Adversarial Attack for Deep Learning Based IoT Appliance Classification Techniques

Abhijit Singh and Biplab Sikdar
Department of Electrical and Computer Engineering
National University of Singapore
Email: abhijit.singh@u.nus.edu, bsikdar@nus.edu.sg

***Abstract*—Simultaneous advancements in Internet of Things and Machine Learning have resulted in fascinating interdisciplinary applications, such as classification tasks based on smart device generated data for diverse applications such as resource allocation, security, and activity classification. However, such applications may be susceptible to attacks by adversarial examples. In this paper, we develop a white-box adversarial attack mechanism to generate adversarial examples for data obtained from smart-meters installed in residential houses, and demonstrate that their statistical properties are indistinguishable from those of the true datapoints. The attack mechanism focuses specifically on Deep Learning based models used to perform appliance classification in smart home environments. The statistical indistinguishability of the adversarial datapoints from the true datapoints indicates that non Machine Learning based solutions may not be able to tackle the challenge posed by adversarial examples. The effectiveness of the proposed techniques is demonstrated using the publicly available United Kingdom-Domestic Appliance-Level Electricity smart-meter dataset.**

***Index Terms*—Adversarial attacks, IoT, machine learning, cyber security.**

I. Introduction

There were several advancements in Internet of Things (IoT) and Machine Learning (ML) over the last few years. These advancements have resulted in burgeoning industrial interest in using these technologies to solve diverse real-world problems [1]. The benefits derived from IoT devices have prompted a new industrial age termed as Industry 4.0 [2]. Smart devices are an integral component of Industry 4.0, as they generate massive amounts of data which can reveal useful insights. This is usually achieved by using Artificial Intelligence (AI) and ML techniques such as deep learning to process the data generated by smart devices.

There are numerous applications and analysis techniques for data generated by IoT devices. Classification is one of the common methods used in IoT applications. Some examples of ML-based classification tasks for IoT devices include smart home applications such as energy management [3] and activity detection [4], smart city applications such as noise classification [5], vehicular traffic monitoring [6] and smart manufacturing [7], and security applications such as detection of unauthorized devices [8]. The use of ML-based classification techniques in these applications has demonstrated improvements in system performance and efficiency. As a result, there is surging interest in the development of ML-based solutions for a wide range of IoT applications.

While ML-based solutions have several advantages, they remain vulnerable to various attacks that may have serious consequences on the applications which deploy them. Some recent research has shown that neural networks, and specifically deep neural networks, are vulnerable to adversarial manipulation when using data from image [9] and audio [10] domains. This brings us to the first question addressed in this paper: *are deep learning methods used with IoT applications like smart-meters also susceptible to similar types of adversarial attacks?* Such attacks may force datapoints to be misclassified as belonging to a particular class, or in a broader sense, reduce the accuracy of the targeted ML classification model. For instance, if a particular datapoint has been adversarially manipulated, a classification model using smart-meter data as input may not correctly identify which appliances are functioning in a smart home. Although the practicality of such adversarial attacks has been illustrated in fields such as computer vision and image recognition, it has not been examined in the context of IoT data. The constraints governing adversarial examples are quite different in image-based domains and in data generated and used in IoT applications. Consequently, existing methods are not directly transferable. Since the use of deep learning to process data generated by smart devices used in IoT applications will only see a rise in the future, it is important to investigate the possibility of adversarial example attacks.

This paper addresses the task of generating adversarial attacks in IoT environments while focusing on smart home applications. Smart homes are a promising application of IoT and are expected to become omnipresent imminently. On top of the improved quality-of-life for the residents, there are solid economic reasons for this as well. An impact assessment study commissioned by the UK government illustrates this point effectively, as it found that adoption of smart-meters would lead to annual savings of £4.6 billion in the UK, due to judicious energy consumption [11]. These homes will be equipped with smart-meters (among other smart devices) which notify users which appliances are consuming most electricity. Behavioral research studies suggest that consumers manage their electricity consumption more competently when given appliance-by-appliance information [12]. Smart-meter data may even be used by suppliers to adjust their electrical production according to the usage patterns of their consumers, or introduce an appliance-specific dynamic pricing regime.

Such applications will require polished AI techniques such as deep learning based Non-Intrusive Load Monitoring from aggregated smart-meter data.

This paper presents a methodology to demonstrate that adversarial examples can be obtained for data generated by smart-meters, and that they are indistinguishable from true datapoints in terms of their statistical properties. *Our work is the first to investigate adversarial attacks on deep learning models used in IoT-based smart home environments.* We develop a white-box attack to generate adversarial datapoints for our use-case scenario (based on the publicly available UK-DALE (United Kingdom-Domestic Appliance-Level Electricity) dataset [13]). To summarize, the contributions of this paper are as follows:

- We develop a methodology to generate adversarial examples for smart-meter data obtained from smart home applications.
- We present an analysis to show that outlier detection methods will not be able to combat these adversarial examples. This demonstrates a need for training and deploying ML models for IoT applications which are robust against such adversarial attacks.

The rest of the paper is organized as follows. The next section provides a brief review of related work. Section III details the classification task, dataset used, model details, and the proposed adversarial attack mechanism. Section IV contains experimental results of our adversarial attack. Section V analyzes the results, and discusses some limitations of this work. Section VI concludes the paper.

II. Related Work

There is substantial literature in the field of computer vision on various types of adversarial attacks when using image-based inputs. The authors of [9] showed that deep neural networks for image classification were sensitive to negligible but targeted perturbations. Their optimization problem was framed as a search for an adversarial image which had minimal distortion when compared to the original image. They used the L-BFGS algorithm [14] to solve the objective function. This attack is quite resource-intensive as L-BFGS algorithm is a quasi-Newton method which requires the computation of a Hessian matrix.

The authors of [15] proposed the Fast Gradient Sign Method (FGSM), which is a one-step method to generate adversarial examples faster and in a less resource-intensive manner. It is a gradient-based method which controls the perturbations by using an L_∞ norm bound of $\|x - x'\|_\infty \leq \epsilon$. Adversarial examples ($x_{adv}$) not targeted at a particular class are generated by maximizing the loss function $F(x, y)$:

$$x_{adv} = x + \varepsilon sign(\nabla_x F(x, y)). \tag{1}$$

This strategy was expanded in [16] to incorporate targeted attacks by tweaking Equation 1 and maximizing the loss function of the target class y':

$$x_{adv} = x - \varepsilon sign(\nabla_x F(x, y')) \tag{2}$$

Although the FGSM attack is fast because it requires only one back-propagation step to generate adversarial examples, it is not very efficient. The authors of [17] improved on this by proposing an iterative version of FGSM.

The authors of [18] specifically focused on the decision boundaries of deep neural network image classifiers around a data point x. Their method of generating adversarial examples had the implicit assumption that each class in a neural network classifier can be separated from other classes by a hyperplane. Their experimental results showed that most of the datapoints were in the neighbourhood of the decision boundaries, which made them sensitive to minute perturbations.

There are few papers which have considered adversarial examples in the AI/ML in IoT domain. In [19], the authors studied adversarial attacks on Feed-forward Neural Networks (FNN) and Self-normalizing Neural Networks (SNN) used for classifying intrusion attacks on IoT networks. They used the BoT-IoT dataset [20] and concluded that while FNN outperformed SNN on the intrusion detection in IoT networks task, the FNN was less robust than SNN when adversarial examples were tested.

The authors of [21] focused on threats from adversarial examples to Covid-19 deep learning systems used in medical IoT devices. These models used image-based inputs. The authors tested various Covid-19 diagnostic methods that use deep learning with relevant adversarial examples, and found that these models were not robust to adversarial perturbations. Similarly, the authors of [22] also used image-based inputs, but they used adversarial perturbations to enhance privacy-preservation of photos stored on IoT devices. Their technique neglected the unimportant pixels of images to reduce image distortions of adversarial examples, and the experimental results show that their proposed method can fool neural network classifiers with just minute changes in visual effects.

Based on our literature survey, we note that, to the best of our knowledge, there is no existing work that has studied generation of adversarial examples in IoT applications in smart homes. Due to the difference in the data generated by these devices as compared to images considered in current literature, the attacker's strategy during the attack can have very different constraints. Thus, our work is novel and offers a useful contribution in an area of increasing importance, as it is crucial to understand the different aspects of security of AI and ML models used to analyze varying types of IoT datasets.

III. Attack Methodology

In this section we present our proposed methodology for generating adversarial examples in smart home scenarios using smart-meter data. We start with an overview of the ML task, the dataset used, and the adversary model, followed by the proposed methodology.

A. Appliance Classification Task

Increasing deployment of smart-meters in residential homes has allowed the collection of fine-grained power consumption data that can be used for various applications. Smart-meter

data can be used to detect and classify appliances being used in a household, which may then facilitate: (i) methods for demand-side management and dynamic pricing mechanisms, (ii) better load forecasting, (iii) consumer profiling and classification, and (iv) feedback to consumers on their usage patterns. Further, appliance usage data can be used for understanding the behavioral and occupancy patterns of the members of the household, and used for purposes such as commercial, emergency, marketing, and law enforcement services [23]. Traditionally, Non-Intrusive Load Monitoring (NILM) and Non-Intrusive Appliance Load Monitoring (NIALM) techniques for disaggregating power consumption into that of individual appliances are based on statistical methods such as maximum likelihood estimation and change detection [24].

In recent times, ML techniques have been used for appliance classification using smart-meter data. This paper considers the deep learning based classification model proposed in [25]. Its efficacy has been demonstrated on the UK-DALE dataset [13]. The NILM strategy proposed in [25] is aimed at household appliance identification using residential smart-meter data and is based on training a deep learning model to perform this classification task, with substantial pre-processing of data. In [25], the multi-class appliance classification task is transformed into a binary classification task. The data is first pre-processed to obtain the inputs that will be used in the deep learning model. This involves splitting the data for each of the possible appliances, and then using the models to predict whether that particular appliance was in use or not. Essentially, a new model has to be trained to classify each appliance.

Figure 1 shows the basic architecture for the classification technique used in [25], which remained the same for all the appliances. The deep learning model uses three hidden layers with 500 neurons in each layer, followed by a linear output layer. The ReLU activation function [26] was used in the hidden layers. The loss function was a log-softmax function, and the optimizer used was Adam [27] with a learning rate of 0.001. Since the model used for the appliance classification task is not the primary focus of this paper, interested readers can refer to our code at [28] for any further details (or to reproduce the results presented in this paper). Note that while the authors of [25] used TensorFlow [29] to build their deep learning architecture, our code is implemented using PyTorch [30].

B. Dataset

This paper uses the UK-DALE dataset [13] to evaluate the proposed adversarial attack generation mechanism. This dataset was also used by the authors of [25] to test their deep learning based NILM method. This publicly available dataset has aggregated and disaggregated appliance data from five households in London, collected over several years. For each house, the whole-house mains power demand and appliance-level power demand are recorded every six seconds using a smart-meter.

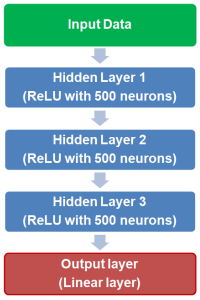

Fig. 1: Model architecture used for the classification task.

TABLE I: Number of training and test datapoints for each appliance (after pre-processing steps of [25])

Appliance	Train datapoints	Test datapoints
Kettle	57310	1000
Vacuum Cleaner	29177	1000
Microwave	125966	1000
Hair Dryer	44535	1000
Toaster	48504	1000

We note that a large fraction of the data comes from just a single house. House 1 has data from November 2012 to April 2017, which is a period of roughly four and a half years. Houses 2-5 have significantly lesser data, which is discontinuous in patches (up to 2 months long). House 2 has data from March 2013 to October 2013, House 3 has just 1 month of data (March 2013), House 4 also has data from March 2013 to October 2013, while House 5 has data from July 2014 to November 2014.

Five appliances are presented for analysis in this paper: kettle, vacuum cleaner, microwave, hair dryer, and toaster. All five of these appliances have labeled data available in the UK-DALE dataset. Table I presents the number of training and test datapoints for each appliance after the pre-processing steps of [25].

C. Adversary Model

This paper considers a practical adversary or threat model for deep learning algorithms used for classification tasks with smart-meter data. The threat model is as follows:

- The adversary can arbitrarily modify the smart-meter readings. When the adversary has physical access to the meter, and/or it has compromised the encryption keys used (if any) and it has compromised any of the network elements, then such attacks are possible.
- We consider a white-box attack where the adversary has access to the trained model. Such attacks are possible, for example, in the case of insider attacks.

D. Adversarial Attack Mechanism

The optimization to be performed by the adversary to generate adversarial examples is formally framed as follows. Assume that we have a function $f(x)$ which classifies the input $x \in R^d$ (a d-dimensional real-valued vector) into one of k possible classes. The adversary seeks to add a perturbation r

Algorithm 1 Generating an adversarial datapoint

Input: True datapoint (*X*), Target class (*target_y*), Trained model (*model*)
Output: Adversarial datapoint (*X_fooling*)
Other variables: Learning Rate (*learning_rate*), Max no. of iterations (*max_iter*), Class scores generated by *model* (*model_score*), Predicted label (*predicted_label*), Score of target class (*target_score*), Gradient of *X_fooling* (*dx*)

1: **function** GEN_ADV_DP(*X, target_y, model*)
2: Use model in evaluation/inference mode
3: Initialize adversarial datapoint (copy of true datapoint: *X_fooling* = *X*)
4: Set *learning_rate* and *max_iter*
5: **for** $i <$ *max_iter* **do** ▷ (loop for gradient ascent)
6: Compute *model_score* of *X_fooling*
7: Extract *predicted_label* from *model_score*
8: **if** *predicted_label* = *target_y* **then**
9: break from for-loop
10: **else**
11: Extract *target_score* from *model_score*
12: Perform backpropagation on *target_score*
13: Extract gradient update of *X_fooling (dx)*
14: Normalize the gradient update: *dx* = *learning_rate* × (*dx*/norm(*dx*))
15: Update *X_fooling* with normalized *dx*: *X_fooling* += *dx* ▷ (gradient ascent)
16: Clear current gradients
17: **end if**
18: **end for**
19: **end function**
20: Return *X_fooling*

to a datapoint x so that it is misclassified. Then, the objective of the adversary can be modeled as:

$$\min \quad \|r\|_2 , \tag{3}$$

$$\text{subject to:} \quad f(x+r) = k_{adversarial\_target} \tag{4}$$

where $k_{adversarial\_target}$ is not equal to $k_{predicted\_class}$, because equality would make it trivial. The formulation above aims to find the smallest adversarial perturbation r which would classify the datapoint x into a target class of the adversary's choice. Since the models targeted in this paper use a binary classification scheme, the optimization in the problem above seeks to flip the true class of the datapoint.

The overall algorithm for executing the adversarial attack mechanism is shown in Algorithm 1. It can be thought of as gradient ascent on the target class, because we are performing a gradient ascent using the model output to execute the above optimization. One of the key steps in this algorithm is step 6. The model score contains real-valued numbers associated with each class. The predicted class is the class which has the highest score associated with it in the *model_score* array. If the predicted class is not the same as the desired target class (step 8), then we extract the real-valued number associated with the desired target class from the *model_score* array,

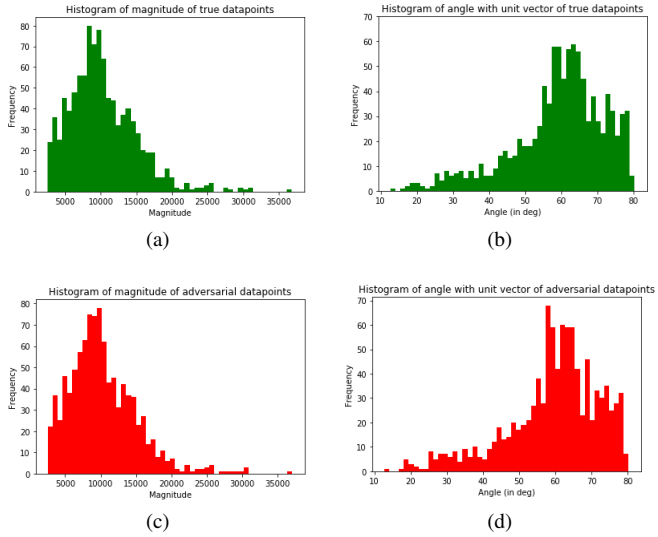

Fig. 2: Histogram of magnitudes ((a) and (c)) and angle with unit vector ((b) and (d)) of true datapoints (green histograms) and adversarial datapoints (red histograms) (when interpreted as a vector in a high-dimensional space). Appliance: Kettle.

and perform backpropagation (step 12). Then, we use the normalized gradient updates to nudge the adversarial datapoint in the desired direction iteratively (steps 13-16), until Equation (4) is satisfied (or we exceed *max_iter*).

IV. Evaluation of Attack Strategy

This section presents the experimental evaluation of the adversarial attack mechanism proposed in Algorithm 1. We were able to successfully generate adversarial examples for all 1000 test datapoints for each of the five appliances. Figures 2 to 6 captured the trend of all the true and adversarial datapoints for each of the appliances respectively. These figures show the histogram of magnitudes and angle with respect to a chosen unit vector for each of the datapoints, when they are interpreted as a vector in a high-dimensional space. The unit vector we chose had the same magnitude in each dimension. The magnitude of each dimension was given by $\frac{1}{\sqrt{d}}$, where d was the number of dimensions. So for 2-dimensional data, the unit vector would be $(\frac{1}{\sqrt{2}}\hat{i} + \frac{1}{\sqrt{2}}\hat{j})$. This scheme of selecting a unit vector was chosen because our objective was to understand and visualize the distribution of the datapoints with respect to a fixed point, and such a unit vector had non-zero magnitudes in all dimensions, serving our needs perfectly. In each figure, the green histograms represent true datapoints, and the red histograms represent adversarial datapoints. Sub-figures (a) and (c) are the histograms of magnitudes for true and adversarial datapoints respectively, and sub-figures (b) and (d) are the histograms of angle with respect to the chosen unit vector, for true and adversarial datapoints respectively. All sub-figures within each figure have the same range on x-axis and y-axis to facilitate easier comparisons.

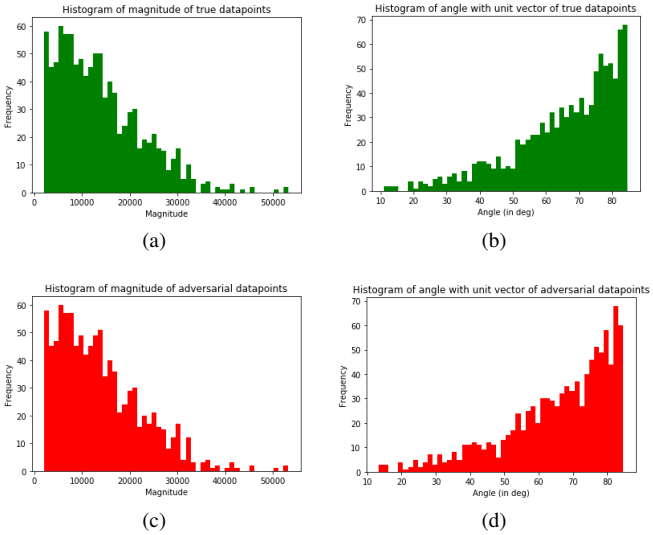

Fig. 3: Histogram of magnitudes ((a) and (c)) and angle with unit vector ((b) and (d)) of true datapoints (green histograms) and adversarial datapoints (red histograms) (when interpreted as a vector in a high-dimensional space). Appliance: Vacuum Cleaner.

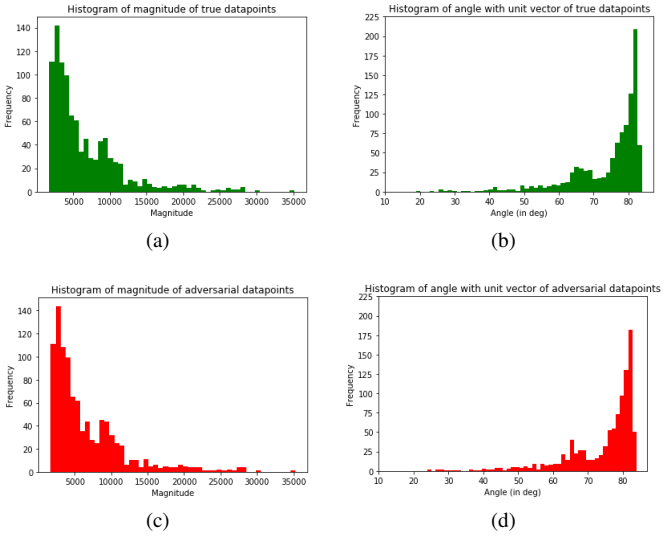

Fig. 4: Histogram of magnitudes ((a) and (c)) and angle with unit vector ((b) and (d)) of true datapoints (green histograms) and adversarial datapoints (red histograms) (when interpreted as a vector in a high-dimensional space). Appliance: Microwave.

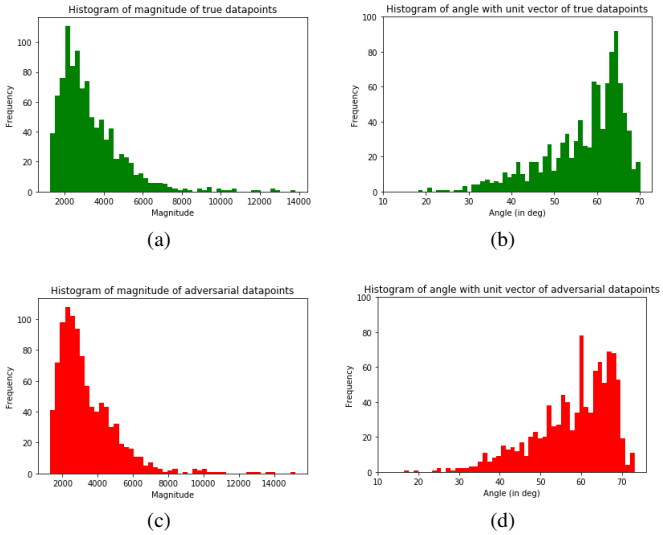

Fig. 5: Histogram Histogram of magnitudes ((a) and (c)) and angle with unit vector ((b) and (d)) of true datapoints (green histograms) and adversarial datapoints (red histograms) (when interpreted as a vector in a high-dimensional space). Appliance: Hair Dryer.

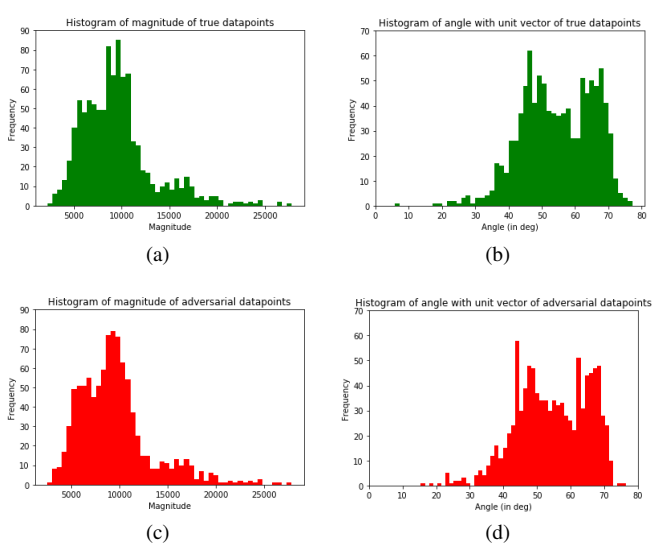

Fig. 6: Histogram of magnitudes ((a) and (c)) and angle with unit vector ((b) and (d)) of true datapoints (green histograms) and adversarial datapoints (red histograms) (when interpreted as a vector in a high-dimensional space). Appliance: Toaster.

V. Discussion

Since it's not possible to visualize high-dimensional spaces, in this paper we interpreted each datapoint as a vector in its original high-dimensional space. Then we compared the magnitude profile and angle made with a suitable unit vector profile for both the true and adversarial datapoints. Our results (Figures 2 to 6) show that the true datapoints and adversarial datapoints have very similar profiles, and almost identical ranges for both magnitude and angle. This finding is important because it shows that these adversarial datapoints are not outliers in the original high-dimensional space. Thus, outlier

detection and filtering methods will not be able to tackle the problem posed by these adversarial datapoints. This underscores the need for developing adversarially-robust models for processing data generated by IoT devices like smart-meters. Further, if on-device computations are required for some applications due to concerns such as data privacy, then these adversarially-robust models must not be too computationally intensive to be deployed on edge devices.

There are certain limitations of this work which call for a discussion as well, and they coincide with possible future extensions. One conceivable future avenue would be to implement black-box adversarial attack mechanisms which only require access to the inference-time predictions of a model, and probe their effectiveness. In this paper, we use a white-box attack, which makes an assumption that it has full access to the trained model. This assumption may not always be valid. In any case, this work demonstrates that making the models used for such tasks available publicly can result in damaging consequences. Further, in this paper we did not investigate the potency of the adversarial attack when defense methods are used to protect the ML model. This is a viable extension which we will work on in the future.

VI. Conclusion

This study demonstrated that deep learning models used for classification tasks on smart-meter generated data are susceptible to adversarial examples, and it is possible to generate adversarial examples that are indistinguishable in terms of their deviation from true datapoints. We developed a gradient ascent strategy for generating such adversarial examples, and targeted models proposed in existing literature. We showed that the proposed attack mechanism is able to generate adversarial datapoints for every test datapoint of all the appliances. Further, we demonstrated that these adversarial datapoints were closely interlaced with the true datapoints, and thus non-ML methods like outlier detection would fail to distinguish them from true datapoints (without significant collateral damage). This points to a need for ML-based solutions.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & information systems engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[3] D. Koolen, N. Sadat-Razavi, and W. Ketter, "Machine learning for identifying demand patterns of home energy management systems with dynamic electricity pricing," *Applied Sciences*, vol. 7, no. 11, p. 1160, 2017.

[4] B. Das, D. J. Cook, N. C. Krishnan, and M. Schmitter-Edgecombe, "One-class classification-based real-time activity error detection in smart homes," *IEEE journal of selected topics in signal processing*, vol. 10, no. 5, pp. 914–923, 2016.

[5] Y. Alsouda, S. Pllana, and A. Kurti, "Iot-based urban noise identification using machine learning: Performance of svm, knn, bagging, and random forest," in *Proceedings of the International Conference on Omni-Layer Intelligent Systems*, 2019, pp. 62–67.

[6] M. Won, "Intelligent traffic monitoring systems for vehicle classification: A survey," *IEEE Access*, vol. 8, pp. 73 340–73 358, 2020.

[7] M. Ghahramani, Y. Qiao, M. Zhou, A. O. Hagan, and J. Sweeney, "Ai-based modeling and data-driven evaluation for smart manufacturing processes," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 4, pp. 1026–1037, 2020.

[8] V. Thangavelu, D. M. Divakaran, R. Sairam, S. S. Bhunia, and M. Gurusamy, "Deft: A distributed iot fingerprinting technique," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 940–952, 2018.

[9] 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.

[10] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 1–7.

[11] F. Lienert and M. Carson, "Smart meter roll-out for the domestic sector (gb)," *Department for Business, Energy & Industrial Strategy, Tech. Rep.*, 2016.

[12] C. Fischer, "Feedback on household electricity consumption: a tool for saving energy?" *Energy efficiency*, vol. 1, no. 1, pp. 79–104, 2008.

[13] J. Kelly and W. Knottenbelt, "The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes," *Scientific data*, vol. 2, no. 1, pp. 1–14, 2015.

[14] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

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

[16] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[17] A. Kurakin, I. Goodfellow, S. Bengio *et al.*, "Adversarial examples in the physical world," 2016.

[18] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[19] O. Ibitoye, O. Shafiq, and A. Matrawy, "Analyzing adversarial attacks against deep learning for intrusion detection in iot networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[20] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[21] A. Rahman, M. S. Hossain, N. A. Alrajeh, and F. Alsolami, "Adversarial examples–security threats to covid-19 deep learning systems in medical iot devices," *IEEE Internet of Things Journal*, 2020.

[22] X. Ding, S. Zhang, M. Song, X. Ding, and F. Li, "Towards invisible adversarial examples against dnn-based privacy leakage for internet of things," *IEEE Internet of Things Journal*, 2020.

[23] E. McKenna, I. Richardson, and M. Thomson, "Smart meter data: Balancing consumer privacy concerns with legitimate applications," *Energy Policy*, vol. 41, pp. 807–814, 2012.

[24] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[25] M. A. Devlin and B. P. Hayes, "Non-intrusive load monitoring and classification of activities of daily living using residential smart meter data," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 3, pp. 339–348, 2019.

[26] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] A. Singh and B. Sikdar. (2020, Dec.) Code related to the paper. [Online]. Available: https://www.ece.nus.edu.sg/stfpage/bsikdar/scripts/adversarial

[29] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 2019, pp. 8026–8037.