019, pp. 1–11.

[7] X. Koutsoukos, G. Karsai, A. Laszka, H. Neema, B. Potteiger, P. Volgyesi, Y. Vorobeychik, and J. Sztipanovits, "Sure: A modeling and simulation integration platform for evaluation of secure and resilient cyber–physical systems," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 93–112, 2018.

[8] H. Neema, J. Sztipanovits, M. Burns, and E. Griffor, "C2wt-te: A model-based open platform for integrated simulations of transactive smart grids," in *2016 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*. IEEE, 2016, pp. 1–6.

[9] H. Neema, P. Volgyesi, B. Potteiger, W. Emfinger, X. Koutsoukos, G. Karsai, Y. Vorobeychik, and J. Sztipanovits, "Sure: An experimentation and evaluation testbed for cps security and resilience: Demo abstract," in *Proceedings of the 7th International Conference on Cyber-Physical Systems*. IEEE Press, 2016, p. 27.

[10] B. Broll and J. Whitaker, "Deepforge: An open source, collaborative environment for reproducible deep learning," 2017.

[11] E. Lee, "The past, present and future of cyber-physical systems: A focus on models," *Sensors*, vol. 15, no. 3, pp. 4837–4869, 2015.

[12] A. R. Khan, A. Mahmood, A. Safdar, Z. A. Khan, and N. A. Khan, "Load forecasting, dynamic pricing and DSM in smart grid: A review," *Renewable and Sustainable Energy Reviews*, vol. 54, pp. 1311–1322, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.rser.2015.10.117

[13] A. P. Douglas, A. M. Breipohl, F. N. Lee, R. Adapa, and W. B. R. Norman, "Risk Due to Load Forecast Uncertainty in Short Term Power System Planning * School of Electrical and Computer Engineering , University of Oklahoma," *IEEE Transactions on Power Systems*, vol. 13, no. 4, pp. 1493–1499, 1998.

[14] R. Billinton and D. Huang, "Effects of load forecast uncertainty on bulk electric system reliability evaluation," *IEEE Transactions on Power Systems*, vol. 23, no. 2, pp. 418–425, 2008.

[15] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures," in *2010 First IEEE International Conference on Smart Grid Communications*. IEEE, 2010, pp. 220–225.

[16] Z. Lattmann, T. Kecskés, P. Meijer, G. Karsai, P. Völgyesi, and Á. Lédeczi, "Abstractions for modeling complex systems," in *International Symposium on Leveraging Applications of Formal Methods*. Springer, 2016, pp. 68–79.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1765–1773.

[19] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," *arXiv preprint arXiv:1702.04267*, 2017.

[20] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017, pp. 3–14.

[21] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 135–147.

[22] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-gan: Protecting classifiers against adversarial attacks using generative models," *arXiv preprint arXiv:1805.06605*, 2018.

[23] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems*, 2018, pp. 7167–7177.

[24] M. Cui, S. Member, J. Wang, S. Member, and M. Yue, "Machine Learning Based Anomaly Detection for Load Forecasting Under Cyberattacks," *IEEE Transactions on Smart Grid*, vol. PP, no. c, p. 1, 2018.

[25] J. Goh, S. Adepu, M. Tan, and L. Z. Shan, "Anomaly Detection in Cyber Physical Systems using Recurrent Neural Networks," *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, pp. 140–145, 2017.

[26] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," 2017.

[27] D. Li, D. Chen, L. Shi, B. Jin, J. Goh, and S.-K. Ng, "MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks," pp. 1–17, 2019. [Online]. Available: http://arxiv.org/abs/1901.04997

[28] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding," 2018.

[29] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," *arXiv preprint arXiv:1802.00420*, 2018.

[30] K. P. Schneider, Y. Chen, D. P. Chassin, R. G. Pratt, D. W. Engel, and S. E. Thompson, "Modern grid initiative distribution taxonomy final report," Pacific Northwest National Laboratory, Tech. Rep., 2008.

[31] https://github.com/gridlab-d/Taxonomy_Feeders, 2015, Accessed: Oct. 2019.