A Generative Model for Evasion Attacks in Smart Grid

Venkata Praveen Kumar Madhavarapu[†], Shameek Bhattacharjee[‡], and Sajal K. Das[†]

[†] Department of Computer Science, Missouri Univ. of Science and Technology, Rolla, USA

[‡] Department of Computer Science, Western Michigan University, Kalamazoo, USA

E-mail: vmcx3@mst.edu, shameek.bhattacharjee@wmich.edu, sdas@mst.edu

***Abstract*—Adversarial machine learning (AML) studies how to fool a machine learning (ML) model with malicious inputs degrade the ML method's performance. Within AML, evasion attacks are an attack category that involves manipulation of input data during the testing phase to induce a misclassification of the data input by the ML model. Such manipulated data inputs that are called, *adversarial examples*. In this paper, we propose a generative approach for crafting evasion attacks against three ML learning based security classifiers. The proof of concept application for the ML based security classifier is the classification of compromised smart meters launching false data injection. Our proposed solution is validated with a real smart metering dataset. We found degradation in compromised meter detection performance under our proposed generative evasion attack.**

***Index Terms*—Adversarial Machine Learning, Smart Grid, AMI**

I. Introduction

An Advanced Metering Infrastructure (AMI) is a sub layer of the smart electrical grid composed of smart meters of customers, electric utility's data management systems and communications networks that connect the two. The AMI allows utilities to collect situational data on loads and power consumption from smart meters installed on the customer site. (see Fig.1). Such data play an important role in tasks such as automated billing, demand response, load forecast and management [2] highlighting the importance of smart meter data integrity.

The possibility for falsification of smart meter data is greatly increased due to the physical and cyber accessibility of smart meters to the end users and malicious actors. Commonly reported data falsification via physical exploits include transduction attacks [5] that report lower than actual usage for lesser bills (such an attack is a Deductive data falsification type). Because of the cyber and interconnected nature of AMI, it is possible for organized adversaries like cyber criminals, utility insiders or business competitors [6] to compromise multiple smart meters' or capture their data and report false readings [4] via a cyber intrusion. An elaborate threat model laying out various data falsification attack types had been laid out in an earlier work [7], which are elaborated later in the threat model.

There exists various machine learning inspired solutions like [3], [7]–[9] to detect organized data falsification attacks from smart meters. The usage patterns can be studied and modeled into an anomaly detection at the device level. However, machine learning methods themselves have been shown to have several algorithmic vulnerabilities [11]. The field of Adversarial Machine Learning (AML) deals with the study of fooling machine learning models through malicious inputs, commonly

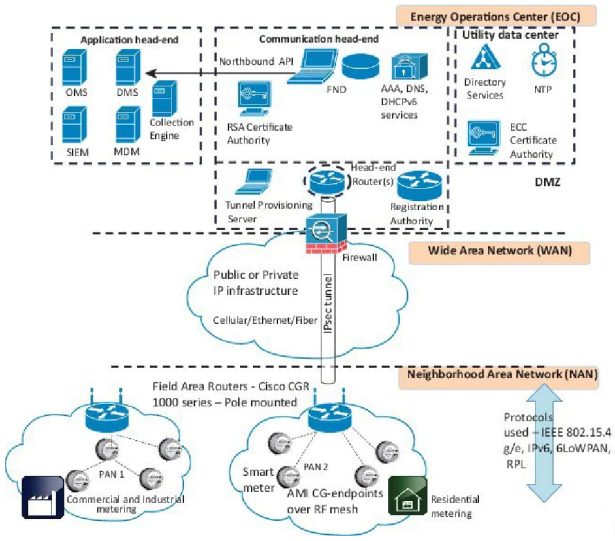


Fig. 1. An Architecture of Advanced Metering Infrastructure (AMI) [10]

known as "Adversarial Examples". The term 'fooling' refers to induce machine learning models to take an undesirable or unintended outcome, when specially crafted adversarial examples are fed as inputs. For instance, by introducing specially crafted words at strategic places, adversaries can make the spam emails bypass a trained spam filter. Here the specially crafted spam email is called the adversarial example and the strategy used to evade the spam filter is the evasion strategy that exploits weaknesses in the classification model for spam filters.

Depending on the degree of knowledge an adversary possesses about the problem domain and the ML method, adversarial attacks can be classified as black-box and white-box. In black-box approach, the adversary has no knowledge about the training data and the machine learning model but has access to the output of the machine learning model for every input given, which are used to design attack strategies. In contrast, in the white-box approach, the adversary has access to the training data and ML model, whose knowledge is used to craft adversarial examples.

While there are various categories of adversarial examples in AML, they can be broadly classified into Evasion attacks and Poisoning attacks. Poisoning attacks occur during the training phase whereas Evasion attacks happen during the testing phase. Our focus of this paper is limited to the study of evasion attacks, where the objective of an adversary is typically to avoid the identification of the true class to which an input belongs. This renders the learning method useless by compelling it to perform randomly with inaccurate results. As an example, if the goal

of a ML method is to detect an attack, and an adversarial example for evasion would be crafting an attack strategy, such that when the ML model confronts this example, the attack input is inferred as benign with a high probability. Fig. 2, depicts a schematic of a the machine learning model with good performance but result in undesired outcomes when the test data is intelligently modified using adversarial examples.

The usage of adversarial examples have been shown widely against deep learning models, support vector machine (SVM) and linear regression in the context of image processing and recognition [11]. For example these techniques were shown to successfully alter image recognition to result in a completely different classification output by just modifying few pixel's intensities that is imperceptible to the human eye. However, research on adversarial examples in the context of other ML models in the context of smart grid is still in its nascent stages. The works in [14] employ AML for smart grid in the context of a modbus injection in energy trading. To the best of our knowledge, there exists no work to handle evasion attacks against attack detectors in AMI, which motivates our work.

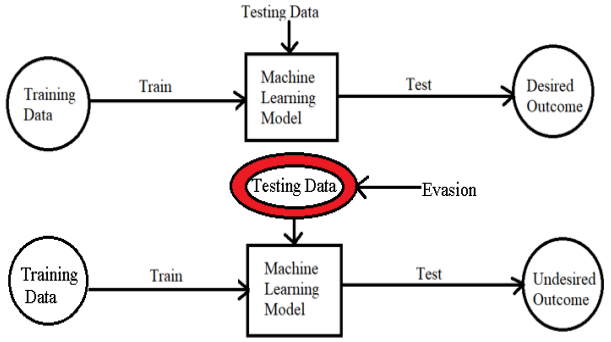


Fig. 2. Evasion attack on machine learning model

Contributions of this work: In this paper, we propose a generative approach for crafting an evasion attack strategy that bypasses compromised smart meter detection classifiers in smart metering, which involve binning the data distribution in discrete partitions or bins. Specifically, using the knowledge of training data and design model, we formulate an optimization problem that aims to find out how to control the falsified data's membership across different discrete bins without sacrificing the adversary's intended operational impact that depends on the targeted margin of false data. The evasion strategy is executed in such a manner that the data membership of the perturbed data samples belong to those bins that are more frequently observed in the benign training instance while preserving the constraint on attack type and data falsification margin (both of which are essential for creating an operational impact on AMI while bypassing the learning method).

The evasion strategy can be applied regardless of target attack type and the target margin of false data. Furthermore, it is experimentally shown that the evasion strategy has transfer-ability property. To prove the transfer-ability of our adversarial example, which is coveted property of adversarial evasion, we also proved that our method of adversarial example, not only degrades performance of the folded Gaussian classifier but also other known clustering and classification methods such as KL distance based Trust [8] and DBSCAN [1]. All of the above are clustering based classification approaches.

The remainder of the paper is organized as follows. Section II describes the background and preliminaries. Section III presents a generative approach to generation of adversarial evasion attack. Section IV shows the effectiveness of the proposed attack strategy compared to existing attacks.

II. Background and Preliminaries

This section discusses the AMI system architecture, data falsification threat model, main ML based security model [7] for which the adversarial examples are being generated.

A. System Model

Consider a set of N smart meters reporting power consumption data to a utility periodically. The P_t^i represents the reading of i-th smart meter at the end of time slot t (hourly in our data set). A Neighborhood Area Network (NAN) which is formed by a collection of houses is governed by a NAN gateway, that acts as a data concentrator to collect data from multiple smart meters in an area. Set of NAN gateways are connected to form a Field Area Network (FAN), governed by a FAN gateway, which in turn connects to the Utility Wide Area Network (WAN) where the data is stored for analysis and decision making.

Description of Datasets: We have used real AMI dataset to validate the proposed solution. The dataset is Pecan Street Project [15] containing hourly power consumption data from 220 houses from a Solar village in Texas, USA, collected between 2014-2016. The Texas dataset exhibits much more randomness and shifting trends in power consumption due to more penetration of renewable energy sources.

B. Threat Model

Due to the presence of NaN gateways, an attacker can intrude these devices to intercept data from a subset of smart meters and intelligently craft adversarial examples (or intercept using a botnet and doing a man-in-the-middle attack). This is realistic because there is time delay between the generation of data and its actual usage. Effectively, [9], has elaborated that data falsification threat model that can be abstracted into four aspects; attack type, strength, scale, and strategies.

Attack scale quantifies the certain fraction $\frac{M}{N} = \rho_{mal} \in [0, 1)$ of the N smart meters compromised by an organized adversary where M is the no. of compromised meters based on their attack budget. For example, $\rho_{mal} = 0.50$, means 50% of the total number of meters are compromised. If N is small, the ρ_{mal} may be high even with a small attack budget.

The attack type dictates the way data falsification is done and depends on the operational impact intent of the data falsification. The [7], [9] shows additive, deductive, alternating switching as the possible attack types from a single meter's perspective and also give motivations behind each attack type. In this paper, we show evasion using only additive and deductive attack types. For example, for deductive attack, the actual power consumption data P_t^i from the i-th compromised meter at time t is modified as $P_t^i - \delta_t$. For additive, it would be $P_t^i + \delta_t$. Falsification of data is achieved by changing the actual power consumption value P_t^i by some amount δ_t (which is strategic value determined by attack strength). We denote

δ_{avg} as the average margin of false data for each compromised meter which is the *attack strength*. The mean of all δ_t values for a compromised smart meter equals δ_{avg}, and δ_{avg} can vary according to the stealth level and intended impact of an adversary and hence it is kept as a variable.

Evasion attacks may be targeted or untargeted based on the goal of the adversary. Targeted attacks can result in classification of data from one class to another specific class, while non-targeted attacks' objective is to mis-classify to any class other than the correct one. In white-box approach, the adversary will have complete knowledge of the weights and all data on which the machine learning was trained, which allow targeted attacks to be launched. In black-box and gray-box attacks, the knowledge will be close to none and partial, respectively and hence they often lead untargeted attacks. Our work in this study is focused on the targeted white-box attacks with reasonable realistic restrictions of what the adversary may not know.

C. Target ML based Security Model

The main security model being evaded is the folded Gaussian trust scoring model proposed in [7]. Later we show that the same adversarial example crafted keeping the [7] as a reference also works against DB-SCAN and KL distance trust.

The Folded Gaussian Model has two modules: First module, is an anomaly based attack detection technique that detects the presence and the attack type and then performs a mean and standard deviation correction according to the attack, at the micro-grid level. The second module is a folded gaussian based trust scoring classifier, at the device level that calculates the trust of every smart meter by comparing the densities of smart meter readings relative the corrected mean and the standard deviation and the attack status indicated by the first module.

The trust scores are calculated over a time period T which we call a frame. The smart meters usually measure a reading every hour. Therefore, number of timeslots in a day is 24. Thus, T is 24 times the number of days in a frame. We taken a 30 days in our study therefore, $T = 24 \times 30$ in our paper. The trust scoring model assigns a numeric label to each reading p_t^i, based on its proximity to the instantaneous corrected mean μ_{MR} measure compared to the sample σ_{MR} (the corrected standard deviation of all p_t^i from the calculated μ_{MR} in the time slot t). The absolute difference between the p_t^i for any meter i and the μ_{MR} is denoted by $\theta_{diff}^i = |p_t^i - \mu_{MR}|$. Given this, each p_t^i is assigned a rating denoted by l according to rule given by Table I, that uses the (68%–95%–99.7%) rule for Gaussian distributions to assign p_t^i as belonging to one of the 4 possible bins. The maximum rating 4 is the one belongs to readings within the first standard deviation $\sigma_{MR} = \Delta_{abs}$ from the μ_{MR}. Similarly, ratings 3,2,1 are obtained if the meter's data falls within the 2nd, 3rd standard deviations and beyond. Over a time frame T, the rating labels recorded on each time slot t for meter i is accumulated and sorted to form a rating vector $r_{sort}^i = r_0 \leq r_1 \leq ... \leq r_{T-1}$

First, a weight parameter x_t between 1 to 4 is calculated via Eqn. 1 where $K = 4$ is the total number of discrete rating levels, where the number of readings in the selected time frame totals to T. The final weights are achieved through Eqn. 2 where $\mu_{BR} = 4$ is the best or highest possible rating level and

TABLE I
DISCRETE RATING LEVELS

Scenario of θ_{diff}^i	Rating (r_t^i)	No. of Readings
$\theta_{diff}^i \leq \Delta_{abs}$	4	$X = \sum I(4,t)$
$\Delta_{abs} < \theta_{diff}^i \leq 2\Delta_{abs}$	3	$Y = \sum I(3,t)$
$2\Delta_{abs} < \theta_{diff}^i \leq 3\Delta_{abs}$	2	$Z = \sum I(2,t)$
otherwise	1	$A = \sum I(1,t)$

σ_{dr}^i denotes the standard deviation of discrete ratings of each meter in the time frame T. The σ_{dr}^i for each meter will be different based on different observations compared to common mixture data, which captures individual differences in behavior. Therefore, the corresponding raw weight cw_t of the rating at time index t yielded from Eq. (2) are normalized as in Eq. (3).

$$x_t = 1 + \frac{(K-1)t}{T-1} \qquad \forall \quad t = 0, 1, ..T-1 \tag{1}$$

$$cw_t^i = \frac{1}{\sigma_{dr}^i \sqrt{2\pi}} e^{-\frac{(x_t - \mu_{BR})^2}{2(\sigma_{dr}^i)^2}} \tag{2}$$

$$w_t^i = \frac{cw_t^i}{\sum_{t=0}^{T-1} cw_t^i} \tag{3}$$

Let $I(l,t)$ be an indicator function which indicates whether a particular rating level l occurs in that time slot. All weights corresponding to each unique rating level l, such $l = 1, 2, 3, 4$ within T is added up, such that

$$I(l,t) = \begin{cases} 1, & \text{If } l \text{ occurs in timeslot } t \\ 0, & \text{Otherwise} \end{cases} \tag{4}$$

$$WD(l) = \sum_{t=1}^{T} w_t I(l,t) \tag{5}$$

$$R^i = \sum_{l=1}^{K} l \times WD(l),\ R^i \epsilon [1,4] \tag{6}$$

The aggregate weight (R^i), when interpreted as a trust score follows a folded Gaussian shape, which means R^i = 4 implies higher trustworthiness followed by a exponential 'discounting' of trust, as R^i decreases. The inverse power law inspired kernel trick is used to convert the scale of R^i between 1 and 4, into a final trust value, TR^i between 0 and 1, as shown below.

$$TR^i = \frac{1}{(K)^\eta} (R^i)^\eta \tag{7}$$

The smart meters with trust score higher than the threshold will be classified as honest. The separation between the honest and malicious is done via a classification threshold that was learnt using k-means.

D. Assumptions of Threat Model

We consider a white-box adversary model, characteristic of rival nation state actors, insider threats, or advanced persistent threats. We assume that the adversary has access to the training dataset and the folded gaussian model. This is not unreasonable since the dataset is an open public dataset from a real AMI deployment and the papers for ML security model are public. In any case, a database hack can give access to training data.

For evasion attacks, during test time, the adversary only knows the data from those meters that he has compromised and have control over and does not have knowledge of data generated by other meters that it did not compromise. This is reasonable assumption since the data generated just now is beyond anyone's control and intrusion into a set of smart meters cannot give any idea about the exact data that will be generated by a distributed set of smart meters in real time during testing.

III. Generative Evasion Strategy

This section proposes the generative strategy for creating adversarial examples. Generative models concern with how data is generated given a classification model that produces a certain output. We observed that [7] uses a discriminative approach for classification. Therefore, our effort is to generate the evasion data distributions using the generative approach that tries to escape the discriminative classifier without sacrificing the operational impact of the original attack.

Overview of Solution: From the folded Gaussian classifier, we know the score of a smart meter device will be higher if more and more smart meter readings are closer to the population mean of power consumption. With this knowledge an adversary's strategy would be to generate an attack strategy that keeps the changed readings closer to the temporal population mean while still changing the actual data. Such proximity of perturbed data towards the population mean would boost the trust score and potentially increase classification error while preserving the required δ_{avg} constraint for operational impact.

Interestingly, there is a underlying design similarity of this approach with DBSCAN and KL distance trust, although the actual mathematics of each approach is different. The similarity is in the discretization of the data into discrete levels and then using the probability density of each level for estimation of scores that combine it into a clustering approach. This is what our adversarial method seeks to harness and gives the power of transferability to DBSCAN and KL distance based trust classifier. While it may not completely evade the classifier, increase in the trust score will degrade the classification accuracy of the folded gaussian method which will have significant effect on large scale smart living IoT systems like AMI.

Problem Set Up: This step shows the evasion data generation from true data samples. The following applies to every compromised smart meter individually and hence we drop the notation i from discussions. The input $\boldsymbol{P}$ is the true electricity readings of a smart meter over T time slots in a frame. The output of generator Q will be the evasion sample of the same size which the adversary needs to generate to escape detection.

$$\boldsymbol{P} = \left[P_1, ..P_t, ..., P_T\right]_{1\times T} \quad \boldsymbol{Q} = \left[P_1^{'}, ..P_t^{'}, ..., P_T^{'}\right]_{1\times T} \quad (8)$$

The generative model's design depends on the architecture of the defense model. A close look into the folded Gaussian trust model reveals that the smart meter is classified as honest, when it has higher trust score. So, the generative model needs to create falsified data per smart meter $\boldsymbol{Q}$ such that it results in a higher trust score even in the presence of an attack of a certain type and *without reducing the strength.*

To accomplish the above, the generator has to find appropriate instantaneous δ_t (eqn.9) perturbations over a time frame that results in the highest trust score possible, while still preserving the strategic target δ_{avg} of the adversary.

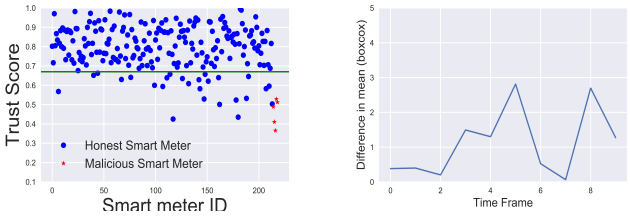
$$\boldsymbol{F} = \left[\delta_1, \delta_2, ..., \delta_t, ..., \delta_T\right]_{1\times T} \quad (9)$$

$$P_t^{'} = P_t \pm \delta_t \qquad \delta_{avg} = \frac{\sum_{t=1}^{T} \delta_t}{T} \quad (10)$$

The working logic of the scoring model indicates that when a data point is within first standard deviation (rating level 4), it contributes to a higher trust. This is because the weight of such observation is proportional to the probability density of observing level 4, the highest in the benign dataset. The rating levels 3,2,1 indicate increasing distance of the data points from the instantaneous sample mean and contribute less to the trust score due to the same density proportionality feature. Therefore, the probability densities in the benign dataset for levels 3,2,1 are exponentially lower, and more number of readings get a lower score in the case of a simple random attack strategy.

This gives an intuition that, if the perturbed data points stay closer to the mean even after false data injection with certain type and the δ_{avg}, it should lead to a higher trust score. To do this, the adversary needs to find the best values for X, Y, Z and A from Table I which will be the number of readings in each discrete rating level in the targeted time frame. The discrete rating levels depend on the sample mean and standard deviation. Ideally, this requires the adversary to estimate or know the values of mean, standard deviation after the attack. Additionally, the threshold for classification that classifies malicious from the honest ones need to be known. Therefore, in the next steps we introduce strategies that an adversary could employ to estimate the mean, standard deviation after a potential attack of a certain strength, and the classification threshold.

Estimating a Safe Threshold (TH): In the [7], the threshold was generated via k-means. The threshold depends on the final distribution of the trust scores and attack incidence flag generated by the anomaly detector. If the attack presence is inferred, the second module knows to perform a k-means classification with $k = 2$ over the resultant trust scores. The meters with scores below the threshold are classified as malicious. To evade, adversary needs to ensure that trust scores after data falsification is just above this classification threshold. Since, the adversary does not know the threshold, the threshold needs to be estimated by the adversary to escape the detection.



(a) (b)

Fig. 3. (a) Safe Threshold (TH) (b) Difference in mean

The adversary can use the training data and the model knowledge to find the trust scores of the set of smart meters based on their readings. Then the adversary will simulate attacks from a set of smart meters of cardinality M^{evade} and calculate the resulting trust scores. Given that the anomaly detection model [3], [7], [9], will detect presence of attacks, a k-means on all the resultant trust scores will result in an estimated classification threshold TH. This gives the adversary a baseline idea that the minimum resulting trust scores after the evasion attacks need to be greater than TH to escape detection.

We used a low value of $M^{evade} = 5$ as shown in Fig. 3(a), since we expect the adversary to be economic in its budget. The threshold with a low M^{evade} will be highest, due to less

number of malicious meters. Therefore, it gives a pessimistic case estimate of the threshold, above which the scores after actual falsification needs to be kept. As $M^{evade} = 5$ increases, it creates more lower trust scores in the final input to the k-means classifier, and therefore the learnt threshold starts to decrease.

Estimating Mean: The exact trust score for Gaussian trust model needs the knowledge of the mean of current data. This is unknown to the adversary as the mean value is based on current time slot and over all smart meters. Let μ_t denote the arithmetic mean at time t after attack. Since μ_t is unknown to the adversary, it needs to be estimated using the knowledge of the data before the attack. The difference in mean over two consecutive time frames is very low and can be seen in boxcox scale from Fig. 3(b). The mean is piecewise stationary although the time series over a longer time horizon changes rapidly. The adversary uses this observation. The value of mean from the previous time frame before the attack at time slot t will be μ_{t-T}. Given $\rho_{mal} = M/N$ the fraction of compromised meters and δ_{avg} is the targeted margin of false data, the estimated arithmetic mean for an additive attack is given by:

$$\mu_t = \mu_{t-T} + \Big(\rho_{mal} * \delta_{avg}\Big) \tag{11}$$

Estimating Standard Deviation: Estimation of the exact standard deviation after attack at run time is impossible for the adversary to know. This is because the adversary has only a knowledge of δ_{avg} and a subset ρ_{mal} of meters it controls, but not exact data in other non-compromised meters at that instant. However, analyzing the dataset we observed that the standard deviation is piecewise stationary. Hence, we can claim that the σ_t is a coarse approximation of the standard deviation from the previous time frame σ_{t-T}.

Estimating δ_{avg} per Each Discrete Level: Once we estimate the mean and standard deviation given the δ_{avg}, we have to estimate the average possible margin of false data for each discrete level. This calculation gives the estimation of maximum false data that can be induced in each discrete level with minimum drop in the trust score. As the current true reading is unknown, we use P_{t-T}, the smart meter reading at time slot t from the previous time frame before the attack. The maximum margin of false data for P_{t-T} in each discrete level for an additive attack is shown in Eq. (12). This will be calculated for all T readings in the time frame.

$$\begin{aligned} \delta_h^X &= \mu_{t-T} + \sigma_{t-T} - P_{t-T} & \delta_h^Y &= \mu_{t-T} + 2\sigma_{t-T} - P_{t-T} \\ \delta_h^Z &= \mu_{t-T} + 3\sigma_{t-T} - P_{t-T} & \delta_h^A &> \mu_{t-T} + 3\sigma_{t-T} - P_{t-T} \end{aligned} \tag{12}$$

Eq. (12) gives the maximum margin for individual readings. For the average value, we need the number of readings in each discrete level. Considering the number of readings in each discrete level over the previous time frame as $X_{hist}, Y_{hist}, Z_{hist}, A_{hist}$, we can calculate the average margin of false data in each discrete rating level using Eq. (13).

$$\begin{aligned} \delta_{avg}^X &= \frac{\sum_{h=1}^{X_{hist}} \delta_h^X}{X_{hist}} & \delta_{avg}^Y &= \frac{\sum_{h=1}^{Y_{hist}} \delta_h^Y}{Y_{hist}} \\ \delta_{avg}^Z &= \frac{\sum_{h=1}^{Z_{hist}} \delta_h^Z}{Z_{hist}} & \delta_{avg}^A &= \frac{\sum_{h=1}^{A_{hist}} \delta_h^A}{A_{hist}} \end{aligned} \tag{13}$$

Optimal Parameters for Evasion: The trust score can be reformulated as Eqn. 14 by combining the Eqs. (5), (6), and (7), where $W(l) = w \times l$. The value of w for each discrete level is extracted using historical data. To create an optimal evasion attack, the trust score (TR) should be just above the threshold (TH) separating the honest and malicious smart meters. At the same time, the readings should meet the targeted margin of false data. To generate the evasion data, we have to estimate the number of values in each discrete rating level that can guarantee evasion and δ_{avg}. For this, we have to solve the optimization problem in eqn. 15 to find the best values for X, Y, Z and A.

$$TR = \frac{1}{(K)^\eta}(X\ W(4) + Y\ W(3) + Z\ W(2) + A\ W(1))^\eta \tag{14}$$

$$\begin{aligned} \min \quad & (TR - TH) \\ \text{s.t.} \quad & \frac{X\delta_{avg}^X + Y\delta_{avg}^Y + Z\delta_{avg}^Z + A\delta_{avg}^A}{T} = \delta_{avg} \\ & TR \geq TH \\ & X + Y + Z + A = T \\ & X, Y, Z, A > 0 \end{aligned} \tag{15}$$

The second constraint is that best possible value for the trust score is just equal to or above the threshold. The third constraint allows to reduce the problem from 4 unknown variables to three unknown variables by replacing A with $T - X - Y - Z$.

The optimization problem has 3 unknown variables and can be solved using linear programming as all the constraints are linear. We used the simplex method to solve the formulated optimization. Upon solving the optimization problem defined above, we get the values of X,Y,Z and A. Now in this step we will generate the evasion data Q. Using the estimated values μ_t, σ_t and known true reading P_t, the δ_t values will be calculated similar to Eqn. 12 over time frame T. Then, we finally create the evasion data using eqn. 10 from the δ_t values.

Note that *the above process is shown considering an additive attack*. For deductive attacks, the only difference is in the estimation of the mean and Eq.(12), and the rest is the same.

IV. Experimental Results

We used three years (2014-16) of Texas dataset. Data from 2016 is used as testing set. The performance of the generative model is shown in terms of missed detection and false alarm degradation under data falsification attacks, with and without the involvement of generative adversarial example.

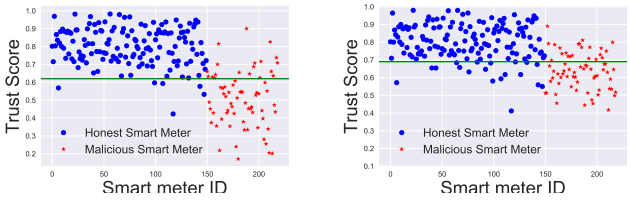


Fig. 4. Additive $\delta_{avg} = 500$ (a) no evasion (b) with evasion strategy

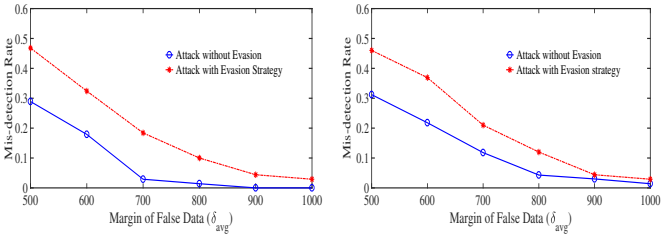
Evasion versus No Evasion: Fig. 4(a) shows an illustration snapshot of the folded gaussian scoring model without our generative evasion strategy, where we can observe a clear separation between malicious and honest meters. The detection

rate (True Positive) is 0.85 and False positive rate is 0.05. In contrast, Fig. 4(b) shows the classification performance of the same folded Gaussian model under our generative adversarial example, keeping all other attack parameters same for fair comparison. We found that the detection rate degraded to 0.55 and False positive rate increased by 0.25. This is indicative that our generative evasion strategy is degrading performance.

Generalizabilty over Different Attack Parameters: Now we investigate whether the success observed in Fig. 4(b) generalizes across any arbitrary margin of false data (δ_{avg}) and attack type. To assess this, we invoke our generative evasion strategy with different target values of δ_{avg}, and generate correspond evasion performance. We repeat this for each attack type: additive and deductive.

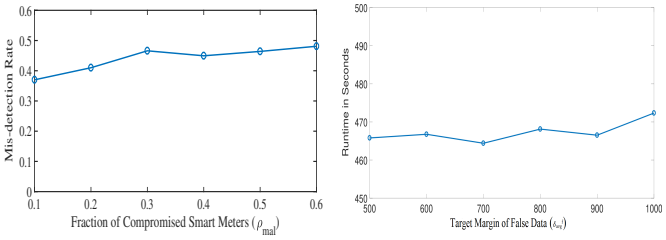
Figs. 5(a) and 5(a) respectively show missed detection rates for additive and deductive attacks across various δ_{avg} with $\rho_{mal} = 0.3$. The red line (evasion strategy) is higher than the blue line (no evasion strategy) regardless of δ_{avg}. This proves a drop in performance is due to the crafted evasion strategy.

Fig. 6(a), proves that the success of evasion attack does not get impacted by the number of compromised meters. Evidence of this can be seen from the consistency of the missed detection performance rate across different values of ρ_{mal} in Fig. 6(a) where the $\delta_{avg} = 500$.



(a) (b)

Fig. 5. Evasion Performance vs δ_{avg} (a) Additive attack (b) Deductive attack

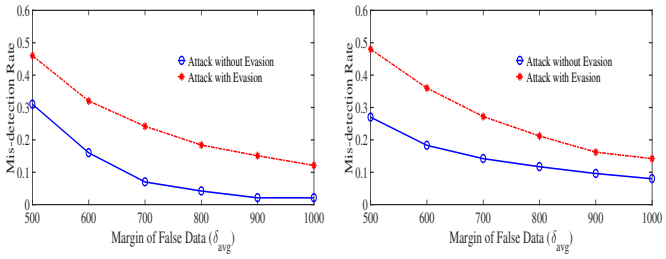


(a) (b)

Fig. 6. Cost of Evasion: (a) Performance across ρ_{mal} (b) Run times

Transferability of our Adversarial Example: To assess the feasibility, we find out the run time complexity to find values of X, Y, Z, A. It is shown to be polynomial time under most circumstances. Fig. 6(b) shows the run time required for calculating optimal results for various margins of false data. We observe that the run time scales well across δ_{avg} values.

Let us show how our generative model can be transferable to other ML based security approaches and how it degrades the corresponding performance. We choose two anomaly scoring approaches: Kullback-Leibler (KL) Distance based Scoring [8] and a generic clustering technique, DBSCAN. The resulting performance are shown in Fig. 7 for additive attack.



(a) (b)

Fig. 7. Transferability of Evasion: (a) KL Distance Trust (b) DBSCAN

V. Conclusion

In this work, we have presented the impacts of evasion attacks in smart grid. We used Generative model to create the optimal evasion samples. Finally, we demonstrated the performance of the proposed solution using real smart metering data from Texas. The results shows that the evasion strategy for falsified data created using our generator using knowledge of the Folded Gaussian Trust classifier negatively impacts the classification accuracy of compromised meter detection. Additionally, we found that our evasion strategy proved to be transferable against other approaches such as DB-SCAN and KL Distance classifer for compromised meter detection. In future, we will extend this work to handle other types of AML techniques like poisoning attacks in smart grid.

Acknowledgements: This research was supported by NSF grants: SATC-2030611, SATC-2030624, and OAC-2017289.

References

[1] M. Ester, Kriegel, J. Sander; X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise". *ACM SIG-KDD*, pp. 226–231, 1996.
[2] D. Mashima and A. A. Cárdenas,"Evaluating electricity theft detectors in smart grid networks." *Recent Advances in Intrusion Detection*, pp. 210-229, 2012.
[3] S. Bhattacharjee, P. Madhavarapu, S. Silvestri, S.K. Das, "Attack Context Embedded Data Driven Trust Diagnostics in Smart Metering Infrastructure", *ACM Trans. on Privacy and Security*, 24(2), pp.1-36, 2021.
[4] S. McLaughlin, D. Podkuiko, and P. McDaniel, "Energy theft in the advanced metering infrastructure." *Int. Workshop on Critical Information Infrastructures Security*, pp. 176-187, 2009.
[5] https://www.maximintegrated.com/content/dam/files/design/technical-documents/white-papers/smart-grid-security-recent-history-demonstrates.pdf
[6] T. Koppel, "Lights out: A cyberattack, A nation unprepared, surviving the aftermath", *Broadway Books*, 2015.
[7] S. Bhattacharjee, A. Thakur, S. K. Das, "Towards fast and semi-supervised identification of smart meters launching data falsification attacks," *ACM Asia' CCS*, pp. 173-185, 2018.
[8] S. Bhattacharjee, A. Thakur, S. Silvestri, and S. K. Das, "Statistical security incident forensics against data falsification in smart grid advanced metering infrastructure." *ACM CODASPY*, pp. 35-45. 2017.
[9] S. Bhattacharjee and S. K. Das, "Detection and forensics against stealthy data falsification in smart metering infrastructure." IEEE Transactions on Dependable and Secure Computing (TDSC), 18(1): pp. 356-371, 2021.
[10] https://www.cisco.com/c/en/us/td/docs/solutions/Verticals/Utilities/FAN/2-0/CU-FAN-2-DIG/CU-FAN-2-DIG2.html
[11] D. Li, R. Baral, T. Li, H. Wang, Q. Li, and S. Xu, "Hashtran-dnn: A framework for enhancing robustness of deep neural networks against adversarial malware." *AAAI AICS*, 2019.
[12] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. Berkay Celik, A. Swami, "Practical black-box attacks against machine learning." *ACM Asia' CCS*, pp. 506-519. 2017.
[13] I. Goodfellow, P. McDaniel, N. Papernot, "Making machine learning robust against adversarial inputs." *Comm. of the ACM*, 61(7): 56-66, 2018.
[14] M. C. Bor, A. K. Marnerides, A. Molineux, S. Wattam, U. Roedig, "Adversarial machine learning in smart energy systems." Proceedings of the Tenth ACM International Conference on Future Energy Systems, pp. 413-415. 2019.
[15] http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm