This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2021.3062722

Stealthy Black-box Attacks on Deep Learning Non-intrusive Load Monitoring Models

Junfei Wang, Student member, IEEE, Pirathayini Srikantha, Member, IEEE

***Abstract*—With the advent of the advanced metering infrastructure, electricity usage data is being continuously generated at large volumes by smart meters vastly deployed across the modern power grid. Electric power utility companies and third party entities such as smart home management solution providers gain significant insights into these datasets via machine learning (ML) models. These are then utilized to perform active/passive power demand management that fosters economical and sustainable electricity usage. Although ML models are powerful, these remain vulnerable to adversarial attacks. A novel stealthy black-box attack construction model is proposed that targets deep learning models utilized to perform non-intrusive load monitoring based on smart meter data. These attacks are practical as there is no assumption of the knowledge of training data, internal parameters, and architecture of the targeted ML model. The profound impact of the proposed stealthy attack constructions on energy analytics and decision-making processes is shown through comprehensive theoretical, practical, and comparative analysis. This work sheds light on vulnerabilities of ML models in the smart grid context and provides valuable insights for securely accommodating increasing prevalence of artificial intelligence in the modern power grid.**

I. Introduction

Today's electric grid is experiencing a major paradigm shift due to the information deluge induced by the proliferation of advanced monitoring and control devices. Grid measurements are generated continuously and in abundance by a large number of sensors (e.g. smart meters and phasor measurement units) deployed across the power grid. In order to capitalize on the insights contained in these datasets, data-driven approaches that leverage on machine learning (ML) constructs are becoming widely utilized by grid operators to perform analytics, predictions, and actuation.

This paper introduces a novel stealthy black-box attack construction that targets a ML model that performs non-intrusive load monitoring (NILM) on smart meter measurements. A smart meter reports the aggregate power consumption over an interval (e.g. minutely, hourly, etc.) of a dwelling. NILM disaggregates these readings into specific appliances that were active in the dwelling over the period under consideration without requiring the installation of sensors for individual appliances to detect the statuses of these devices. NILM offers tremendous insights into the power usage patterns of consumers and allows for the automation in energy applications that include home energy management systems and demand response programs that aim to increase economical and sustainable power usage [1]–[4]. NILM also allows for the detection of fraudulent activities such as illegal operations (e.g. marijuana growing) and electricity theft. NILM systems are typically implemented via ML models and these offer tremendous potential for enabling elevated situational awareness and timely incidence response in the power grid. However, these are also associated with vulnerabilities that can be exploited by adversaries to induce debilitating effects on power consumers and grid operations [5].

The ML-based NILM model considered in this paper takes in as input the power consumption of a household over an one-hour period (available through smart meter measurements) and operates on it to identify or compute the probabilities of specific appliances that have been active at the last minute of the input period. The proposed attack construction is not limited to the specific application of NILM and can be applied to any deep learning based ML model that takes in as input smart meter data and outputs discrete labels or class probabilities. The attack is constructed for the black-box scenario where the attacker does not rely on any internal knowledge of the ML model (e.g., parameters and/or architecture of the ML model) to craft the attack. This is a more generalized scenario as this eliminates the assumption that detailed knowledge of the attacked model is available. This is a practical adversarial attack pertinent to common smart grid applications (e.g., demand response, smart home energy management systems, etc.) that utilize deep learning based ML constructs for information processing and actuation. Moreover, the proposed attack construction is stealthy as it is designed to craft adversarial perturbations that can lead to the misclassification of targeted ML models without being detected by anomaly detection and error checking mechanisms deployed to identify malicious smart meter datasets.

There are two main phases involved in the proposed attack construction process: 1) Substitute model training that attempts to mimic the original ML model; and 2) Crafting perturbations to inputs using the substitute model that pass error checking mechanisms but result in erroneous outputs by the original ML model. As the attacker does not have access to the internal model parameters, he/she will strategically design a finite number of queries to the original ML model to construct the substitute model. As only a finite number of queries will be made to avoid attack detection, these must be augmented in a manner that allows the efficient recovery of the decision boundaries in the original ML model. Once the substitute model is trained, it is utilized to design minimal perturbations to real inputs to stealthily fool the targeted ML model.

Thus, the contributions of this paper are five-fold: 1) The

J. Wang is with the Department of Electrical and Computer Engineering, Western University, ON, Canada. P. Srikantha is with the Department of Electrical Engineering and Computer Science, York University, Toronto, ON, Canada.; E-mails: jwan577@uwo.ca and psrikan@yorku.ca.

(c) 2021 Crown Copyright. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2021.3062722

inputs of ML models targeted by the proposed black-box attack are smart meter readings, and outputs of these ML models are either discrete labels or probabilities which allows the proposed attack mechanism to target a wide range of ML models deployed in the smart grid context; 2) For the training of the substitute model, the proposed novel algorithm for augmenting the training dataset using a momentum-based method is more effective in identifying decision boundaries than the traditional Jacobian-based method proposed in [6]; 3) This work presents insights into the cost function selection, model selection and transferability for emulating the targeted model by the proposed black-box attack; 4) Adversarial perturbations to smart meter data are designed using the trained substitute model so that these are stealthy and successfully result in rendering incorrect outputs from the targeted ML model via a novel projected gradient based technique that incorporates confidence margins; and 5) The experiment section performs comprehensive comparisons with existing work (e.g. [6] and [7]) where the performance of the proposed attack construction for smart meter datasets is demonstrated.

The remainder of this paper is organized as follows. In Section II, a literature survey is presented that provides an exposition pertaining to existing work in the area of stealthy attack construction for deep learning models. Then in Section III, the methodology utilized in this paper is presented where details on the threat model and proposed attack strategy are described in detail. In Section IV, comprehensive results that demonstrate the efficacy of the proposal on different datasets along with comparisons to state-of-the-art are presented. Finally, the paper is concluded in Section V by discussing key insights from this work and future extensions.

II. Related Work

The notion of adversarial examples in the context of ML was initially identified in [8] where discontinuities in input-output mappings of image classification ML models were exploited to construct imperceptible perturbations to input data that will lead to misclassifications by the targeted ML model. Following this, a wide body of literature that includes [9]–[11] focussed on white-box ML attack constructions. The main assumption made by these white-box attack constructions is that the full knowledge of the targeted ML model is available to adversaries. Various techniques utilized for attack constructions include: iterative local linearization of the ML model to craft effective input perturbations [9]; construction of a saliency map using the Jacobian of the targeted ML model to perturb the most sensitive input components (e.g. [10] [11]). A more practical mode of attack is the black-box attack where internal parameters or architectures of the ML models being attacked are not known. As such, in [6], the authors attempted to convert the black-box problem into a white-box problem by training a substitute model that mimics the original ML model using a Jacobian data augmentation algorithm. Then, this is utilized to construct adversarial perturbations via the fast gradient sign method (FGSM). A vast majority of literature on black-box attacks focuses on the application of image classification and few studies in the smart grid context exist.

As such, there are several recent proposals in the literature focusing on adversarial distortion of power signals generated in the smart grid [7], [12]–[14]. Niazazari and Livani [14] directly apply the technique proposed in [6] for attacking power event diagnostics ML models. Zhou et al. [7] propose white-box attacks on regression ML models designed for power grid load prediction which is a modified version of [6]. S. Ali, et al. [12] targets the power state estimation system by applying two existing adversarial example crafting algorithms. Y. Chen, et al [13] adopts local gradient estimation for both maximizing and minimizing load forecasting results by only tampering with weather features. The attack construction proposed in this paper fundamentally differs from existing work as it designs *black-box* adversarial perturbations targeting ML models operating on *smart meter* data. There exist fundamental differences in the properties of input datasets (e.g. images versus real power readings), outputs from the ML model under attack (e.g. classification versus regression) and the attack model (e.g. black-box versus white-box) which entail novel techniques for the crafting of successful stealthy attacks on the targeted ML model. The proposal is divided into two parts. The first part proposes a novel data augmentation technique to train an effective substitute model that adequately mimics the targeted ML model using finite number of queries. In the second part, a projected gradient approach is proposed that allows for the construction of adversarial perturbations that are both effective and stealthy. These are discussed in detail in the remainder of this paper.

III. Methodology

In this section, the threat model is first presented which highlights the system settings, assumptions and targeted ML model utilized in the construction of the proposed stealthy attack construction. Then, the proposed attack construction algorithm is detailed where specifics on the attack strategy and various elements of the black-box attack are described.

A. The Threat Model

In order to execute an adversarial attack, it is necessary to exploit an existing vulnerability in the system under consideration. A vulnerability is a system flaw that can be accessed and exploited via external entities [15].

In the system settings under consideration, aggregate power consumption by each consumer entity is recorded by smart meters and sent to the electric power utility companies (EPUs). EPUs then store this measurement data locally or in cloud locations. In the literature, assumptions that include the controlling single or a set of devices (e.g., smart meters) to tamper reading data [7] [16], compromising the communication infrastructure [17] [18], and directly infiltrating into the control/data centres of EPUs to modify locally stored data [19] [20] are commonly made. Attackers can launch proposed adversarial construction by exploiting any one of the assumptions of vulnerabilities. Thus, this paper aims to strategically craft perturbations that will be applied to smart meter readings via these access mechanisms to exploit vulnerabilities in ML models. These perturbations will bypass existing checks that

(c) 2021 Crown Copyright. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2021.3062722

are utilized by EPUs to ensure the integrity of the smart meter data (e.g. data filters, and pre-processing).

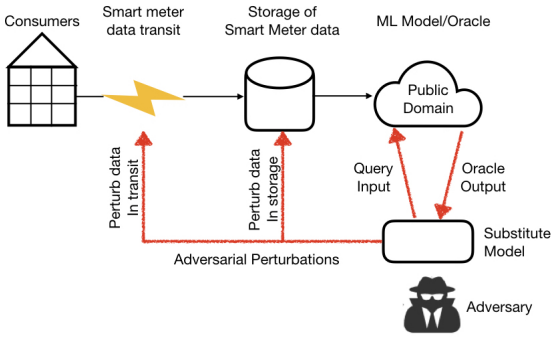


Fig. 1: The threat model.

This threat model is illustrated in Fig. 1 where the flow of smart meter measurements from the source (e.g., consumer home) to the EPU and various points of attacks are illustrated. The adversary will be a remote individual who is able to exploit vulnerabilities in communication protocols and software systems. NILM is typically employed to process smart meter data for making informed decisions. Adversaries will act to cause misinformation from NILM that will lead to erroneous decision-making in the power system. Three different modes of impacts of the proposed attack are outlined in the following. First is the adverse impact on demand response programs. NILM is used to measure the flexibility of consumers participating in demand response programs [2]. EPUs transmit demand side management signals to customers and remotely control flexible appliances identified by NILM [2]–[4]. Goals of demand response programs include the reduction of peak demands in the system via load shedding and appliance rescheduling. Misleading information regarding the active statuses of appliances will drive EPUs to misestimate the load flexibility in the grid and miscalculate control signals that can drive imbalance in demand and supply and contribute to the overloading of the distribution networks that can eventually result in cascading outages. The second mode of impact is related to HEMS that use NILM to schedule appliances' usage in residential buildings for reducing energy consumption [21] during peak periods. The proposed attack can lead HEMS systems to operate inefficiently. The third mode of impact is related to surveillance programs that use NILM to aid with identifying fraudulent activities (e.g. detect energy theft [22] and indoor marijuana growing operations [23]).

Due to the growing trend of ML being provisioned as a service in cloud computing, EPUs are empowered to train their ML models on the cloud and make them available in public hosts such as Google AI platform [24] without exposing original training data, internal parameters, and architecture (black-box model). A limited number of input-output queries to these ML models can be made by the general public. These queries are utilized by the adversary to train a local substitute model. This substitute model is then used to craft stealthy perturbations to smart meter readings. The adversary then applies these perturbations to smart meter measurements which are either in transit from the original source or stored in a cloud location by capitalizing on existing vulnerabilities in software and communication protocols. In practice, EPUs deploy error checking mechanisms to detect anomalies in smart meter readings. These involve the measuring of distances between actual readings and historical data distribution. Machine learning constructs such as regression [25] [26], clustering [27] [28], and generative [29] [30] based techniques are utilized to design these anomaly detection mechanisms. One-class SVM [31] and Autoencoder Forest [32] are real-world use cases deployed in Europe and Asia which are clustering-based and generative-based algorithms respectively. These utilize distance based measures to flag anomalous entries. These are associated with a threshold that distinguishes tampered data from legitimate data. Fraudulent activities such as power theft and meter tampering entail high thresholds that detect spikes or significant drops in the readings. For example, Liu and Nielsen [25] proposed a prediction-based anomaly detection framework to measure the distance between the prediction data and observed data. These techniques will be able to detect unusual outliers in the smart meter readings. There are still large margins between predicted data and the set thresholds even when the confidence parameter is set to be 95%. With our proposal, the crafted perturbation is designed to have minimal perturbations and fall within a specific deviation ratio across all dimensions of the data. This prevents the tampered data points from being flagged as outliers and the underlying patterns in the data will not be modified. Thus, existing error checking mechanisms deployed by EPUs will not be effective in flagging the perturbations generated by our attack strategy.

This threat is a practical reality and is illustrated via the following example. Consider the mobile application called Trickl that is released by London Hydro which is an EPU company in Ontario [33]. This application supports NILM in the pre-production phase and allows new queries to be executed every minute to infer the active status of appliances in a consumer's household. As such, an attacker can execute one query every minute and be well within the minimum query allocation for each minute. Even if 2000 queries are required to train the substitute model that approximates the targeted ML model, these queries can be easily executed over a period of 33 hours. This approximated model will then be utilized to craft stealthy adversarial perturbations.

1) Vulnerability, Access and Exploitation: The specific vulnerability considered in this paper is the inherent ambiguity between decision boundaries of ML models and the true decision boundary as illustrated in Fig. 2. Supervised ML models are trained using a finite number of training examples which are manually collected by domain experts to represent the "ground truth". Thus, generalization errors and feature selection problems reflect the inability of perfectly capturing the actual decision boundaries of the ground truth by the ML model being trained. These issues introduce "blindspots" or ambiguities that we capitalize on for the attack construction presented in this paper [34]. Fig. 2 conceptually illustrates this phenomena where for the same dataset, the ML model and the ground truth results in different outputs (i.e., regions lying in the non-overlapping areas). In these regions, adversaries can strategically modify target points so that these geometrically

(c) 2021 Crown Copyright. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2021.3062722

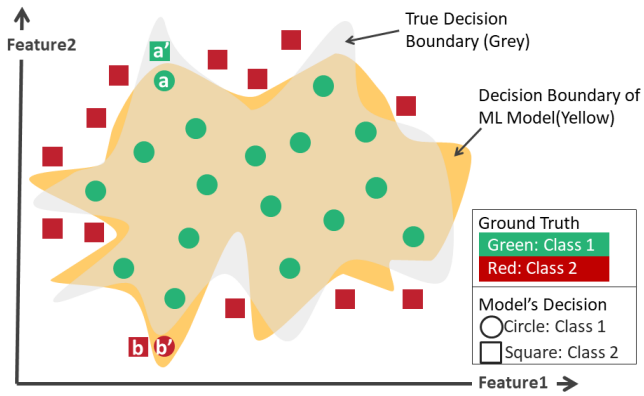


Fig. 2: Ambiguity in decision boundaries.

travel across a decision boundary and enters a 'blind zone'. These perturbations will be difficult to detect by the operator or error checking mechanisms while rendering the output to be different from ground truth. For instance, consider points a and b which are perturbed to be a' and b'. These adversarial perturbations will result in misclassification by the targeted ML model. Thus, in this paper, we treat the output from the targeted ML model when unperturbed/legitimate input data is passed to be the ground truth. In the specific application of NILM, strategic perturbations applied to the smart meter inputs can result in the corresponding ML model reporting appliances that were not active as active and vice versa. This can impose significant ramifications in smart grid applications such as billing, demand response and analytics.

2) Assumptions: This paper adopts the black-box attack model which is a more practical attack paradigm than the white-box paradigm. As such, ML models are typically trained internally by solution providers and public access to the inputs and corresponding outputs are made available to third-party entities that can access these models for various smart grid applications [35]. This public querying access is typically made available via cloud systems or application program interfaces (API) [35]. HTTP request-response protocol is one typical method by which queries can be executed [36]. For example, consider an ML model residing in the Google AI platform. For sending a query to this model by the general public, an HTTP request will be constructed where the credentials, ML model information, and a JSON format query payload consisting of the input to the ML model are included. Then, the ML model will return an HTTP response which will include the JSON format payload containing the prediction output corresponding to the input query. The internal components of the ML models along with the original training dataset are hidden from the public.

This implies that an adversary can pose as a third-party entity and query the targeted ML model (i.e. obtain the output for a specific input). The targeted ML model will be referred to as the Oracle in the remainder of this paper. Limited queries are made by the attacker to construct a substitute model that adequately imitates the Oracle. Training a substitute model transforms the attack construction from a black-box problem to white-box problem. Internal parameters of this substitute model are then utilized to construct stealthy perturbations that are applied to valid smart meter data to fool the Oracle.

Next, in order to apply the perturbations to smart meter data, an assumption is made that mechanisms for exploiting existing cyber-security flaws are available and this assumption is commonly made in smart grid security literature [37], [38]. Thus, this paper focused on the adversarial crafting of stealthy perturbations to smart meter data that successfully fool the Oracle. The perturbations are applied to smart meter measurements collected over an interval of 60 minutes when either in transit or in storage. The specific smart meter measurements attacked depends on the end goal of the adversary. For instance, if the sole purpose of the attack is to cause erroneous outputs from the Oracle, then the attacker will aim to perturb as many legitimate measurements possible. This work also makes no assumptions regarding the complexity of the targeted ML model. In fact, the NILM model considered for illustration purposes is based on a deep learning model.

3) Targeted ML Model for NILM: Next, the specific Oracle considered in this paper is presented. Although the proposed algorithm is applicable in the general smart grid context, this paper focuses on the NILM problem for demonstration purposes. In the literature, ML has been leveraged to perform NILM, and Wang et al. [1], Lan et al. [39], Mauch and Yang [40] are examples of some recent work in this area. This paper focuses on [1] for the Oracle. This NILM system is a deep learning based model which cannot be replicated easily when the parameters are unknown and thus allows us to showcase the efficacy of the proposed attack mechanism. Although this NILM model can be designed to be more efficient (i.e. output representation, etc.), improving it is not in the scope of this work and will be investigated in future work. This ML model takes in an interval (i.e. one hour) of minutely smart meter reading obtained for a household. Two different types of outputs are supported: one indicates the operational statuses of appliances (discrete) and the second is the probabilities. As a black-box approach is adopted, the only set of information an adversary will have of the Oracle is the dimension of the input $\vec{x}$ and the type of output y. Thus, the model can be defined as a mapping $f : \vec{x} \rightarrow y$ where $\vec{x}$ is the smart meter time-series reading and the output y is either discrete or probabilistic.

B. Stage 1: Substitute Model Construction

Here, the first stage of the attack construction is presented which is the construction of the substitute model (i.e. the approximation of the Oracle model). Considering a black-box attack construction, the internal details of the Oracle are unavailable to the adversary. In order to overcome this issue, a substitute model is constructed by the adversary to imitate the Oracle. When the substitute model closely represents the Oracle, the original black-box problem is transformed into a white-box problem. In this case, the internal parameters of the substitute model can be readily utilized by the adversary to fine-tune perturbations applied to legitimate smart meter readings that stealthily evade traditional validation mechanisms

(c) 2021 Crown Copyright. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2021.3062722

and successfully fool the Oracle. However, the design and training of the substitute model is not a trivial task as the amount of information available to the adversary about the Oracle is scarce. The only access an adversary has to the Oracle will be a limited number of queries. These queries are limited as the adversary will prefer not to draw attention to him/her via a large number of queries. Thus, these queries must be carefully crafted to obtain useful insights regarding the decision boundaries of the targeted ML model which can then be utilized to construct a representative substitute model.

1) Substitute Model Selection: First, the architecture of the substitute model constructed by the adversary is presented. As highlighted in recent literature, various internal architectures that include convolutional neural networks [39] [41] [42], recurrent neural networks [40] [43] [44], and auto-encoders [45] have been leveraged to construct ML models for NILM applications. One important objective is that the substitute model must be able to imitate any Oracle operating on smart meter data. To achieve this, the universal approximation theorem (UAE) is evoked which states that a feed-forward neural network (FFNN) will be able to approximate any smooth decision boundary given that a sufficient number of layers and nodes are incorporated into the network [46]. Thus, the internal architecture of the substitute model is selected to be FFNN. The output of the substitute model, $\hat{y}$, is the probability of each appliance being active over the last minute of the interval under consideration and thus is continuous. As the output layer of the substitute model is selected to be the softmax function, $\hat{y}$ will range between 0 and 1 (i.e. $\hat{y} \in (0, 1)$). Due to the continuous nature of the output, the adversary can utilize gradient-based approaches for computing the adversarial perturbations as discussed later in this paper.

Although the UAE justifies the use of FFNN, it does not provide any insights on the *learnability* of the substitute model [46]. Thus, the perturbations computed using the substitute model may result in fooling the substitute model but not the Oracle. When a perturbation results in successfully fooling both the substitute model and the Oracle, it is referred to as a *transferable* attack construction. The more similar the substitute model is to the Oracle, the greater will be the transferability of the attack construction. However, since the attack is a black-box construction, the similarity between the substitute and original models cannot be verified by the adversary. This paper draws upon the insights provided by [47] which demonstrates that the attack transferability depends on the complexity of the substitute model. The lower the complexity of the substitute model, the greater the transferability will be. This is illustrated in Fig. 3 where Fig. 3a illustrates the targeted ML model and its decision boundary. Fig. 3b illustrates the decision boundary computed by a more complex substitute model and Fig. 3c represents a less complex substitute model. It is clear from this example that the more complex model is composed of local optima where data points will not be transferable to the original model and this is not the case with the simpler model.

Next, the cost functions utilized by the adversary to tune the weight parameters in the substitute model are presented. The substitute model is an approximation of the Oracle by the

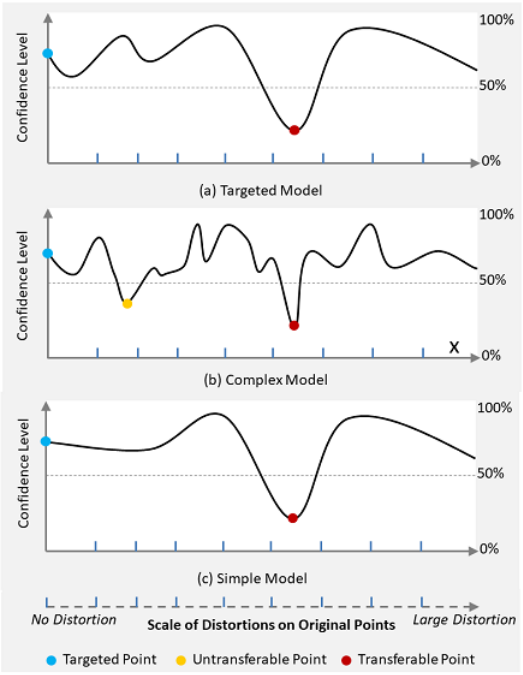


Fig. 3: Transferability based on model complexity.

adversary as the internal parameters, architecture and training construction of this model are not made available to the public. Thus, a generalized approach is necessary for the proposed attack mechanism to successfully approximate a wide variety of ML models that operate on smart meter data. As such, in the training of the substitute model, the loss function should be selected so that penalty is imposed for the output of the substitute model deviating from that of the Oracle. Including specific features such as consistency and temporal attributes will render the cost function specific for the application of the Oracle and this will not allow for generalizability. Two different loss functions are defined for the two different types of outputs considered. For the first type, the output of the Oracle is discrete (i.e. outputs 1 for the active class, while other classes are set to zero). The cost function for this case is selected to be the cross-entropy cost C defined in Equation 1:

$$C = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j}\log(\hat{y}_{i,j}) + (1 - y_{i,j})\log(1 - \hat{y}_{i,j}) \tag{1}$$

where N denotes the total number of samples in the training set, M denotes the total number of appliances under consideration. $y_{i,j}$ is the discrete output representing whether the j-th appliance is actually active (label of 1) or not (label of 0) for the input x_i. $\hat{y}_{i,j}$ is the corresponding probabilistic output from the substitute model. As the softmax function is used in the output layer, $\hat{y}_{i,j}$ will not take values 0 and 1.

The second type of output from the Oracle is probabilistic indicating the confidence of each appliance being active. Thus, the output for each appliance is a value ranging from 0 to 1. One example of a NILM model with four appliances is the set of outputs: $y_{i,1} = 0.9,\ y_{i,2} = 0.88,\ y_{i,3} = 0.01,\ y_{i,4} = 0.38$.

(c) 2021 Crown Copyright. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2021.3062722

We use a different loss function for the probabilistic case. If we use Eq. 1, then information will be lost when the probabilistic outputs are converted into discrete labels. According to reference [48], the Kullback-Leibler divergence metric allows for better transferability and knowledge distillation for probabilistic outputs. For this reason, we use this for training the substitute model for the case where the output from the Oracle is probabilistic. This cost function is defined in Equation 2:

$$C = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j}\log(\frac{y_{i,j}}{\hat{y}_{i,j}}) \tag{2}$$

where $y_{i,j}$ is the probabilistic output from the Oracle, and $\hat{y}_{i,j}$ is the substitute model's output.

2) Training Data Augmentation: In order to construct the training dataset, the adversary will execute a limited number of queries to the Oracle and utilize the output from the Oracle to augment or expand the training dataset to train the substitute model. A brute-force approach will require an infinite number of queries and will not be appropriate. A more efficient approach is utilized where the training dataset is initially populated with a small number of data points which are then utilized to iteratively craft synthetic data points that explore the decision boundaries in the targeted ML model more efficiently. The Jacobian dataset augmentation technique introduced in [6] is leveraged as a baseline for comparison purposes in this paper and this is referred to as the vanilla augmentation algorithm in the remainder of the paper. This algorithm is modified in order to propose the novel data augmentation algorithm that better suits the smart meter data.

Initial Dataset Collection Adversarial black-box attack constructions in the literature typically target image classification ML models. There exists an abundance of background knowledge on images which can be utilized to reconstruct representative initial data points. For instance, in a handwriting recognition model, the general structure of letters and numbers is common knowledge and can be easily utilized to construct a set of images that represent each class for the initial training points. However, with smart grid applications like NILM, insights regarding power signals are not readily available. Thus, in this case, the data points forming the initial dataset cannot rely on prior knowledge.

In order to overcome this issue, this work refines the goal of the data augmentation process. Instead of extracting a high-fidelity surrogate, the proposed augmentation process aims to iteratively draw better representation of the decision boundaries of the Oracle. Hence, the goal is not to recover the original training set to reproduce the Oracle model. The initial dataset will consist of simple data points that include constant power consumption over the 60 minute interval and power consumption that changes over 10 minute intervals. The only condition imposed on these data points is that when these are passed as queries to the Oracle, the outputs must be a balanced representation of various states of each appliance identified by the Oracle.

Vanilla Augmentation Algorithm Dataset augmentation algorithms utilized in the literature for adversarial blackbox attacks are generally composed of the following steps: 1) Iteratively generate synthetic datapoints, 2) Identify the output labels for these by passing these as inputs to the Oracle; and 3) Calibrate the substitute model to adjust to the augmented dataset. These three steps are repeated until the threshold set by the adversary for the maximum number of queries to the Oracle is met so that the decision boundaries in the substitute model approach the Oracle model.

The vanilla data augmentation algorithm introduced in [49] crafts synthetic training inputs by first identifying the directions in which the substitute model's output is varying and then applying an adjustment along the opposite of these directions to selected data points in the training set. The Jacobian matrix $J_{\hat{f}}$ of the function $\hat{f}$, where $\hat{f}$ represents the substitute model, contains information about these directions of change and is defined in Equation 3 [50]:

$$J_{\hat{f}} = \begin{bmatrix} \frac{\partial \hat{f}_1}{\partial x_1} & \cdots & \frac{\partial \hat{f}_1}{\partial x_p} \\ \vdots & \ddots & \vdots \\ \frac{\partial \hat{f}_k}{\partial x_1} & \cdots & \frac{\partial \hat{f}_k}{\partial x_p} \end{bmatrix} \tag{3}$$

where k is the number of output units in this model, and the (i, j) entry in $J_{\hat{f}}$ is the partial derivative of $\hat{f}$ with respect to the i^{th} output class and j^{th} component of the input $\vec{x} \in \mathbb{R}^p$. The new training sample crafted should represent the decision boundary of the Oracle. In order to realize this, it is necessary to identify the direction in which the output of the substitute model is least confident (i.e. direction in which the probability of an input belonging to the current class selected by the Oracle is lower). Let f denote the Oracle. The afore-mentioned logic results in the following rule in Equation 4 for the vanilla data augmentation technique:

$$S_{\rho+1} \leftarrow \{\vec{x} + \lambda\,\mathrm{sign}(J_{\hat{f}}[f(\vec{x})]) : \vec{x} \in S_\rho\} \cup S_\rho \tag{4}$$

where $S_{\rho+1}$ is the training set that is being currently augmented, ρ denotes the augmentation iteration, $\vec{x}$ is a training point obtained from S_ρ, $f(\vec{x})$ is the label obtained from the Oracle for the input $\vec{x}$, $sign$ is the function that returns 1 if the input is positive and -1 if the input is negative, $J_{\hat{f}}[f(\vec{x})]$ is the row of the gradient whose index corresponds to the class the Oracle maps to for input $\vec{x}$ and λ is a tuneable parameter which alternates between a negative and positive value every 3 iterations that allows for better exploration of the decision boundaries. When the output of the Oracle is probabilistic, a threshold is used to select the class label (i.e. if the probability is above the threshold, this class is active and inactive otherwise). This newly synthesized data point is then passed as input to the Oracle in order to obtain the label or confidence of the classes that this point belongs to. After every augmentation, the substitute model is retrained to account for the new point.

There are two main issues with the vanilla data augmentation algorithm. The exploration of decision boundaries is highly dependent on 1) The initial training dataset; and 2) Distribution of the original training set utilized for the Oracle. These pose a significant problem as constructing representative initial data points for the NILM problem is not trivial and the

(c) 2021 Crown Copyright. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2021.3062722

adversary has no knowledge of the dataset used to train the Oracle. Another issue is striking a balance with λ which is a parameter utilized to explore the decision boundaries. Smart grid problems like the NILM problem are composed of highly unbalanced data points (e.g. *inactive* state of an appliance like the furnace occurs majority of the time (i.e. more than 90%) compared to the *active* state). Thus, smaller values of λ will add incremental noise to the original training points and will not be effective in escaping the majority class for exploring the decision boundaries. On the other hand, larger values of λ will result in overshooting and not be able to explore the decision boundaries as dictated by the gradient directions.

Proposed Data Augmentation Algorithm This paper overcomes the afore-mentioned challenges by proposing a novel data augmentation algorithm that attempts to discover data points around the decision boundaries of the Oracle. When the training dataset is composed of sufficient data points around the decision boundaries, the substitute model can be trained to behave like the Oracle. Unlike the vanilla data augmentation algorithm proposed in [49], the proposal differs in three main regards: 1) A new data point is not augmented in only one step; 2) Momentum in addition to the gradient is considered in the augmentation step; and 3) A static change that is dictated by the fixed parameter λ is not utilized to perform the augmentation of a new data point.

As per the first difference, the proposed algorithm starts with a randomly selected point in the current training set and keeps adjusting it in the direction of least confidence so that this point crosses over from the current class to the next class in the currently trained substitute model. The vanilla algorithm applies only one update and does not search the space to find points that cross the decision boundary. This is illustrated in Fig. 4 (a). In Step 1, the vanilla algorithm

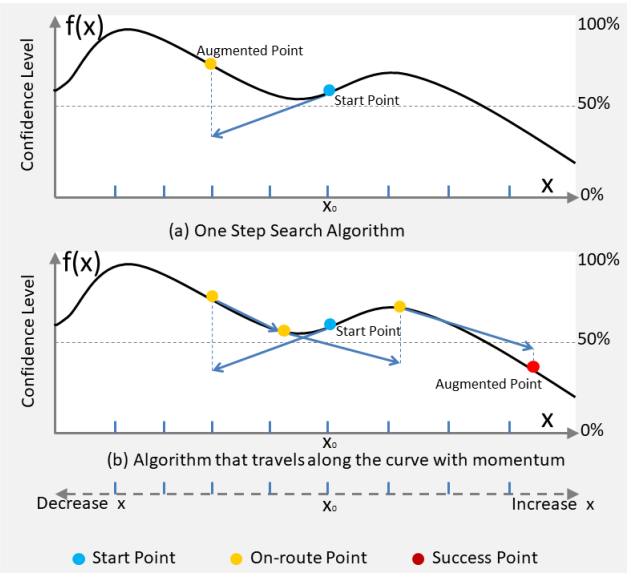


Fig. 4: Vanilla vs proposed data augmentation algorithm.

identifies the direction in which the confidence of belonging to the current class decreases. When an update is made in this direction, due to the fixed parameter λ, the update overshoots and lands at a point that is more confident than the original point. With the proposed algorithm, as illustrated in Fig. 4 (b), the point is repeatedly revised until it crosses over the decision boundary. As momentum is also considered in the updates and the updates are not forced to have a specific amplitude, the revised point is able to move across the decision boundary in a dynamic and accelerated manner without being stuck at local points with zero gradient. The proposed algorithm is detailed in Algorithm 1.

The intuition behind Algorithm 1 is illustrated via a simple example presented in Fig. 5. The initial training set of 5 random points are depicted as red and green dots in Fig. 5 (a). These are labeled via queries made to the Oracle model and used to train the initial substitute model presented in Fig. 5 (b) using solid curves. The goal to generate new points in Fig. 5 (c) is to discern ambiguous regions between decision boundaries of the two models, so each point moves in the direction of reduced confidence of belonging to the original class label as determined by the gradient of the substitute model. These points are adjusted until different output labels result from the substitute model. The yellow points in Fig. 5 (c) represent the final resting place of these points, and and shall be labeled appropriately using the Oracle, as shown in Fig. 5 (d). Unless the substitute model closely represents the Oracle, these newly augmented points can belong to either class as per the Oracle. The substitute model is then refined using the ten training points and the resultant decision boundary illustrated in Fig. 5 (d). It is clear through visual inspection that the refined substitute model better represents the Oracle's complex decision boundary. Thus, Algorithm 1 identifies differences between two models (i.e. two decision boundaries) and then retrains the model to eliminate them and better represent the Oracle.

In Algorithm 1, λ and α are parameters that represent step-size and the weight of the momentum, α is typically set to 0.9 to balance the contribution of the gradient term and the momentum, θ is the stopping criteria based on accuracy (i.e. performance of the substitute model on the synthetically generated dataset), and $\vec{x}$ is a training example contained within the current iteration of the training dataset S_ρ. At each augmentation iteration, the newly augmented data points in S_ρ are labeled by the Oracle. max_ρ is a parameter that imposes an upper limit on the number of queries that can be made to the Oracle. This limit can be made available by the hosting service (e.g. [24]) or be self-imposed by the adversary to prevent detection of the ongoing attack. These new points are utilized to retrain the substitute model $\hat{f}$ and points that result in misclassifications in the re-trained substitute model are discarded. Then, in the subsequent search for a new point within the nested *while* loop, the adjustment to the current data point is iteratively computed using the momentum term v and gradient $J_{\hat{f}}[y]$. It is important to note that in this update, the actual gradient is utilized for the update rather than the sign (e.g. vanilla algorithm). This update is applied in the direction of lower confidence of the updated point belonging to the current class. After the point crosses the decision boundary or if the maximum search iteration n is reached, the current augmentation iteration ends. If the new point results in a

(c) 2021 Crown Copyright. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2021.3062722

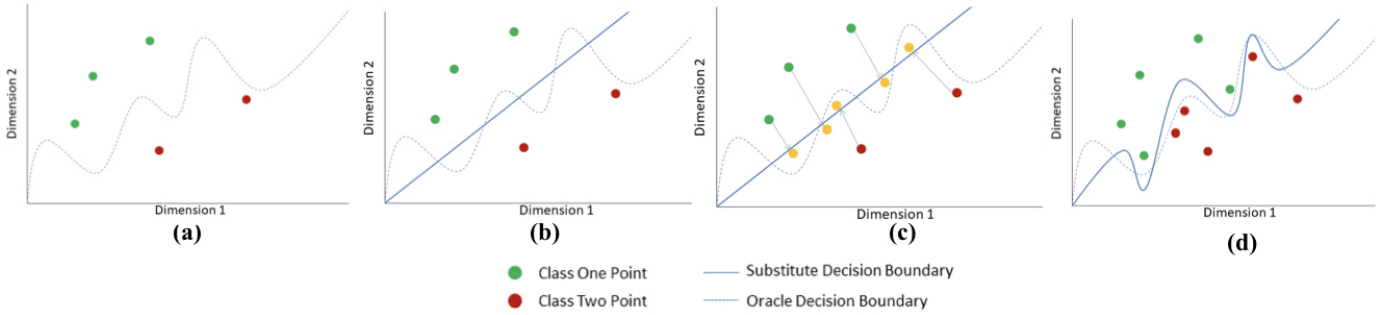


Fig. 5: Intuition behind Algorithm 1.

Algorithm 1 Substitute model training and data augmentation:

Input: $f, S_0, \lambda, \alpha, \theta$

Define $\hat{f}$
$\rho = 0$
loop
  $\hat{D} \leftarrow (\vec{x}, y \leftarrow f(\vec{x}))) : \vec{x} \in S_\rho$ ▷ Label the data
  if $accuracy(\hat{f}, \hat{D}) > \theta$ or $\rho > max_\rho$ **then**
    $break$
  end if
  $trainSubstitute(\hat{f}, \hat{D})$, $discardPoints(S_\rho, \hat{f})$
  for each $\vec{x} \in S_\rho$ **do**
    $\hat{y} \leftarrow \hat{f}(\vec{x}), \hat{v} = 0, i = 0$
    while $\hat{f}(\vec{x}) == \hat{y}$ and $i < n$ **do**
      $\hat{v} \leftarrow \alpha\hat{v} + \lambda\nabla_x J_{\hat{f}}[y]$
      $\vec{x} \leftarrow \vec{x} - \hat{v}$
      $i \leftarrow i + 1$
    end while
    if $\hat{f}(\vec{x})! = \hat{y}$ **then**
      $S_\rho \leftarrow \vec{x} \cup S_\rho$
    end if
  end for
  $\rho = \rho + 1$
end loop
return $S_\rho, \hat{f}$

change of class, the new point is added to the training set. This is repeated until the accuracy threshold θ is met.

C. Stage 2: Adversarial Perturbations

The main objective in the design of these perturbations to input data is to cause misclassifications by the Oracle with minimal detectability. Thus, these perturbations should not be apparent to error checking mechanisms that vet the smart meter measurements and this can be achieved by minimizing the magnitude of the perturbations. Mathematically, this problem amounts to the following:

$$\min_{\delta\vec{x}} \ |\delta\vec{x}|$$
$$\text{s.t. } \hat{f}(\vec{x} + \delta\vec{x}) \neq \hat{f}(\vec{x})$$

where δx is the adversarial perturbation applied to the original input $\vec{x}$. In the following, the fast gradient sign method (FGSM) proposed in [6] that is widely utilized in the literature to craft these adversarial perturbations is presented. Then, the problems associated with FGSM for the NILM problem is discussed. Then, the proposed adversarial perturbation algorithm is introduced.

FGSM utilizes the sign of the gradient of the cost function C of the substitute model $\hat{f}$ which is taken around the original input to devise the adversarial perturbations in Equation 5.

$$\delta_{\vec{x}} = \lambda \, \text{sign}(\nabla_{\vec{x}} C_{\hat{f}}) \tag{5}$$

where λ is the parameter that is tuned so that the label produced by the substitute model for the perturbed input data (i.e. $\vec{x} + \delta_{\vec{x}}$) changes from the original label to a different one and fools the Oracle. Although this is a very straightforward method, there exist three main problems with this approach for the NILM problem. One is that the magnitude of perturbation applied to each component or feature of $\vec{x}$ will be the same. Thus, even if one component of $\vec{x}$ need not be perturbed as much as the other component, it will experience the same magnitude of perturbation which can result in detection. The second issue is with the parameter λ which can be increased until the substitute model misclassifies the perturbed point resulting in easy detection of the attack. Thirdly, as there is no constraint imposed on λ, these perturbations can result in infeasible outputs (e.g. negative power readings) which can be easily detected.

Thus, in order to prevent detection, it is necessary to craft adversarial perturbations in a *stealthy* manner so that these attacks are not obvious and cannot be easily detected by error-checking mechanisms. Stealthiness is incorporated into the proposed attack algorithm via three main approaches. Firstly, in the proposed algorithm, the perturbations are constructed using actual cost gradients of the substitute model rather than the signs of these gradients. Secondly, the magnitude of perturbations that can be applied to the smart meter measurements is limited to a ratio threshold r. This way, there are no distinctive spikes or dips in the smart meter readings that can be detected by error checking mechanisms. Thirdly, a confidence margin constraint m is applied to ensure the transferability of the perturbation to the Oracle model from the substitute model. These are detailed in the following.

A projected gradient ascent (PGA) method is proposed that imposes a limit on the values each dimension of input $\vec{x}$ can take. The perturbation δx_i applied to each component i of $\vec{x}$ depends on the value taken by the gradient of the cost function

(c) 2021 Crown Copyright. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2021.3062722

C. The gradient can now be exactly calculated according to Equation 6 as the attacker has access to the internal parameters and architecture of the substitute model.

$$\delta_{\vec{x}} = \lambda \nabla_{\vec{x}} C_{\hat{f}} \tag{6}$$

where λ is the step size parameter. Then, each component of δx is subjected to an upper limit dictated by the maximum deviation ratio r. The maximum perturbation $\delta_{i_{max}}$ that can be applied to dimension i is defined in Equation 7.

$$\delta_{i_{max}} = x_i * r \tag{7}$$

Input perturbations are applied iteratively so that these remain within the boundaries defined by r while moving in the direction that increases the cost incurred by the substitute model. The updating rule is defined in Equation 8.

$$\vec{x} \leftarrow f_p(\vec{x} + \delta_{\vec{x}}, r) \tag{8}$$

where f_p is projection of the perturbed $\vec{x}$ onto the stealthy space defined by r. These updates are repeated until either the substitute model output changes from the original class to another class or the algorithm exceeds the maximum number of iterations n.

In reality, however, the substitute model is not an exact copy of the Oracle. In order to ensure that the effect of the perturbations applied to the substitute model transfer over to the Oracle, the notion of confidence margin m is introduced where the probability of the dominant output class of the perturbed input is higher than that of the dominant output class of the original unperturbed input by m. When constructing perturbations, this confidence margin is maintained. Algorithm 2 summarizes the proposed perturbations applied to smart meter data as outlined in the above.

Algorithm 2 Adversarial Perturbation Crafting:

Input: $\hat{f}, \lambda$, C, m, n, r, target example $\vec{x}$

$\hat{y} \leftarrow \hat{f}(\vec{x})$ ▷ Save the original result
repeat n times
  $\delta_{\vec{x}} = \lambda \nabla_{\vec{x}} C_{\hat{f}}$
  $\vec{x} \leftarrow f_p(\vec{x} + \delta_{\vec{x}}, r)$
until $\text{argmax}\hat{f}(\vec{x}) \neq \text{argmax}(\hat{y}), \max \hat{f}(\vec{x}) - \max\ \hat{y} \geq m$
return $\vec{x}$

IV. Results

In this section, the performance of the proposed attack construction algorithm is evaluated via practical experimental studies conducted on Oracle models trained on various datasets and comprehensive comparisons with recent work.

A. Experimental Setup

All studies presented in this paper are implemented using Tensorflow 2.2.0 and Keras 2.3.0 and are run on the Google Colab Cloud Tensor Processing Unit infrastructure [51]. Deep learning based NILM proposed in [1] is utilized as the target application for evaluating the efficacy of the proposed algorithm which supports six active appliances: washer, dryer, dish washer, furnace, oven, and heat pump. This work focuses on altering the specific state of the furnace with the proposed attack algorithm so that the efficacy of the algorithm can be demonstrated. It is important to note that NILM is one of many ML applications in the smart grid context. The proposal in this paper is not limited to NILM applications as it is designed for any ML applications that operate on smart meter measurements.

In order to showcase the versatility of the proposed attack construction algorithm, two NILM Oracle models are considered, which are trained using two different datasets: Almanac of Minutely Power (AMP) [52] and Pecan Street (PS) [53]. The AMP dataset is composed of eleven attributes (e.g. real power, current, voltage, energy, etc.) measured for each one of the 20 common appliances present in a household located in British Columbia over a two year period at a granularity of 1-minute. The PS dataset consists of power measurements of individual appliances present in 25 households located across New York, California and Austin. These are recorded at various granularities (e.g. 1-second, 1-minute, and 15-minutes) over a 6 month period. This work utilizes the PS dataset recorded at a granularity of 1-minute to maintain consistency with the AMP dataset. The ML models constructed for the AMP and PS datasets are referred to as AMP and PS Oracles respectively. These are composed of long-short term memory units (LSTM) and 5 hidden layers. The total number of model parameters present in the AMP and PS Oracles are 77,606 and 94,577 respectively. More parameters are used for the PS Oracle to account for the greater complexity of the PS dataset.

The proposed attack construction algorithm is compared with two recent proposals in the literature. The first is the vanilla FGSM algorithm proposed in [6] that performs black-box attacks with specific examples pertaining to images and the second is the ℓ_0 FGSM algorithm proposed in [7] which is a white-box attack construction in the smart grid context. Performance evaluation is divided into two parts where the following components of the proposal are considered individually: 1) substitute model construction and 2) adversarial perturbations. The proposed substitute model construction is compared with the vanilla data augmentation algorithm proposed in [6]. This work does not compare with the proposal in [7] as it is a white-box attack construction that assumes that the full parameter set of the Oracle model is available to the attacker. Hence, there is no need to approximate the Oracle via a substitute model for this proposal. The proposed adversarial perturbation algorithm is compared with both FGSM and ℓ_0 FGSM algorithms proposed in [6] and [7] respectively.

B. Substitute Model Construction

In this section, the performance of the proposed substitute model construction algorithm is evaluated. As discussed in Section III-B1, the substitute model is a FFNN. Substitute models that approximate the AMP and PS Oracle models are composed of 4 hidden layers with 18,426 parameters and 6 hidden layers with 84,746 parameters respectively. Both discrete and probabilistic outputs are considered for the AMP substitute model whereas for the PS substitute model only discrete outputs are considered.

(c) 2021 Crown Copyright. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2021.3062722

As such, the first step in the construction of the substitute model is the augmentation of the initial training set which is composed of five training points. For illustrative purposes, Table I lists the initial set of five data points utilized for the appliance class *Furnace*. Columns $T1$ to $T6$ represent six consecutive 10-minute intervals and the numerical values listed in these columns indicate aggregate power consumption in kilowatts (KW) for a household. The last two columns indicate the outputs from the AMP and PS Oracles (i.e. whether the furnace was active or inactive over the intervals $T1$ to $T6$).

TABLE I: Initial dataset for substitute construction (KW).

Sample	T1	T2	T3	T4	T5	T6	AMP	PS
1	0	0	0	0	0	0	inactive	inactive
2	0.5	0.5	0.5	0.5	0.5	0.5	inactive	active
3	4.5	3.5	4.5	5.5	4.5	5.5	active	active
4	10	0.5	0	1	1.5	3	active	inactive
5	2	10	8	1	1.5	7	active	active

Next, the performance of the vanilla data augmentation algorithm presented in [6] is considered. As discussed in Section III-B2, this algorithm depends on the parameter λ. The performance of this algorithm is firstly investigated by assessing the *accuracy* of the AMP substitute model constructed using the training set generated by this algorithm for various values of λ. The accuracy metric captures the percentage of outputs from the substitute model that matches the outputs from the Oracle model. The test data is composed of 10,000 points where 50% of the data points belong to the active state for each appliance class and the remainder belong to the inactive state. This allows for a balanced representation of the two states an appliance can take. The results obtained from the vanilla augmentation algorithm are plotted in Fig. 6. It is evident that the highest accuracy of 78% is achieved for $\lambda = 0.2$.

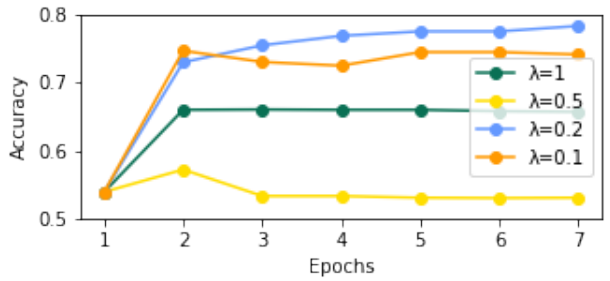


Fig. 6: Impact of λ on substitute model accuracy.

The accuracies of the AMP and PS substitute models trained using the substitute model training algorithm proposed in this paper are listed in Table II. The first row in this table is directly comparable to the vanilla augmentation algorithm as the same AMP Oracle model is approximated by both algorithms. The AMP substitute model has an accuracy of 92.99% which is much higher than the best performance of 78% from the vanilla algorithm. Another accuracy is also included when the output of the AMP Oracle is probabilistic which is comparable to the discrete case. However, more queries to the Oracle were necessary for the probabilistic case when compared to the discrete case. This is expected as the probabilistic outputs capture more information and the training process of the substitute model will be more nuanced requiring more training data points. The last row in the table consists of the results for the PS substitute model for the probabilistic case. The accuracy is lower, however, this is expected as the PS dataset is much more complex (i.e. more households) in comparison to the AMP dataset (i.e. one household). This is still higher than the substitute model resulting from the vanilla algorithm for the simpler AMP dataset. Also, another interesting observation is that the number of epochs needed to train the substitute models for reaching the accuracies listed in the table increases as the complexities of the associated datasets and outputs increase.

TABLE II: Substitute model accuracy with proposal.

Dataset	**Condition**	**Epochs**	**Queries**	**Accuracy**
AMP	Discrete	7	1,210	92.99 %
AMP	Probabilistic	8	2,174	90.79 %
PS	Probabilistic	11	2,816	83.88 %

C. Adversarial Perturbations Crafting

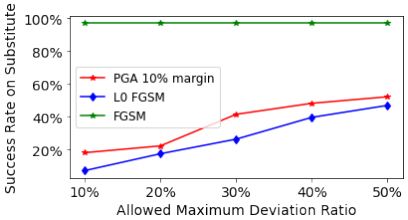
Next, PGA which is the proposed adversarial perturbations crafting algorithm is evaluated in this section. As imposing a limit on the magnitude of the perturbations applied to increase the stealthiness of the proposed attack, these attack perturbations may not successfully fool the substitute or Oracle models. For this reason, two metrics are introduced: *success*, and *transfer* rates. Success rate R_s refers to the percentage of adversarial examples that results in fooling the substitute model while satisfying all underlying constraints (e.g. maximum deviation ratio r and confidence interval m). Transfer rate R_t refers to the percentage of adversarial examples that fool the Oracle ML model. The test set is composed of 2,000 labeled data points with a balanced representation of the active and inactive states.

Comparisons are made with the FGSM and ℓ_0-FGSM algorithms. With the FGSM algorithm, the perturbation amplitude ϵ is iteratively increased until the resulting perturbations fool the substitute model or the maximum number of iterations is reached. PGA and ℓ_0-FGSM algorithms search adversarial perturbations until the maximum allowable deviation ratio r is reached in addition to the two other criteria used for FGSM. The ℓ_0 norm constraint used in the ℓ_0-FGSM algorithm is set to be 30% of the total number of input dimensions which is same as the setup presented in the original paper.

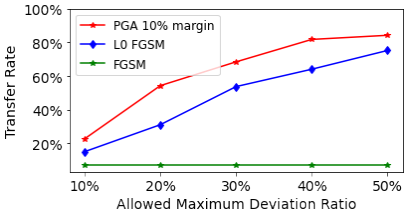
It is important to note that FGSM is a black-box attack algorithm with no constraints imposed on the magnitude of perturbations that can be applied to the input data. This implies that these perturbations are not stealthy and can be easily detected by error checking mechanisms. On the other hand, the ℓ_0-FGSM algorithm is a white-box attack algorithm with constraints imposed on the perturbations crafted. The internal parameters of the model under attack are available to the attacker and thus the crafted adversarial perturbations will be tailored to the attacked Oracle model. With the proposal, the attacker has access to limited knowledge regarding the model

(c) 2021 Crown Copyright. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2021.3062722

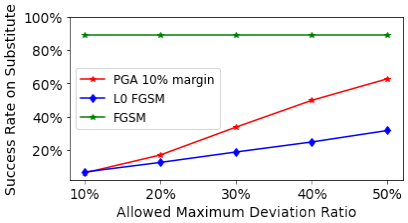


(a) Comparison of R_s with different r.

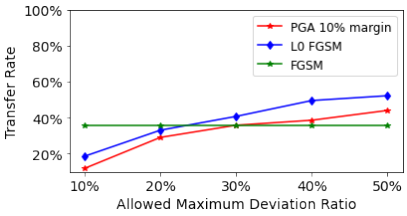


(b) Comparison of R_t with different r.

Fig. 7: Training based on the AMP dataset.



(a) Comparison of R_s with Different r on PS Dataset



(b) Comparison of R_t with different r.

Fig. 8: Training based on the PS dataset.

under attack and constraints on perturbations are enforced for attack stealthiness.

As such, results from these comparative studies are presented in Fig. 7 and Fig. 8 for the AMP and PS datasets. For all cases, the maximum deviation ratio (e.g. perturbation limit) r is modified and the resulting values of R_s and R_t are recorded. As no constraints on perturbations are imposed with the FGSM algorithm, the outcomes for this algorithm do not change with various values of r. This algorithm exhibits superior performance with the success rates as unlimited perturbations can be applied until the substitute model is fooled. However, the resulting lack of stealthiness will be problematic. It is also interesting to observe that the transfer rate of the FGSM algorithm are very low in comparison to the other two algorithms compared.

With the ℓ_0-FGSM, the attacks are designed using the internal knowledge of the trained parameters of the substitute and Oracle models. For the AMP dataset, this algorithm results in slightly lower success, transfer rate in comparison to the PGA algorithm (i.e. Fig. 7a-7b). For the PS dataset, this algorithm performs slightly better than the PGA algorithm for transfer rate (i.e. Fig. 8b). This is not the case for the success rate (i.e. Fig. 8a). Overall, even though the proposed attack construction algorithm is privy to much less information in comparison to the ℓ_0-FGSM, it results in comparable and mostly better performance than the ℓ_0-FGSM algorithm.

Next, the impact of the confidence margin m on the success and transfer rates of the PGA algorithm are analyzed for fixed $r = 0.3$. m is utilized to increase the likelihood of the success of the adversarial example crafted using the substitute model on the Oracle model. These results are tabulated in Table III. It is clear that as m increases, the transfer rate increases and this is as expected. However, the success rates are declining. This is mainly due to the increasingly stringent constraints imposed by larger values of m that must be satisfied by the crafted adversarial perturbations.

TABLE III: Performance with different confidence margins.

Confidence Margin (%)	Success Rate (%)	Transfer Rate (%)
m = 10	38.46	61.43
m = 20	35.71	64.61
m = 30	30.77	71.43
m = 40	28.02	72.55
m = 50	21.42	74.36

The impact of the distribution of the active and inactive states on the success rates of PGA is also presented in Fig. 9. The confidence margin m is fixed (10%) for results presented in this figure. Adversarial perturbations that result in transitions from active to inactive states result in high success rates which increase with r and reach values close to 100%. On the other hand, transitioning from inactive to active states indicate increasing rates of success with increasing values of r. However, these success rates are not as high as the transition from active to inactive states. This can be attributed to the distribution of the active and inactive states. The active states are more concentrated whereas the inactive states are more dispersed. This implies that greater perturbations are necessary for successful transition from the inactive to active states than the opposite case.

D. Stealthiness of Attack Construction

Next, the stealthiness of the proposed attack construction, which pertains to the ability to bypass visual or existing checking mechanisms (e.g. abnormal power readings) is presented. Fig. 10 illustrates the smart meter reading for a household over a 24 hour period where perturbation (crafted using $r = 20\%$ and $m = 10\%$) is applied to a single one hour window highlighted by the orange curve. It is clear that the perturbed data is not distinguishable from actual smart meter readings

(c) 2021 Crown Copyright. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2021.3062722

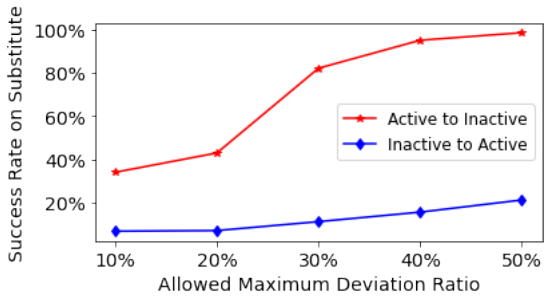


Fig. 9: Success rates for active and inactive states.

and results in successfully directing the targeted ML model to produce incorrect outputs over the attacked interval. These erroneous outputs by the Oracle will lead to serious consequences that include: over-billing (e.g. consumer extortion) or under-billing (e.g energy theft) for specific use of particular appliances; incorrect computation of direct load control, and so on.

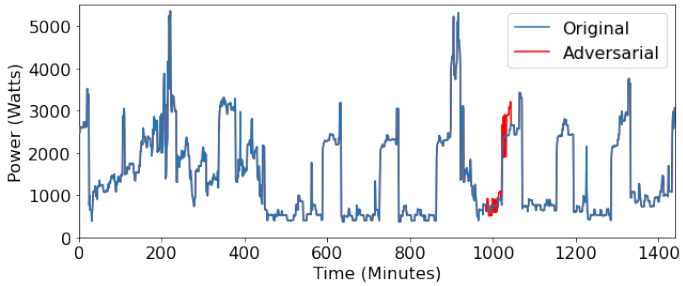


Fig. 10: Stealthiness of 24 hour smart meter readings.

E. Impact of the Proposed Attack

First, the impact of incorrectly classifying a flexible appliance in a real-time HEMS system proposed in [54] is studied. For illustration purpose, the targeted appliance in this case study is a furnace. The scheduling algorithm is based on the Markov Decision Process (MDP) which selects the best policy to minimize total power consumption and costs over a time interval. However, if adversaries manipulate the furnace's state and return 'off' state to the HEMS when it is actually 'on', the MDP will start at the wrong system state and lose control of the furnace. In this case, the furnace will remain in operation even during the peak periods. Assuming the targeted furnace consumes 600 watts, applying the time-of-use prices for Ontario in 2019 as listed in Table IV will result in the contribution of the furnace to the monthly bill to be $99.72.

TABLE IV: Time-Of-Use weekday price in Ontario, Canada

TOU Price Period	2019 TOU Price
Off-peak (7 p.m. - 7 a.m.)	10.1¢/kWh
Mid-peak (11 a.m. - 5 p.m.)	14.4¢/kWh
On-peak (7 a.m. - 11 a.m. and 5 p.m. - 7 p.m.)	20.8¢/kWh

Next, the impact of the proposed attack on demand response programs is studied. Load flexibility is first evaluated by EPUs to evaluate the potential for peak shaving in the system. This involves the disaggregation of smart meter data using NILM [4]. The proposed attack can be used to lead to the incorrect estimation of load flexibility in the system. If a flexible appliance is classified as 'off' although it is actually 'on', it will not be actuated by the EPU. This will affect the ability of the EPU to shave demand peaks when a large number of flexible loads are attacked. For example, if it is estimated that 100 kW of the demand during peak hours in a day stem from furnaces and the NILM misclassifies the state of these appliances, then this peak reduction potential is lost. When the system is overloaded, this loss in flexibility can lead to cascading outages and failures.

V. Conclusions

This paper presents a novel black-box attack construction algorithm targeting ML models operating on smart meter data. The efficacy of the proposed stealthy adversarial perturbation algorithm on a deep-learning based ML model that performs appliance disaggregation on smart meter readings has been successfully demonstrated. The proposal in this paper outperforms state-of-the-art black-box attack paradigms proposed in the literature. This work sheds light onto new attack modes that are introduced due to inherent vulnerabilities in ML models and these attacks can fool sophisticated ML models with minimal information at hand. As the modern smart grid is moving towards increased automation with the integration of ML constructs, this poses a real threat to the reliable operations of the power grid. As future work, we intend to investigate how ML models deployed in the smart grid settings can be designed to be inherently robust to adversarial attacks such as that presented in this paper. In a more broader context, we also aim to design more efficient ML models for performing tasks in the smart grid context such as appliance disaggregation.

References

[1] J.Wang, et al."Ensemble-based Deep Learning Model for Non-intrusive Load Monitoring". Proceedings of the IEEE Canada Electrical Power and Energy Conference (EPEC 2019), Montreal, Quebec, Canada, Oct. 2019.
[2] H.Yue, et al. "Estimating Demand Response Flexibility of Smart Home Appliances via NILM Algorithm." 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). Vol. 1. IEEE, 2020.
[3] J.Ponoćko and JV.Milanović. "Forecasting demand flexibility of aggregated residential load using smart meter data." IEEE Transactions on Power Systems 33.5 (2018): 5446-5455.
[4] A.Lucas, et al. "Load Flexibility Forecast for DR Using Non-Intrusive Load Monitoring in the Residential Sector." Energies 12.14 (2019): 2725.
[5] Y. Chen, Y. Tan and D. Deka, "Is Machine Learning in Power Systems Vulnerable?," 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Aalborg, 2018, pp. 1-6.
[6] N.Papernot, et al."Practical black-box attacks against machine learning." Proceedings of the 2017 ACM on Asia conference on computer and communications security. 2017.
[7] X.Zhou, et al. "Evaluating Resilience of Grid Load Predictions under Stealthy Adversarial Attacks." 2019 Resilience Week (RWS). Vol. 1. IEEE, 2019.
[8] C.Szegedy, et al. "Intriguing Properties of Neural Networks." Dec. 2013.
[9] S.Moosavi-Dezfooli, et al. "DeepFool: a Simple and Accurate Method to Fool Deep Neural Networks." Nov. 2015.

(c) 2021 Crown Copyright. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2021.3062722

[10] N.Papernot, et al. "The Limitations of Deep Learning in Adversarial Settings." Nov. 2015.
[11] S.Moosavi-Dezfooli, et al. "Universal Adversarial Perturbations." Oct. 2016.
[12] S.Ali, et al. "Evasion Attacks with Adversarial Deep Learning Against Power System State Estimation."
[13] Y.Chen, et al. "Exploiting vulnerabilities of load forecasting through adversarial attacks." Proceedings of the Tenth ACM International Conference on Future Energy Systems. 2019.
[14] I.Niazazari and H.Livani. "Attack on Grid Event Cause Analysis: An Adversarial Machine Learning Approach." 2019.
[15] P.Foreman. "Vulnerability management." Auerbach Publications, 2009.
[16] L.Tian and T.Shu. "Adversarial False Data Injection Attack Against Nonlinear AC State Estimation with ANN in Smart Grid." International Conference on Security and Privacy in Communication Systems. Springer, Cham, 2019.
[17] L.Tian, et al. "False data injection attack in smart grid topology control: Vulnerability and countermeasure." 2017 IEEE Power & Energy Society General Meeting. IEEE, 2017.
[18] Y.Guo, et al. "Modeling distributed denial of service attack in advanced metering infrastructure." 2015 IEEE power & energy society innovative smart grid technologies conference (ISGT). IEEE, 2015.
[19] Z.Yu and W.Chin. "Blind false data injection attack using PCA approximation method in smart grid.", IEEE Transactions on Smart Grid 6.3 (2015): 1219-1226.
[20] M.Esmalifalak, et al. "Stealth false data injection using independent component analysis in smart grid." 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm). 2011.
[21] A.Ruano, et al. "NILM techniques for intelligent home energy management and ambient assisted living: A review." Energies 12.11, 2019: 2203.
[22] J.R.Herrero, et al. "Non intrusive load monitoring (nilm): A state of the art." International Conference on Practical Applications of Agents and Multi-Agent Systems. Springer, Cham, 2017.
[23] S.Desai, et al. "A survey of privacy preserving schemes in IoE enabled Smart Grid Advanced Metering Infrastructure." Cluster Computing 22.1, 2019: 43-69.
[24] "Google AI," Google. [Online]. Available: https://ai.google/. [Accessed: 12-Dec-2020].
[25] X.Liu and P.S.Nielsen. "Scalable prediction-based online anomaly detection for smart meter data." Information Systems 77, 2019: 34-47.
[26] X.Liu and P.S.Nielsen. "Regression-based online anomaly detection for smart grid data".2016.
[27] K.Zheng, et al. "Electricity theft detecting based on density-clustering method." 2017 IEEE Innovative Smart Grid Technologies-Asia (ISGT-Asia). IEEE, 2017.
[28] L.Tian and M.Xiang. "Abnormal power consumption analysis based on density-based spatial clustering of applications with noise in power systems." Automation of Electric Power Systems 5 (2017): 64-70.
[29] Q.Zhang, et al. "Electricity Theft Detection Using Generative Models." 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, 2018.
[30] D.Li, et al. "Anomaly detection with generative adversarial networks for multivariate time series." 2018.
[31] N.O'Mahony. "Anomaly detection with generative adversarial networks for multivariate time series", DELL EMC Research Europe. [Online]. Available: https://smartgrid-cybersecurity.events/wp-content/uploads/2017/04/Security-Analytics-for-Smart-Grid-Anomaly.pdf. [Accessed: 12-Dec-2020].
[32] Y.Hu. "Autoencoder Forest for Anomaly Detection from IoT Time Series", SP Group. [Online]. Available: https://www.datacouncil.ai/talks/time-based-autoencoder-ensemble-for-anomaly-detection-from-iot-time-se [Accessed: 12-Dec-2020]
[33] M. Chester, "Engaging Residential Customers with Demand Response on 15 Minutes' Notice," Energy Central. [Online]. Available: https://energycentral.com/c/em/engaging-residential-customers-demand-response-15-minutes%E2%80%99-notice-exclusive. [Accessed: 23-Jun-2020].
[34] H.Zhang, et al. "The limitations of adversarial training and the blind-spot attack." 2019.
[35] E.Hossain, et al. "Application of Big Data and Machine Learning in Smart Grid, and Associated Security Concerns: A Review." IEEE Access, vol. 7, no. 99, IEEE, 2019, pp. 13960–88.
[36] "Google AI Platform Prediction" Google. [Online]. Available: https://cloud.google.com/ai-platform/prediction/docs/reference/rest/v1/HttpBody. [Accessed: 12-Dec-2020].
[37] Y.Sun, et al. "Smart Meter Privacy: Exploiting the Potential of Household Energy Storage Units." IEEE Internet of Things Journal, vol. 5, no. 1, IEEE, Feb. 2018, pp. 69–78.
[38] G.Eibl and D.Engel. "Differential Privacy for Real Smart Metering Data." Computer Science - Research and Development, vol. 32, no. 1, Springer Berlin Heidelberg, Mar. 2017, pp. 173–82.
[39] Z.Lan, et al. "A non-intrusive load identification method based on convolution neural network." 2017 IEEE Conference on Energy Internet and Energy System Integration (EI2). IEEE, 2017.
[40] L. Mauch and B. Yang, "A new approach for supervised power disaggregation by using a deep recurrent LSTM network", Proc. of the 3rd IEEE GlobalSIP, 2015, pp. 63-67.
[41] PBM.Martins, et al. "Application of a deep learning generative model to load disaggregation for industrial machinery power consumption monitoring." 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). IEEE, 2018.
[42] P.Dash, and K.Naik. "A Very Deep One Dimensional Convolutional Neural Network (VDOCNN) for Appliance Power Signature Classification." 2018 IEEE Electrical Power and Energy Conference (EPEC). IEEE, 2018.
[43] J.Cho, Z.Hu, and M.Sartipi. "Non-Intrusive A/C Load Disaggregation Using Deep Learning." 2018 IEEE/PES Transmission and Distribution Conference and Exposition (T&D). IEEE, 2018.
[44] P.Held, et al. "Frequency invariant transformation of periodic signals (FIT-PS) for signal representation in NILM." IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2016.
[45] R.Bonfigli, et al. "Denoising autoencoders for non-intrusive load monitoring: improvements and comparative evaluation." Energy and Buildings 158, 2018: 1461-1474.
[46] BC.Csáji. "Approximation with artificial neural networks." Faculty of Sciences, Etvs Lornd University, Hungary 24.48, 2001: 7.
[47] A.Demontis, et al. "Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks." In 28th USENIX Security Symposium (USENIX Security 19). USENIX Association, Santa Clara, CA, 2019.
[48] G.Hinton, et al. "Distilling the Knowledge in a Neural Network." 2015.
[49] N.Papernot,et al. "cleverhans v2. 0.0: an adversarial machine learning library, "https://github.com/tensorflow/cleverhans/. 2016
[50] A.Chakraborty, et al. "Adversarial attacks and defences: A survey." 2018.
[51] "Google Colaboratory," Google. [Online]. Available: https://colab.research.google.com/notebooks/intro.ipynb. [Accessed: 13-Jun-2020].
[52] S.Makonin, et al. "AMPds: A public dataset for load disaggregation and eco-feedback research." 2013 IEEE Electrical Power & Energy Conference. IEEE, 2013.
[53] Pecan Street Dataset, "https://dataport.pecanstreet.org/"
[54] C.Vivekananthan, et al. "Real-time price based home energy management scheduler." IEEE Transactions on Power Systems 30.4 (2014): 2149-2159.

Junfei Wang Junfei Wang (Student Member, IEEE) received the B.Sc. degree in Communication Engineering from Beijing Institute of Petrochemical Technology, Beijing, China in 2008, and M.Eng degree from Beijing University of Posts and Telecommunications, Beijing, China in 2010, and M.E.Sc degree from Western University, London, ON, Canada in 2020. He is now a Ph.D. student at York University, Toronto, ON, Canada. He has practical experience on data-driven application analysis, management and innovation. His main research interests include trustworthy machine learning and representation learning in smart grid.

(c) 2021 Crown Copyright. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2021.3062722

Pirathayini Srikantha is a Canada Research Chair (Tier 2) and an Assistant Professor in the Department of Electrical Engineering and Computer Science at York University. She received her B.A.Sc. degree in Systems Design Engineering from the University of Waterloo in 2009 and her M.A.Sc. degree in Electrical and Computer Engineering from the same institute in 2013. She obtained her Ph.D. degree from The Edward S. Rogers Sr. Department of Electrical and Computer Engineering at the University of Toronto in 2017. She is currently serving as an Associate Editor for the IEEE Transactions on Smart Grid journal. She is a certified Professional Engineer (P.Eng.) in Ontario. Her main research interests are in the areas of large-scale optimization and distributed control for enabling adaptive, sustainable and resilient power grid operations. Her work has been published in premier smart grid journal and conference venues. Her research efforts have received recognitions that include the best paper award (IEEE Smart Grid Communications) and runner-up best poster award (ACM Women in Computing).

(c) 2021 Crown Copyright. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.