Black-Box Adversarial Attack for Deep Learning Classifiers in IoT Applications

Abhijit Singh
Department of Electrical and Computer Engineering
National University of Singapore
Singapore
abhijit.singh@u.nus.edu

Biplab Sikdar
Department of Electrical and Computer Engineering
National University of Singapore
Singapore
bsikdar@nus.edu.sg

***Abstract*—The increasing adoption of Internet of Things (IoT) has resulted in the availability of big data, which can reveal valuable insights if processed efficiently. Classification tasks are very important in such applications, and Artificial Intelligence is widely used to solve these problems. This paper demonstrates that Deep Learning classifiers used in IoT environments are vulnerable to black-box adversarial attacks. Such attacks can render these models ineffective by causing severe performance issues. This paper develops a black-box adversarial attack mechanism to generate adversarial examples for data obtained from smart meters installed in residential houses. An analysis is presented to demonstrate that the statistical properties of these adversarial examples are indistinguishable from those of the true examples, and the performance of the targeted models degrades sharply when exposed to the proposed attack. Further, the inherent properties of the attack mechanism imply that it is able to evade the entire class of gradient masking based defence methods. The effectiveness of the proposed black-box adversarial attack is demonstrated on the publicly available United Kingdom-Domestic Appliance-Level Electricity smart meter dataset.**

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

I. Introduction

Internet of Things (IoT) applications are increasingly using Artificial Intelligence (AI). The transition towards Industry 4.0 [1] is hastening the generation of vast quantities of data, and Machine Learning (ML) techniques such as deep learning are particularly useful in gaining new insights by efficiently processing large volumes of data.

Over the past few years, ML techniques have been used in diverse IoT applications such as energy management [2] and activity detection [3] in smart homes, load classification [4] and detection of anomalous meter readings or energy theft [5] in smart grids, and vehicular traffic monitoring [6] and noise classification [7] in smart cities. Using ML-based classification techniques in these scenarios has provided remarkable improvements in performance and efficiency of systems. As a result, there has been a rise in interest in the development of ML-based IoT solutions in a wide range of applications. One such area is Non-Intrusive Load Monitoring (NILM). NILM has been a difficult problem to solve using automated feature extraction methods like Deep Neural Networks (DNN). However, the authors of [8] have recently proposed NILM strategies which use deep learning models and get competitive results while addressing the household appliance classification problem.

While there has been a lot of research on using AI techniques such as deep learning to solve various complex tasks, not enough attention has been paid to the adversarial robustness of such solutions. The authors of [9] were the first to show that DNNs used for image classification tasks were sensitive to small distortions in the input. An adversary could use this to their advantage to generate targeted perturbations, which when added to the input, could force the DNN classifier into misclassifying that input. Following this development, several adversarial attacks were proposed for popular data domains such as image, audio [10] and text [11] data.

However, the vulnerability of ML models for IoT applications has not been as well-explored. As DNNs have been found to be one of the best performing model architectures, a majority of the ML models in IoT applications use DNNs for their classification tasks. This makes them particularly susceptible to adversarial attacks [12]. Such attacks may increase the error-rates of models by forcing the targeted models into classifying adversarial datapoints as belonging to a different class than its true class. For example, if an adversary targets an appliance classification model which uses smart meter data as input, it may misclassify an appliance A as being appliance B, which will severely impact its downstream tasks such as dynamic pricing or demand-side management, causing economic losses to the utility providers.

In this paper, a methodology is developed which shows that ML algorithms used in IoT applications are vulnerable to adversarial attacks. In particular, this paper considers smart meters in power grids. We show that adversarial examples may be generated for such environments without requiring any knowledge about the targeted oracle models or access to the data that they were trained on, and that the adversarial examples are indistinguishable from true examples in terms of their statistical properties. The effectiveness of the proposed attack methodology is demonstrated and validated on the public UK-DALE (United Kingdom-Domestic Appliance-

This research was supported by Ministry of Education, Singapore, under grants R-263-000-E78-114 and R-263-001-E78-114. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of Ministry of Education, Singapore.

Level Electricity) dataset [13]. To summarize, the contributions of this paper are as follows:

- This paper develops a black-box method for generating adversarial examples for data from smart meters that is used in smart home applications. The inherent properties of this methodology imply that an entire class of defensive schemes will fail to protect ML models from this attack.
- This paper presents an analysis which shows that outlier detection methods are unable to detect and protect against such adversarial examples. Thus, it is imperative to develop and implement ML models in IoT environments that are robust against adversarial attacks.

The paper's organization is as follows. Section II presents an overview of the related work. Descriptions of the classification task, datasets used for evaluation, threat model, and the proposed method for generating adversarial examples are presented in Section III. Section IV presents the experimental results of the adversarial attack. Section V analyzes and discusses the results. Section VI concludes the paper.

II. Related Work

Black-box adversarial attacks are a well-studied problem in the field of computer vision. There is extensive literature on different types of black-box attacks using popular image datasets such as MNIST [14], CIFAR-10 [15], and ImageNet [16]. The authors of [17] were the first to propose a black-box method for generating adversarial examples for deep neural network based image classifiers. They trained a local substitute model using synthetic data, which was obtained after implementing a Jacobian-based data augmentation method. The labels for the synthetic data were generated by querying the targeted oracle model. The substitute model was used to generate adversarial examples, which were then used to attack the oracle.

As most of the black-box attacks were based on transferability, [18] proposed a novel Gradient Estimation black-box attack. This attack used finite-differences method to estimate the gradient and required access to the probability vector generated by the targeted oracle model, rather than just the output label. However, each adversarial example required a large number of queries to the oracle, which scaled with the dimension d of the input.

The simplified attack proposed in [19] also required continuous-valued confidence scores from the oracle. This attack was based on the intuition that if there was only a small distance between the input and the decision boundary, then there was no need to be careful about the exact direction along which the adversarial datapoint must be pushed. While this method was query-efficient compared to the other methodologies at that time, it still required over a thousand queries to the oracle on average, before a successful adversarial example was generated.

In the domain of AI-powered applications in IoT, only a few papers have considered adversarial examples, and these have primarily been white-box adversarial attacks. In [20], the authors studied FGSM [21], Projected Gradient Descent (PGD) [22], and BIM [23] adversarial attacks on Feed-forward Neural Networks (FNN) and Self-normalizing Neural Networks (SNN) used for classifying intrusion attacks on IoT networks using the BoT-IoT dataset [24]. In [25], various attacks such as L-BFGS [9], FGSM [21], DeepFool [26], BIM [23] and PGD [22] were compared on Covid-19 deep learning systems used in medical IoT devices. The authors used the CORD dataset [27], and found that the deep learning models were not robust to adversarial perturbations.

Focusing specifically on smart meter datasets, the authors of [28] were the first to propose a white-box adversarial attack on deep learning models used for Non Intrusive Load Monitoring (NILM) applications. They demonstrated the effectiveness of their attack on models trained on UK-DALE and REFIT datasets. A black-box adversarial attack was proposed in [29] for a model trained on Almanac of Minutely Power (AMP) dataset [30] and Pecan Street dataset, which classified whether a furnace was in an active state or inactive state. The methodology in [29] was inspired from [17], and involved using data augmentation techniques on some datapoints used to train the oracle, to create synthetic data, and using this synthetic data to train a substitute model.

In the existing literature, there is no work that considers the problem of black-box adversarial attacks on deep learning based classification tasks in IoT environments, which require zero queries on the oracle or access to the data that it was trained on. Thus, this work makes a novel and important contribution in an area of growing importance.

III. Attack Methodology

In this section, we give a brief overview of the ML task, datasets used, threat model, and the proposed black-box methodology for generating adversarial examples for such scenarios.

A. Appliance Classification Task

Smart meters are being increasingly deployed in homes to collect power consumption data. This can facilitate detection and classification of appliances being used in a household, which can enable downstream tasks such as consumer profiling and classification, improved load forecasting, dynamic pricing mechanisms and methods for demand-side management. Traditional Non-Intrusive Load Monitoring (NILM) techniques for disaggregation of power consumption into that of individual appliances are based on statistical methods such as maximum likelihood estimation and change detection [31].

Recently, the authors of [8] have proposed deep learning based appliance classification methods. The NILM strategy proposed in [8] uses substantially pre-processed residential smart meter data to train a deep learning model to perform the appliance classification task. The standard practice in this area is to transform the task into a binary classification problem, because existing methods are unable to reliably decompose appliance profiles of multiple appliances from smart meter data at the same time. A high-level overview of this pipeline

is as follows: (i) extraction of appliance profiles from the aggregated data one at a time (the rest of the input is treated as noise), (ii) followed by training a model to predict whether that particular appliance was in use or not. Thus, a new model is trained to classify each appliance.

B. Dataset

The UK-DALE dataset [13] was used by the authors of [8] to test their deep learning based NILM method. It is a publicly available dataset that has aggregated and disaggregated appliance data collected over several years from five households in London. It has smart meter data from each household that captures the mains power demand for the whole-house, as well as appliance-level power demand, recorded at a granularity of six seconds. This paper considers the five appliances recorded in the dataset: kettle, vacuum cleaner, microwave, hair dryer, and toaster.

C. Threat Model

This paper considers the following realistic threat model for deep learning classifiers for use on smart meter data:

- The adversary has the ability to modify the data from smart meters. The adversary gains such capabilities if it can launch physical attacks on a smart meter, and/or it has cracked any encryption keys in use, and if it has compromised any network element such as access points and routers.
- We consider black-box attacks where the adversary has no knowledge of the targeted deep learning model (oracle), or access to the data that it was trained on. Additionally, in two of the four attack versions, the oracle is not queried at any stage. This allows for a stronger evaluation of the oracle's robustness and provides a lower bound on the adversary's effectiveness in terms of performance.

D. Adversarial Attack Mechanism

A black-box attack is one where the adversary has no information about the targeted deep learning model (oracle). In this paper, the adversary does not have access to the data that it was trained on either. To implement the proposed adversarial attack, a local substitute model is first trained on the same classification task as the oracle. The training data used for the substitute is distinct from the training data used for the oracle. Practically, this is achieved by using 80% of the available training data of the UK-DALE dataset to train the oracle model, and the rest of the 20% is used to train the substitute model.

Once the substitute model is trained, a white-box attack is performed on the substitute model to generate adversarial examples. For the attack versions which do not make any queries on the oracle, the attack follows Algorithm 1. For the attack versions which make a query on the oracle, the attack follows Algorithm 2. Once the adversarial examples are generated, they are then used to attack the oracle.

In a classification problem that is solved using DNNs, the model tries to learn a function f that is parameterized by weights and hyperparameters θ:

Algorithm 1 Generating an adversarial example without querying the oracle

Input: True example (*X*), Target class (*y_tar*), Trained substitute model (*sub_mod*)
Output: Adversarial example (*X_adv*)
Other variables: Maximum iterations allowed (*iter_max*), Learning Rate (*lr*), Class scores generated by *sub_mod* (*sub_mod_sc*), Predicted label (*label_pred*), Target class' score (*tar_sc*), Gradient of *X_adv* (*dx_adv*)

function CREATE_ADV_EX(*X, y_tar, sub_mod*)
  Use substitute model in evaluation/inference mode
  Initialize adversarial example (copy of true example: *X_adv* = *X*)
  Set *lr* and *iter_max*
  for $i <$ *iter_max* **do**
    Compute *sub_mod_sc* of *X_adv*
    Extract *label_pred* from *sub_mod_sc*
    if *label_pred* = *y_tar* **then**
      break from for-loop
    else
      Extract *tar_sc* from *sub_mod_sc*
      Perform backpropagation on *tar_sc*
      Extract gradient update of *X_adv (dx_adv)*
      Normalize the gradient update: *r* = *lr* $\times$ (*dx_adv*/norm(*dx_adv*))
      Update *X_adv* with *r*: *X_adv* $+=$ *r*
      Clear current gradients
    end if
  end for
end function
Return *X_adv*

$$f(x;\theta) = y \tag{1}$$

where x and y are training inputs and labels respectively. In probabilistic terms, the training procedure can be expressed as trying to maximize the probability that θ models the underlying relationship between the training inputs x and labels y:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n} \log p(y_i|x_i,\theta) \tag{2}$$

where the training dataset is assumed to have n datapoints. After training, the trained weights and hyperparameters θ^* are obtained. During inference, these can then be used to make predictions on test datapoints x_{test} by computing:

$$\max \{p(y_{test} = k_i|x_{test},\theta^*)\}, \tag{3}$$

where $k_i = k_1, k_2, ..., k_c$ for c possible classes. Thus, the adversarial objective can be framed as:

$$\min \left\| (x_{test} - x^{'}_{test}) \right\|_2, \tag{4}$$

such that,

$$\max \{p(y_{test} = k_i|x^{'}_{test},\theta^*)\} = p(y_{test} = k_{adv}|x^{'}_{test},\theta^*)$$

Algorithm 2 Generating an adversarial example while querying the oracle

Input: True example (*X*), Target class (*y_tar*), Trained substitute model (*sub_mod*), Trained oracle model (*orc_mod*)
Output: Adversarial example (*X_adv*)
Other variables: Maximum iterations allowed (*iter_max*), Learning Rate (*lr*), Class scores generated by *sub_mod* (*sub_mod_sc*), Class scores generated by *orc_mod* (*orc_mod_sc*), Predicted label (*label_pred*), Score of target class (*tar_sc*), Gradient of *X_adv* (*dx_adv*)

1: **function** CREATE_ADV_EX(*X, y_tar, sub_mod, orc_mod*)
2: Use substitute and oracle model in evaluation/inference mode
3: Initialize adversarial example (copy of true example: *X_adv* = *X*)
4: Set *lr* and *iter_max*
5: **for** $i <$ *iter_max* **do**
6: Compute *sub_mod_sc* of *X_adv*
7: Compute *orc_mod_sc* of *X_adv*
8: Extract *label_pred* from *orc_mod_sc*
9: **if** *label_pred* = *y_tar* **then**
10: break from for-loop
11: **else**
12: Extract *tar_sc* from *sub_mod_sc*
13: Perform backpropagation on *tar_sc*
14: Extract gradient update of *X_adv (dx_adv)*
15: Normalize the gradient update: $r = lr \times (dx\_adv/\text{norm}(dx\_adv))$
16: Update *X_adv* with *r*: $X\_adv \mathrel{+}= r$
17: Clear current gradients
18: **end if**
19: **end for**
20: **end function**
21: Return *X_adv*

where k_{adv} is the class prediction as desired by the adversary, and $x^{'}_{test}$ is the adversarial datapoint. To generate $x^{'}_{test}$, Algorithms 1 and 2 are used. As they are both largely similar, we describe Algorithm 1 in detail, and then mention the key differences in Algorithm 2.

The algorithm begins by using the substitute model in inference mode, and making a copy of the true datapoint. This serves as the initial value of the adversarial datapoint. After setting a suitable learning rate, a loop begins where the completion criteria is checked at the top of the loop. The current value of the adversarial datapoint is used as input to the substitute model, and the predicted label is extracted from the model prediction. If the predicted label is the same as that desired by the adversary, the loop is exited and the adversarial datapoint is returned. If it is not, then the score of the target class is extracted from the model prediction and backpropagation gradients are computed starting from it. The gradient update of the input is then extracted, normalized, and multiplied with the learning rate. This value r (step 14) is then used to update the adversarial datapoint, and the process continues until either the adversarial objective is achieved, or the maximum number of iterations allowed are exceeded.

The only differences between Algorithms 1 and 2 are the addition of step 7 and modification of step 8 in Algorithm 2. In step 7, the model score generated by the oracle is computed, because this is used in step 8 to check whether the goal has been accomplished. Thus, while the adversarial example generation process is happening on the substitute, the completion check is done on the oracle. The oracle is only queried a maximum of four times for each datapoint.

TABLE I: Differences between the four attack versions used. Guide to acronyms: Ground-truth test set labels used to determine target class of adversarial datapoints (GT label), Oracle test set predictions used to determine target class of adversarial datapoints (OG label), attack versions 1-4 (v1-v4).

Attack Version	GT label	OG label	Oracle Queried
v1	✓	×	×
v2	✓	×	✓
v3	×	✓	×
v4	×	✓	✓

IV. Evaluation of Attack Strategy

This section presents the experimental evaluation of the adversarial attack strategy proposed in Algorithms 1 and 2. Table I summarizes the different attack versions. In attack v1, the ground-truth label of the test set datapoints is used when determining which class the adversarial datapoint should be predicted as, to consider that the oracle has been forced into a misclassification, and the oracle is never queried during the adversarial example generation process. In attack v2, the ground-truth test set labels are used again, but the oracle is queried a maximum of four times during the adversarial example generation process. Attack v3 is analogous to v1, with the only difference being that ground-truth labels of the test set datapoints are discarded, and instead, the test set predictions made by the oracle are used to determine the target class of adversarial datapoints. Similarly, the attack v4 is analogous to v2, with the difference that the test set predictions made by the oracle are used to determine the target class of the adversarial datapoints. Thus, attacks v1 and v3 use Algorithm 1, while attacks v2 and v4 use Algorithm 2.

Table II reports the percentage of test datapoints for which the adversarial examples generated on the substitute model were able to force the targeted oracle model into misclassifying them, for all the four attack versions.

To provide a visual representation of the results, Figures 1 and 2 compare the accuracy of the targeted oracle models with cases when they are attacked by adversarial datapoints generated in the different attack versions. Figure 1 compares the accuracy of the oracle models in no-attack scenario, under attack v1, and under attack v2, when ground-truth labels are used. Figure 2 compares the accuracy of the oracle models in no-attack scenario, under attack v3, and under attack v4, when oracle-generated labels are used (and as a result, the accuracy of the oracle is always 100%).

Figure 3 presents the histogram of magnitudes and angle with respect to a particular unit vector for each datapoint, when they are interpreted as a vector in a high-dimensional space, following the same scheme as that proposed in [28]. The histograms in green denote true datapoints, while the histograms in red denote adversarial datapoints. Sub-figures

TABLE II: Percentage of adversarial datapoints generated using substitute models which could force the oracle model into misclassifying them. Guide to acronyms: Appliances (Appl.), Kettle (KT), Vacuum Cleaner (VC), Microwave (MW), Hair Dryer (HD), Toaster (TS), attack versions 1-4 (v1-v4).

Appl.	v1	v2	v3	v4
KT	35.9%	71.3%	47.5%	76.7%
VC	30.0%	56.4%	21.8%	57.9%
MW	37.2%	60.9%	35.5%	54.7%
HD	45.9%	74.4%	52.5%	82.1%
TS	42.6%	68.6%	37.3%	69.7%

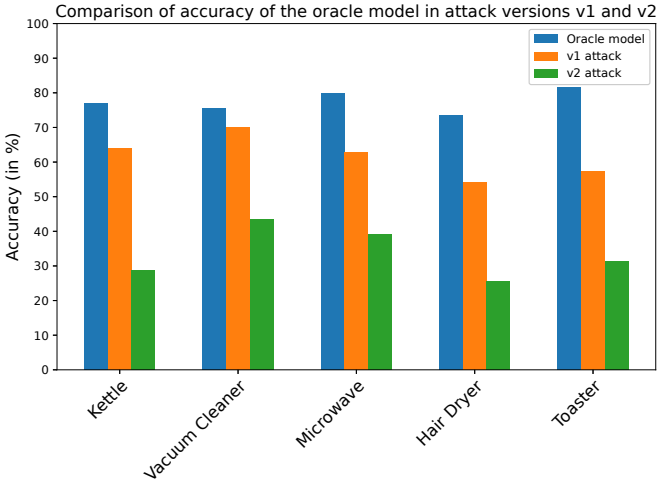

Fig. 1: Comparison of accuracy of the oracle models when using ground-truth labels in no-attack scenario (blue), under attack v1 (orange), and under attack v2 (green).

(a) and (c) show the histograms of magnitudes for true and adversarial datapoints, respectively, and sub-figures (b) and (d) show the histograms of angle with respect to the chosen unit vector, for true and adversarial datapoints, respectively. To enable easy comparison, sub-figures inside each figure use the same ranges for their x-axis and y-axis. We note that the histograms of the true and adversarial datapoints have a close resemblance to each other. The figure presented is for attack v1 on one appliance classifier. The figures for the other appliances and attack versions are similar, and are available for viewing in the linked code [32].

V. Discussion

The four attack versions were used to understand the impact of the adversarial attack from different perspectives. In attacks v1 and v2, since the ground-truth labels of the test were used to determine the target class of the adversarial datapoints, the results provide the actual number of misclassifications made by the oracle. However, there is also an argument for the fact that in real-world scenarios, ground-truth labels for new datapoints for which the oracle needs to make predictions will not be available. Thus, it is important to judge the efficacy of the adversarial attack based on how many adversarial datapoints were misclassified by the oracle, when the oracle's

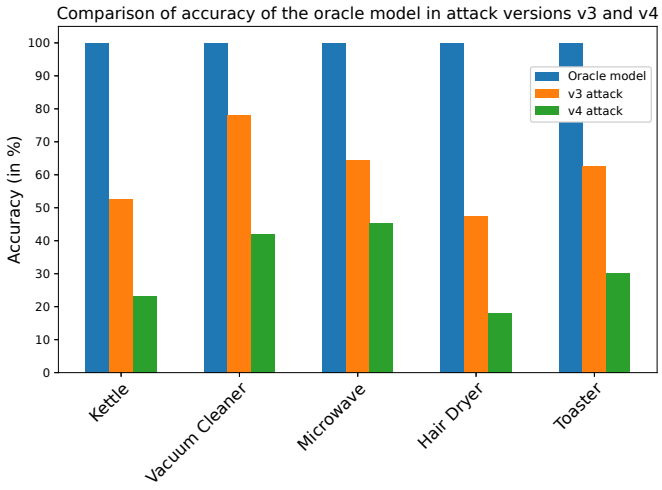

Fig. 2: Comparison of accuracy of the oracle models when using oracle-generated labels in no-attack scenario (blue), under attack v3 (orange), and under attack v4 (green).

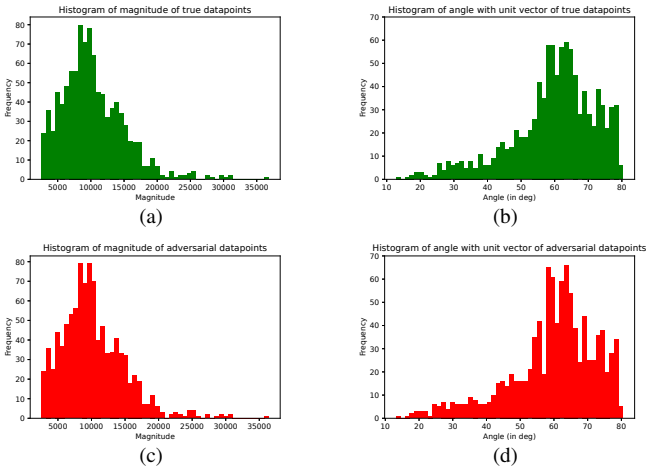

Fig. 3: Histograms of the magnitude ((a) and (c)) and angle with unit vector ((b) and (d)) for true (green) and adversarial (red) datapoints (when interpreted as a vector in a high-dimensional space). Appliance: Kettle. Other appliances can be found at [32].

predictions on true datapoints are considered to be the new "ground-truth". To do this, attacks v3 and v4 were used.

The results in Table II show the efficacy of the proposed methodology. It is clear that even attacks v1 and v3, which have the hardest constraints of not being able to query the oracle at all, have high rates of success of forcing the oracle models into misclassifying them. Once the constraints are relaxed in attacks v2 and v4, we see even higher rates of success, as expected.

Figures 1 and 2 make it easier to see the efficacy of the proposed adversarial attack. These figures graphically compare the accuracy of the oracle models in their original state versus when they are subjected to the four versions of the proposed

attack. We see that there is a sizable drop in accuracy when the oracle is subjected to attack v1, and a much steeper drop in attack v2. This remains the same for attacks v3 and v4 as well. With such low accuracies, the models will not be suitable for deployment in any applications.

Figure 3 shows that the true and adversarial datapoints are statistically indistinguishable from each other. As a result, outlier detection methods will not be able to defend against these adversarial datapoints.

An important aspect of the proposed adversarial attack methodology is that it does not require estimation of the oracle's gradient information. This means that it will be able to evade all defence methods that rely on variations of gradient masking, which gives the adversary an advantage over the defender. This illustrates the need to deploy strong defence methods, particularly those which are agnostic to the attack methodology used by an adversary, if ML models are to be deployed in large-scale applications.

VI. Conclusion

This paper demonstrated that ML classifiers trained on data generated in IoT environments, such as those from smart meters, are vulnerable to black-box adversarial attacks, and that the adversarial examples are statistically indistinguishable from true examples. This paper developed a strategy for generating adversarial examples on a locally trained substitute model which did not have access to the training data used for the oracle model or any knowledge about its model details, to target ML classifiers for NILM methods proposed in existing literature. The proposed attack methodology had high rates of success on all the appliances on a publicly available smart meter dataset. Further, as the proposed attack methodology does not require estimation of the oracle's gradients, it can evade an entire class of defence methods.

References

[1] 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.

[2] 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.

[3] 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.

[4] S.-l. Yang, C. Shen *et al.*, "A review of electric load classification in smart grid environment," *Renewable and Sustainable Energy Reviews*, vol. 24, pp. 103–110, 2013.

[5] R. Punmiya and S. Choe, "Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 2326–2329, 2019.

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

[7] 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.

[8] 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.

[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] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "Hotflip: White-box adversarial examples for text classification," *arXiv preprint arXiv:1712.06751*, 2017.

[12] A. Singh and B. Sikdar, "Adversarial attack for deep learning based iot appliance classification techniques," in *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*. IEEE, 2021, pp. 657–662.

[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] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[15] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[17] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506–519.

[18] A. N. Bhagoji, W. He, B. Li, and D. Song, "Practical black-box attacks on deep neural networks using efficient query mechanisms," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 154–169.

[19] C. Guo, J. Gardner, Y. You, A. G. Wilson, and K. Weinberger, "Simple black-box adversarial attacks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2484–2493.

[20] 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.

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

[22] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

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

[24] 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.

[25] 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.

[26] 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.

[27] L. L. Wang, K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Eide, K. Funk, R. Kinney, Z. Liu, W. Merrill *et al.*, "Cord-19: The covid-19 open research dataset," *ArXiv*, 2020.

[28] A. Singh and B. Sikdar, "Adversarial attack and defence strategies for deep learning based iot device classification techniques," *IEEE Internet of Things Journal*, 2021.

[29] J. Wang and P. Srikantha, "Stealthy black-box attacks on deep learning non-intrusive load monitoring models," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 3479–3492, 2021.

[30] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. V. Bajić, "Ampds: A public dataset for load disaggregation and eco-feedback research," in *2013 IEEE electrical power & energy conference*. IEEE, 2013, pp. 1–6.

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

[32] A. Singh and B. Sikdar. (2022, Apr.) Code related to the paper. [Online]. Available: https://www.ece.nus.edu.sg/stfpage/bsikdar/code/bb_attack_defence.zip