# Robust Detection of Electricity Theft Against Evasion Attacks in Smart Grids

Abdulrahman Takiddin*, Muhammad Ismail†, and Erchin Serpedin‡

*Department of Electrical and Computer Engineering, Texas A&M University at Qatar, Doha, Qatar
†Department of Computer Science, Tennessee Technological University, Cookeville, TN, USA
‡Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA

***Abstract*—Electricity theft cyber-attacks pose significant threats to smart power grids. In these attacks, malicious customers hack into their smart meters and manipulate the integrity of their energy consumption readings to reduce their electricity bills. Recently, machine learning techniques have been successfully employed to detect such cyber-attacks. However, the developed detectors have been tested against simple attacks. In this paper, we investigate the performance of electricity theft detectors against evasion attacks that are designed to reduce the reported value of the energy consumption and at the same time fool the machine learning-based detector model via adversarial samples. Furthermore, we propose a strong evasion attack that significantly degrades the performance of a set of benchmark detectors. Our results reveal that evasion attacks can deteriorate the detection rate (DR) and false alarm (FA) rate by $\sim 20\%$. To address such evasion attacks, we propose an ensemble learning-based detector that integrates auto-encoder with attention (AEA), long-short-term-memory (LSTM), and feed forward deep neural networks. The developed detector maintains a stable detection performance against evasion attacks with a deterioration in performance by only $1-5\%$ in DR and FA.**

***Index Terms*—Electricity theft, evasion attacks, cyber-attacks, smart grids, robust detector, adversarial samples.**

## I. Introduction

Electricity thefts cause financial losses of up to \$6 billion in the United States and Canada annually [1]. They also negatively impact the power grid's performance by overloading it [2]. Therefore, utility companies deploy advanced metering infrastructures, in which smart meters are capable of monitoring consumers' energy consumption regularly, which limits traditional (physical) electricity thefts [3]. Unfortunately, smart meters are vulnerable to cyber-attacks in which malicious customers hack into their meters to manipulate the electricity consumption readings to reduce their electricity bills [4].

### A. Related Work and Limitations

Machine learning (ML) techniques have been successfully employed to detect electricity theft cyber-attacks in smart grids. Both shallow and deep learning models have been adopted in literature. Shallow detectors include classifiers that employ fuzzy inference and support vector machine (SVM), which presented an accuracy of $72\%$ [5]. Another shallow learning detector exploiting an auto-regressive integrated moving average (ARIMA) representation reported a detection rate (DR) and false alarm (FA) rate of $89\%$ and $11\%$, respectively [6]. Also, a multi-class SVM-based shallow detector exhibited DR and FA of $94\%$ and $11\%$, respectively [1]. On the other hand, within the deep learning techniques, a deep feed forward model offered a DR of $92\%$ [7] and a deep belief network-based detector showed a DR of $93.7\%$ [8]. Also, a deep recurrent neural network-based detector presented a DR of $93\%$ [9], [10].

One common limitation of the aforementioned detectors is that they have been tested only against simple electricity theft attacks. These detectors have not been tested against sophisticated attacks such as evasion attacks that not only decrease the reported electricity consumption value but also can fool an ML-based detector, and hence, may go undetected. Thus, evasion attacks may deteriorate the detection performance. Until now, no reports have been mentioned in the literature about the impact of such evasion attacks on the detection performance. In addition, no effective solutions have been proposed to improve the electricity theft detectors' robustness against evasion attacks.

### B. Contributions

To overcome the limitation of the existing methods, we propose an electricity theft detector that is robust by being capable of detecting simple and evasion attacks. To achieve this objective, the following contributions are carried out:

- To quantify evasion attacks' impact, we examine the performance of an SVM model as a shallow benchmark detector as well as feed forward, long-short-term-memory (LSTM), and auto-encoder with attention (AEA) models as deep benchmark detectors throughout multiple levels of evasion attacks. We adopt two types of benchmark evasion attacks, namely, Fast Gradient Sign Method (FGSM) and Basic Iterative Method (BIM), which fool the detector by adding constant perturbations into the electricity readings. We use these benchmark evasion attacks to generate and inject adversarial samples into the test set at different attack levels. According to our simulation results, benchmark detectors suffer from $12-20\%$ performance

This publication was made possible by NPRP10-1223-160045 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

degradation in DR with the benchmark FGSM and BIM evasion attacks. Deep detectors are be $3-6\%$ more robust against evasion attacks compared to the shallow detector.
- We design stronger types of evasion attacks that extend the BIM attacks by making the perturbation values change in successive time steps via the k-nearest neighbors (KNN) algorithm. According to our simulation results, the DR performance of the benchmark detectors further decreases by $18-23\%$ when injecting the adversarial samples generated by the proposed BIM-KNN attack.
- To improve the robustness of the electricity theft detectors against evasion attacks, we design a robust electricity theft detector that fuses an AEA, LSTM, and feed forward models via sequential ensemble learning. Each individual model's output is carried over to the following model for additional processing to make a final decision. Our investigations show that our proposed detector is more robust against the benchmark and strong evasion attacks compared to both types of benchmark detectors as its performance deteriorates by only $1-5.5\%$ when tested against the strong evasion attack. For all detectors, we apply a hyper-parameter optimization algorithm based on sequential grid search that boosts the performance and decreases the computational complexity.

We organize the rest of the paper as follows. Section II discusses the utilized datasets (benign and malicious) as well as the evasion attacks. Section III presents the evasion attacks' impact on the benchmark detectors. Section IV presents the proposed detector's design as well as the simulation results. Section V concludes the paper.

## II. Dataset Preparation

In this section, we introduce the electricity consumption data that is utilized for training and testing the electricity theft detectors under investigation. The benign energy consumption dataset is adopted from the public Irish Smart Energy Trial dataset [11]. The malicious data contains six simulated general cyber-attack functions [1]. The adversarial samples are generated using three evasion attacks.

### A. Benign Dataset

For the electricity theft detectors' training and testing, we adopt the Irish Smart Energy Trial dataset. This dataset was published by the Sustainable Energy Authority of Ireland and is publicly available [11]. It contains $25,000$ readings per customer from $3,000$ smart meters of residential units that are reported once every 30 minutes along 18 months. Entry $E_c(d,t)$ of matrix $\boldsymbol{E}_c$ denotes the value of electricity consumption for customer $c$ during day $d$ and time period $t$. For honest customers, the reported energy $R_c(d,t)$ is equal to actual energy consumed $E_c(d,t)$, $R_c(d,t) = E_c(d,t)$.

### B. Malicious Dataset

For malicious customers, $R_c(d,t) \neq E_c(d,t)$. To develop a malicious dataset, we utilize the false data injection approach [1], where six cyber-attack functions $f(\cdot)$ are utilized to produce $R_c(d,t)$ under simulated cyber-attacks as follows.

- $f_1(E_c(d,t)) = \alpha E_c(d,t)$ lessens the actual energy consumption by $\alpha$, which is a constant fraction.
- $f_2(E_c(d,t)) = \beta(d,t)E_c(d,t)$ utilizes a dynamic fraction $\beta(d,t) < 1$.
- $f_3(E_c(d,t))$ records zero energy consumption during $[t_i(d), t_f(d)]$ and the real energy consumption otherwise, i.e.,

$$f_3(E_c(d,t)) = \begin{cases} 0 & \forall t \in [t_i(d), t_f(d)] \\ E_c(d,t) & \forall t \notin [t_i(d), t_f(d)]. \end{cases}$$

- $f_4(E_c(d,t)) = \mathbb{E}[E_c(d)]$ reports a constant electricity consumption value throughout the day. Operator $\mathbb{E}[\cdot]$ represents the expectation (averaging) operation.
- $f_5(E_c(d,t)) = \beta(d,t)\mathbb{E}[E_c(d)]$ considers a dynamic fraction $\beta(d,t) < 1$ of $\mathbb{E}[E_c(d)]$.
- $f_6(E_c(d,t)) = E_c(d, T-t+1)$ rearranges the recorded electricity consumption during the day to record higher consumption during the low tariff period.

We apply these cyber-attack functions to the electricity consumption profile matrix $\boldsymbol{E}_c$ of the customer. Six malicious matrices are constructed for each of the customers. Each row in a matrix depicts an energy consumption profile sample during the day and is associated with a label. If the sample is benign (malicious), the label is 0 (1).

### C. Evasion Attacks

Evasion attacks refer to manipulating malicious electricity readings in a way that makes them seem benign to fool the ML model via adversarial samples. Hence, the detector classifies them as benign. Adversarial samples are generated by applying evasion attack functions and injecting them into the test set [12]. Herein, we test different types of white-box evasion attacks as a proof-of-concept [13]. We adopt two evasion attacks, namely, FGSM [14] and BIM [15]. Also, we propose a stronger evasion attack that extends the BIM attacks using the k-nearest neighbors algorithm (BIM-KNN).

*1) Benchmark Evasion Attack Functions:* We adopt benchmark evasion attacks to generate adversarial samples to investigate the evasion attacks' impact on shallow and deep benchmark detectors. The benchmark evasion attacks rely on a constant or series of bounded perturbation values. The resulting perturbation value is a small value that is subtracted from $\boldsymbol{E}_c$ to fool the detector while reducing the electricity consumption.

*a) FGSM Attack:* This attack uses the gradients of the ML model in order to generate adversarial samples [14]. To get the perturbation value, for an input electricity reading sample $E_c(d,t)$, FGSM uses the gradients of the loss function of the model with respect to $E_c(d,t)$ to create a similar reading $R_c^{\text{adv}}$ that maximizes the loss. This is done based on a one-step gradient update along the direction of the gradient's sign at each time step. This process is represented as

$$R_c^{\text{adv}}(d,t) = E_c(d,t) - \epsilon \text{ sign}\left(\nabla_{E_c(d,t)} J\left(\phi, E_c(d,t), \boldsymbol{y}\right)\right), \quad (1)$$

where $R_c^{\text{adv}}(d,t)$ is the reported generated adversarial sample, $E_c(d,t)$ is the actual electricity reading, $\epsilon$ is the perturbation magnitude, sign refers to applying the signum function, $\nabla_{E_c}$ is the model gradients, $J$ is the model's loss function, $\phi$ denotes the model parameters, and $\boldsymbol{y}$ is the original (true) label.

*b) BIM Attack:* This attack extends the FGSM attack by applying it over time steps with a small step size $\alpha$ and clipping the obtained time series elements after each iteration [15]. It is stronger than the FGSM-based attacks since it is capable of generating adversarial samples that have similar patterns to the original readings using small changes or perturbations in an iterative manner such that [15]

$$R_c^{\text{adv}}(d,t+1) = \text{Clip}_{E_c(d,t),\epsilon}\{R_c^{\text{adv}}(d,t) - \alpha \text{sign}\left(\nabla_{E_c(d,t)} J\left(\phi, R_c^{\text{adv}}(d,t), \boldsymbol{y}\right)\right)\}, \quad (2)$$

where the clip function is applied after each time step $t$ in order to ensure that the reported readings have similar patterns to the original ones. In (2), $\alpha$ denotes the small perturbation value in each time step and $\epsilon$ is the maximum perturbation magnitude. Herein, $\epsilon = 0.1$ since having lower perturbation values decreases the chances of spotting the difference and hence increases the chance of fooling the detector.

The limitation of the benchmark evasion attacks is that in FGSM, the perturbation value is a constant, and hence, might be spotted by the detector. In BIM, the procedure is iterative, but the perturbation values are still bounded, which might be also detected. Thus, it is necessary to develop a stronger evasion attack and train the detectors accordingly to be more robust against complex types of electricity theft cyber-attacks.

*2) Proposed Evasion Attack:* To overcome the limitation of the aforementioned evasion attacks, we design a stronger evasion attack that can better fool the detector with small undetected perturbation values $\alpha$. In our proposed BIM-KNN evasion attack, $\alpha$ is different for each of the generated adversarial samples. $\alpha$ depends on the average value of a reading sample $E_c(d,t)$ and four surrounding readings. To find $\alpha$ for $E_c(d,t)$, in a sample series of readings, $\mathcal{E}_c = [E_c(d,t-2)$, $E_c(d,t-1)$, $E_c(d,t)$, $E_c(d-1,t+1)$, $E_c(d-1,t+2)]$, we get $\bar{\mathcal{E}}_c$ as the average value of the readings in $\mathcal{E}_c$. $\alpha$ at time $t$ is: $\alpha = \bar{\mathcal{E}}_c\ E_c(d,t)$. This ensures that $\alpha$ changes for each reading since each reading has different surrounding readings with different average values. The number of nearest neighbors $k$ is set to $k = 2$ in both directions of the reading to have reported values with similar average as the actual values. This way, $\alpha$ stays small while fooling the detector using small changing values that have similar patterns as the original readings. $R_c^{\text{adv}}$ is generated similar to (2), but without being bounded by $\epsilon$.

**Illustrative Example:** During the day, launching the cyber-attacks introduced in Section II.B leads to a 5 kWh average theft. Launching evasive attacks also leads to a 5 kWh average theft. However, evasive attacks are less detectable and fool the detectors, while other attacks can be easily detected.

## III. Impact of Evasion Attacks

This section first presents the benchmark detectors and the injection process of adversarial samples into the test dataset. Then, it investigates the impact of evasion attacks on the performance of the shallow and deep detectors.

### A. Benchmark Detectors

This subsection presents the shallow SVM detector as well as the deep feed forward, LSTM, and AEA detectors that we use as benchmarks to study the impact of evasion attacks.

*1) Shallow Detector:* Using shallow machine learning techniques, shallow detectors do not fully capture the patterns within in the electricity readings. The SVM model is a classifier that is static and trained on labeled data (benign and malicious) to learn and predict the labels of the samples during testing.

*2) Deep Detectors:* Using deep learning techniques, deep detectors have the capability of capturing the different patterns in the electricity reports. The feed forward-based classifier is static and does not fully capture the temporal correlation within the electricity consumption time-series data. The LSTM-based classifier is a recurrent neural network that is efficient when it comes to capturing the sequential information and temporal correlations in the customers' electricitiy consumption time-series data. The AEA-based anomaly detector is trained on the benign data only to detect anomalies (theft) in the test samples.

*a) Auto-encoder with Attention:* The AEA model consists of an encoder and decoder with LSTM recurrent layers [16], and an attention layer. The LSTM encoder's input is the reported consumption $\boldsymbol{r}$. Then, the encoder encodes the time-series vector into a hidden state. The encoder contains an input layer and succeeded by $L_{\text{A}}$ hidden LSTM layers with $N_{\text{A}}$ LSTM cells in each layer. The inputs to the successive attention layer are the encoder's output and the decoder's hidden state. This done to allocate distinct scores and weights to each time step, where the time steps that contribute more towards getting the desired output are given higher importance [17], [18]. Then, the decoder's reconstructed output and the attention layer's output are concatenated and fed into the decoder, as shown in the encoder and decoder sections of Fig. 1.

At time $t$, an LSTM cell presents a state $c_t$ and outputs a hidden state $h_t$. Accessing the LSTM cell is managed by input $i_{\text{E},t}$, output $o_{\text{E},t}$, and forget $f_{\text{E},t}$ gates for the encoder and input $i_{\text{D},t}$, output $o_{\text{D},t}$ , and forget $f_{\text{D},t}$ gates for the decoder. The LSTM cell receives the reported consumption value, $\boldsymbol{r}_t$, the previous LSTM cells' hidden states within the same layer ($h_{\text{E},t-1}$ and $h_{\text{D},t-1}$ for the encoder and decoder, respectively), and the cell state ($c_{\text{E},t-1}$ and $c_{\text{D},t-1}$ for the encoder and decoder, respectively). Specifically, we have

- $\boldsymbol{i}^l_{\text{E/D},t} = \varphi(\boldsymbol{W}^l_i \boldsymbol{r}^l_t + \boldsymbol{U}^l_i \boldsymbol{h}^l_{\text{E/D},t-1} + \boldsymbol{V}^l_i \boldsymbol{c}^l_{\text{E/D},t-1} + \boldsymbol{b}^l_i)$.
- $\boldsymbol{f}^l_{\text{E/D},t} = \varphi(\boldsymbol{W}^l_f \boldsymbol{r}^l_t + \boldsymbol{U}^l_f \boldsymbol{h}^l_{\text{E/D},t-1} + \boldsymbol{V}^l_f \boldsymbol{c}^l_{\text{E/D},t-1} + \boldsymbol{b}^l_f)$.
- $\boldsymbol{c}^l_{\text{E/D},t} = f^l_{\text{E/D},t} c^l_{\text{E/D},t-1} + i^l_{\text{E/D},t} \tanh(\boldsymbol{W}^l_c \boldsymbol{r}^l_t + \boldsymbol{U}^l_c \boldsymbol{h}^l_{\text{E/D},t-1} + \boldsymbol{b}^l_c)$.
- $\boldsymbol{o}^l_{\text{E/D},t} = \varphi(\boldsymbol{W}^l_o \boldsymbol{r}^l_t + \boldsymbol{U}^l_o \boldsymbol{h}^l_{\text{E/D},t-1} + \boldsymbol{V}^l_o \boldsymbol{c}^l_{\text{E/D},t} + \boldsymbol{b}^l_o)$.
- $\boldsymbol{h}^l_{\text{E/D},t} = \boldsymbol{o}^l_{\text{E/D},t} \tanh(\boldsymbol{c}^l_{\text{E/D},t})$

The hidden states $\boldsymbol{h}^{L_A}_{\text{E},t}$ and $\boldsymbol{h}^{L_A}_{\text{D},t-1}$ are received by the attention layer and outputs a context vector $\boldsymbol{c}_{\text{v},t}$, which is obtained via an alignment scoring function $\boldsymbol{m}$, softmax function $\boldsymbol{s}$, and multiplication layer as,

- $\boldsymbol{m} = \Gamma(\boldsymbol{h}_{\mathrm{E},t}^{L/2}, \boldsymbol{h}_{\mathrm{D},t-1}^{L})$, with feed forward model $\Gamma$.
- $\boldsymbol{s} = \exp(\boldsymbol{m}) / \sum_{|\boldsymbol{m}|} \exp(\boldsymbol{m})$.
- $\boldsymbol{c}_{\mathrm{v},t} = \sum_{T} \boldsymbol{s} \times \boldsymbol{h}_{\mathrm{E},t}^{L/2}$.

Then, the decoder's hidden layers recieve the concatenation of $\boldsymbol{c}_{\mathrm{v},t}$ and the reconstructed output $r_{\mathrm{A}}$, which is denoted by $\sum(\boldsymbol{c}_{\mathrm{v},t}, r_{\mathrm{A}})$.

*b) Long-Short-Term-Memory Neural Network:* In this deep classifier, the LSTM layers are efficient in exploiting the sequential information patterns and temporal correlation in the time series electricitiy consumption data [19]. The detector's recurrent part consists of $L_{\mathrm{M}}$ hidden LSTM layers with $N_{\mathrm{M}}$ LSTM cells in each layer.

*c) Feed Forward Neural Network:* In this deep classifier, information flows in one direction, without any loops. It consists of $L_{\mathrm{F}}$ hidden layers with $N_{\mathrm{F}}$ neurons in each layer that are used to learn more informative features.

### B. Train and Test Data

The AEA model is trained on benign data only. Thus, all customer data are concatenated and split into disjoint train and test sets with $2:1$ ratio. For the final testing, we concatenate the malicious samples with the benign test set. However, this may result in misleading results since we have more malicious than benign data. To have balanced data, we implement the adaptive synthetic sampling approach (ADASYN) [20] to over-sample the minor class. To have equal influence of all customer samples at all periods during training, we apply feature scaling to the train set, which results in a scaled train set $X_{\mathrm{TR}}$ with zero-mean and unit-variance, which is also applied to the test set $X_{\mathrm{TST}}$ with $Y_{\mathrm{TST}}$ labels.

The rest of the investigated models are multi-class classifiers that utilize benign and malicious data for training and testing. Hence, all customers' benign and malicious samples are concatenated. To balance such samples, ADASYN is employed. Then, we split the concatenated dataset into disjoint train and test sets by $2:1$ ratio. The resultant scaled version $X_{\mathrm{TR}}$ with label $Y_{\mathrm{TR}}$ as well as $X_{\mathrm{TST}}$ with label $Y_{\mathrm{TST}}$ are then obtained by applying feature scaling.

Then, to study the evasion attacks' impact on the detectors' performance while testing, we inject adversarial samples into the test data. We consider adversarial samples at different attack penetration levels where they represent $5\%$, $10\%$, and $15\%$ of the test data. We inject each type of evasion attack separately and report the simulation results accordingly.

### C. Hyper-parameter Optimization

To capture each detector's ultimate performance, we conduct hyper-parameter optimization [19]. For the deep detectors, we optimize the hyper-parameters using the following search spaces. (1) Number of hidden feed forward $L_F$ and LSTM $L_{M/A}$ layers from $\mathcal{L}_{(.)}$ = {2, 4, 6, 8}. (2) Number of neurons/cells in the hidden feed forward $N_F$ and LSTM layers $N_{M/A}$ from $\mathcal{N}_{(.)}$ = {100, 200, 300, 500, 1000}. (3) Optimizers $O$ from $\mathcal{O}$ = {Adam, Adamax, Adadelta, SGD}. (4) Dropout rate $B$ from $\mathcal{B}$ = {0, 0.2, 0.4, 0.5}. (5) Weight constraints $G$ from $\mathcal{G}$ = {0, 1, 3, 5}. (6) Hidden activation functions $A_H$ from $\mathcal{A}_{\mathrm{H}}$ = {ReLU, Sigmoid, Linear, tanh} and Output activation functions $A_O$ from $\mathcal{A}_{\mathrm{O}}$ = {Softmax, Sigmoid}. To lessen the computational complexity, we perform sequential grid-search where we optimize each hyper-parameter in sequential steps separately to get the ultimate value during each step.

### D. Performance Evaluation

This subsection presents first the evaluation metrics. It then presents simulation results that quantify the impact of the evasion attacks on the benchmark detectors.

*1) Evaluation Metrics:* Let true positive (TP) refer to a malicious reading that is correctly determined as malicious while true negative (TN) refers to a benign reading that is determined as benign. False positive (FP) denotes a benign reading that is incorrectly determined as malicious and false negative (FN) refers to a malicious reading that is incorrectly determined as benign. As performance metrics, we use detection rate (DR = TP/(TP + FN)) to determine the malicious readings that are detected as malicious and false alarm (FA = FP/(FP+TN)) to determine benign readings that are detected as malicious.

*2) Evaluation Results:* To train the detectors, we use Keras sequential API. The number of epochs $I = 50$ and the batch size $K = 100$. We use SGD optimizer, 0 weight constraint and dropout rate, ReLU hidden activation function, and Sigmoid output activation function as the initial hyper-parameters.

*a) Optimal Hyper-parameters:* The optimal hyper-parameters for the feed forward model are: 6 layers with 500 neurons, Adamax optimizer, no dropout rate, weight constraint of 3, ReLU hidden activation function, and Sigmoid output activation function. For the LSTM model: 8 layers with 300 cells, Adam optimizer, 0.2 dropout rate, weight constraint of 5, ReLU and Softmax hidden and output activation functions, respectively. For the AEA: the encoder contains 3 layers with $(500, 300, 200)$ LSTM cells and the decoder contains 3 layers with $(200, 300, 500)$ LSTM cells, SGD optimizer, no dropout rate, weight constraint of 1, and Sigmoid for the hidden and output activation function.

*b) Theft Detection:* For the AEA, to recognize benign samples from malicious ones, we compare the reconstruction error to a threshold, which is determined by the median of the interquartile range (IQR) of the receiver operating characteristic (ROC) curve. If the score is below the threshold, the sample is considered as benign, else malicious. This gives the predicted label $\boldsymbol{Y}_{\mathrm{PRED}}$. The optimal threshold is found to be $0.51$. For the LSTM and feed forward classifiers, we directly get $\boldsymbol{Y}_{\mathrm{PRED}}$. For each model, we compare $\boldsymbol{Y}_{\mathrm{PRED}}$ and $\boldsymbol{Y}_{\mathrm{TST}}$ to produce the confusion matrix and calculate the DR and FA.

*c) Detection Performance:* The evasion attacks' impact on the benchmark detectors is presented in Table I. We report the results using the detector's DR and FA with the generated cyber-attacks introduced in Section II.B, without adversarial samples ($0\%$) as well as with $5\%$, $10\%$, and $15\%$ of adversarial samples along with the generated cyber-attacks of Section II.B.

Table I shows the impact of the basic and strong evasion attacks on the benchmark detectors. Deep detectors outperform

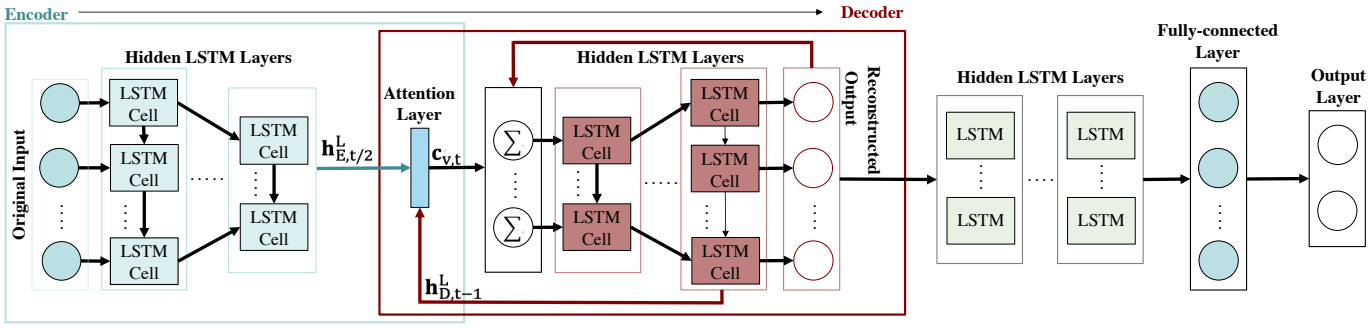


Fig. 1. Illustration of the proposed robust detector in Section IV.

TABLE I
IMPACT OF EVASION ATTACKS ON BENCHMARK DETECTORS

| | | | Evasion Percentage | | | |
|---|---|---|---|---|---|---|
| **Attack** | **Model** | **Metric** | **0%** | **5%** | **10%** | **15%** |
| **FGSM** | **SVM** | **DR** | 89.2 | 84.3 | 78.4 | 71.5 |
| | | **FA** | 10.2 | 14.6 | 21 | 27.4 |
| | **Feed Forward** | **DR** | 90.8 | 86.7 | 81.4 | 75.4 |
| | | **FA** | 9.3 | 13.5 | 18.8 | 24.2 |
| | **LSTM** | **DR** | 91.5 | 88.2 | 83 | 77.5 |
| | | **FA** | 7 | 10.9 | 15.9 | 21.8 |
| | **AEA** | **DR** | 94.1 | 91.2 | 86.6 | 81.6 |
| | | **FA** | 5.2 | 8.7 | 13.5 | 19 |
| **BIM** | **SVM** | **DR** | 89.2 | 83.8 | 77.4 | 70 |
| | | **FA** | 10.2 | 15.2 | 22.3 | 29.9 |
| | **Feed Forward** | **DR** | 90.8 | 86.5 | 80.4 | 74 |
| | | **FA** | 9.3 | 14 | 20.4 | 26.6 |
| | **LSTM** | **DR** | 91.5 | 87.4 | 81.5 | 75.4 |
| | | **FA** | 7 | 11.6 | 17.6 | 24 |
| | **AEA** | **DR** | 94.1 | 90.3 | 85.2 | 79.5 |
| | | **FA** | 5.2 | 9.4 | 15.1 | 21.1 |
| **BIM-KNN** | **SVM** | **DR** | 89.2 | 82.6 | 75.1 | 66.7 |
| | | **FA** | 10.2 | 16.4 | 24.3 | 32.9 |
| | **Feed Forward** | **DR** | 90.8 | 85.4 | 78.3 | 70.4 |
| | | **FA** | 9.3 | 15.1 | 22.3 | 29.7 |
| | **LSTM** | **DR** | 91.5 | 86.4 | 79.4 | 71.6 |
| | | **FA** | 7 | 12.4 | 18.9 | 26.3 |
| | **AEA** | **DR** | 94.1 | 89.3 | 83.3 | 76.7 |
| | | **FA** | 5.2 | 10.2 | 16.3 | 23.2 |

the shallow detector by $1.6-4.9\%$ without evasion attacks. For all detectors, the average deterioration rates in the detectors' performance with $5\%$, $10\%$, and $15\%$ of basic evasion attacks are $4.2\%$, $9.7\%$, and $15.8\%$ for the FGSM attacks, and $4.8\%$, $11.1\%$, and $17.9\%$ for the BIM attacks, respectively. For the strong evasion attack, the average deterioration rates for all detectors with $5\%$, $10\%$, and $15\%$ of the BIM-KNN evasion attack are $5.8\%$, $13\%$, and $21\%$, respectively.

## IV. Robust Electricity Theft Detection

The previous section has demonstrated the damaging impact of evasion attacks on the performance of a set of benchmark electricity theft detectors. We aim, in this section, to propose a robust detector that can maintain a stable detection performance against evasion attacks. Fig. 1 shows the architecture of the proposed detector, which places an input layer, AEA based on LSTM cells, additional recurrent layers, fully-connected layer, and output layer in sequence using sequential ensemble. These layers are placed in this specific order in order to help distinguish benign from malicious behaviors and to capture the temporal correlations in the data. Sequential ensemble extracts distinctive features by dealing with the detectors in series, which boosts the detection performance [21]. This is achieved by feeding the output of the AEA into the additional recurrent layers to capture more hidden features from the reconstructed data. After that, the output of the recurrent layers is reshaped by the fully connected layer for decision making at the output layer. The output layer consists of two neurons denoting a malicious energy consumption report and a benign report. A given reading's real label is denoted by a one-hot vector, such that $y(x_c(d)) = (0\ 1)^T$ for the honest customer, while $y(x_c(d)) = (1\ 0)^T$ for the malicious customer.

### A. Robust detector training

For the AEA, LSTM, and fully connected layers, the optimal bias values and weights are learned in the training stage. Herein, the optimization objective is to minimize the cross-entropy cost function in (3).

$$C = \min_{\Theta} \frac{-1}{|X_{\mathrm{TR}}|} \sum_{X_{\mathrm{TR}}} \{y^{\mathrm{T}}(\boldsymbol{x}) \ln(\tilde{y}) + (1 - y^{\mathrm{T}}(\boldsymbol{x})) \ln(\tilde{y})\}, \quad (3)$$

where the model parameters $\boldsymbol{W}$ and $\boldsymbol{b}$ in all AEA, LSTM, and feed forward layers are denoted by $\Theta$, the total number of training samples is represented by $|X_{\mathrm{TR}}|$ with the same number of rows as $X_{\mathrm{TR}}$, the predicted label of the detector is denoted by $\tilde{y}$, and the transposition operation is denoted by T.

We use an iterative gradient descent optimization algorithm to train the proposed robust detector. Hence, $X_{\mathrm{TR}}$ is split into equal-sized $M$ mini-batches. Then, feed forward and back-propagation are executed for $I$ (total) iterations. To compute the predicted output vectors, the training samples in the mini-batch are passed through all the network's layers in the feed forward stage. To calculate the cost function's (3) gradient given the weights of the network, the mini-batches are uzilized in the back-propagation stage [19]. To update the iterations' weights and biases, the computed gradients are utilized.

### B. Experimental Results

This section discusses the proposed detector's optimal hyper-parameters. It also evaluates the evasion attacks' impact on

the proposed detector's performance, where we use the same datasets introduced in Section II as well as the evaluation metrics, hyper-parameter optimization method, and initialization values introduced in Section III.D.

*1) Optimal Hyper-parameters:* The encoder contains 3 layers with $(500, 300, 200)$ LSTM cells and the decoder contains 3 layers with $(200, 300, 500)$ LSTM cells. The number of additional recurrent LSTM layers is 6, each additional LSTM layer contains 300 LSTM cells. The fully connected layer contains 500 neurons. The used optimizer is Adam, the dropout rate is 0, and the weight constraint is 1. ReLU and Sigmoid are used for the hidden and output activation functions, respectively.

*2) Detection Performance:* Simulation results of the performance of the proposed detector when evasion attacks are used to inject adversarial samples are shown in Table II. The results are shown using the DR and FA of the detector with the generated cyber-attacks introduced in Section II.B, without adversarial samples ($0\%$) as well as with $5\%$, $10\%$, and $15\%$ of adversarial samples along with the generated cyber-attacks. Without injecting adversarial samples, the proposed detector outperforms the benchmark detectors by $1.6 - 6.5\%$ in DR and $2.9 - 7.9\%$ in FA. The average deterioration rates of the proposed detector with $5\%$, $10\%$, and $15\%$ of the BIM-KNN evasion attacks are $1.2\%$, $2.9\%$, and $5.6\%$, respectively. This means that the proposed detector's robustness is better than the benchmark detectors by $3.5 - 5.3\%$, $7.8 - 11.1\%$, and $11.8 - 16.9\%$ in DR with $5\%$, $10\%$, and $15\%$ of the strongest evasion attacks, respectively. Additionally, with this strong evasion attack, the proposed model still offers stable performance of $90.1\%$ in DR and $7.9\%$ in FA when injecting $15\%$ adversarial samples, which still outperforms the shallow detector's performance without evasion attacks.

TABLE II
IMPACT OF EVASION ATTACKS ON THE PROPOSED DETECTOR

| | | Evasion Percentage | | | |
|---|---|---|---|---|---|
| **Attack Type** | **Metric** | **0%** | **5%** | **10%** | **15%** |
| **FGSM** | **DR** | 95.7 | 95.1 | 93.9 | 92 |
| | **FA** | 2.3 | 3.2 | 4.6 | 6.6 |
| **BIM** | **DR** | 95.7 | 94.8 | 93.4 | 91.3 |
| | **FA** | 2.3 | 3.3 | 4.9 | 7.1 |
| **BIM-KNN** | **DR** | 95.7 | 94.4 | 92.7 | 90.1 |
| | **FA** | 2.3 | 3.4 | 5.2 | 7.9 |

*3) Complexity:* The proposed detector is trained offline for only 2 hrs. When used online, a detection is made in 2 secs.

## V. CONCLUSION

This paper investigated the impact of different evasion attacks, as well as cyber-attacks, on electricity theft detectors. Specifically, we examined the impact of the basic FGSM and BIM-based evasion attacks with constant perturbation value on shallow and deep detectors. Also, we proposed a stronger evasion attack (BIM-KNN), in which the perturbation value changes in an iterative process that fools the detector and further deteriorates its performance. Based on our simulation results, the benchmark detectors severely suffer from performance degradation by $17-23\%$ in DR with evasion attacks. To enhance the performance of the detectors, we proposed a robust detector that combines AEA, LSTM, and fully connected layers using sequential ensemble. The proposed detector maintains a stable performance and deteriorates by $1.3 - 5.6\%$ in DR and FA with strong evasion attacks.

## REFERENCES


[1] P. Jokar, N. Arianpoo, and V. C. Leung, "Electricity theft detection in AMI using customers' consumption patterns," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 216–226, 2016.

[2] C. Lin, S. Chen, C. Kuo, and J. Chen, "Non-cooperative game model applied to an advanced metering infrastructure for non-technical loss screening in micro-distribution systems," *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2468–2469, Sep. 2014.

[3] V. B. Krishna *et al.*, "Evaluating detectors on optimal attack vectors that enable electricity theft and der fraud," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 4, pp. 790–805, 2018.

[4] "Electric Sector Failure Scenarios and Impact Analyses Version 3.0, National Electric Sector Cybersecurity Organization," Dec 2015. [Online]. Available: http://smartgrid.epri.com/doc/NESCOR-15.pdf

[5] J. Nagi *et al.*, "Improving SVM-based nontechnical loss detection in power utility using the fuzzy inference system," *IEEE Transactions on power delivery*, vol. 26, no. 2, pp. 1284–1285, 2011.

[6] V. Badrinath Krishna, R. K. Iyer, and W. H. Sanders, "ARIMA-Based modeling and validation of consumption readings in power grids," in *Critical Information Infrastructures Security*. Springer International Publishing, 2016, pp. 199–210.

[7] M. Ismail, M. Shahin, M. Shaaban, M. Shahin, E. Serpedin, and K. Qaraqe, "Efficient detection of electricity theft cyber attacks in ami networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6.

[8] He. Y *et al.*, "Real-Time detection of false data injection attacks in smart grid: A deep Learning-Based intelligent mechanism," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2505–2516, Sep. 2017.

[9] M. Nabil *et al.*, "Deep recurrent electricity theft detection in ami networks with random tuning of hyper-parameters," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 740–745.

[10] M. Nabil, M. Mahmoud, M. Ismail, and E. Serpedin, "Deep recurrent electricity theft detection in AMI networks with evolutionary hyper-parameter tuning," in *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2019, pp. 1002–1008.

[11] "Irish Social Science Data Archive," Last accesed: Oct 2020. [Online]. Available: http://www.ucd.ie/issda/data/commissionforenergyregulationcer/

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

[13] A. Nazemi and P. Fieguth, "Potential adversarial samples for white-box attacks," *arXiv preprint arXiv:1912.06409*, 2019.

[14] H. I. Fawaz *et al.*, "Adversarial attacks on deep neural networks for time series classification," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[15] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[16] T.-W. Sun and A.-Y. A. Wu, "Sparse autoencoder with attention mechanism for speech emotion recognition," in *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2019, pp. 146–149.

[17] Z. Zhao *et al.*, "Automatic assessment of depression from speech via a hierarchical attention transfer network and attention autoencoders," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 2, pp. 423–434, 2020.

[18] A. Vaswani *et al.*, "Attention is all you need," 2017.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[20] H. He *et al.*, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks*. IEEE, 2008, pp. 1322–1328.

[21] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Robust electricity theft detection against data poisoning attacks in smart grids," *IEEE Transactions on Smart Grid*, pp. 1–1, 2020.