Detection of Electricity Theft False Data Injection Attacks in Smart Grids

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

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

***Abstract*—Malicious customers hack into their smart meters to reduce their electricity bills using various cyberattack types. Such actions lead to financial losses and stability issues in the power grid. Existing research on machine learning-based detection offers promising detection performance. However, such detectors have been tested on a single type of cyberattacks and report performance accordingly, which is not a realistic setup since malicious customers may inject different types of cyberattacks. In this work, we examine the robustness of state-of-the-art machine learning-based electricity theft detectors against a combination of false data injection attacks (FDIAs). Specifically, we inject traditional, evasion, and data poisoning attacks with low, medium, and high injection levels then report the detection performance. Our results show that sequential ensemble learning-based detection offers the most stable detection performance that degrades only by 5.3% when subject to high injection levels of FDIAs compared to $15.7-18.5\%$ degradation rates for the stand-alone detectors.**

***Index Terms*—Electricity theft, FDIAs, cyber-attacks, smart grids, robust detection, malicious samples.**

I. Introduction

In the United States, electricity thefts lead to financial losses of $6 billion annually [1]. Additionally, inaccurate measurements lead to incorrect decisions, which may overload the power grid [2]. To overcome this, utility companies employ advanced metering infrastructures (AMIs) in which smart meters regularly report customers' energy consumption. Yet, such embedded systems are still vulnerable to various cyberattack types that customers perform in order to reduce their electricity bills. Specifically, malicious customers can launch false data injection attacks (FDIAs) to manipulate the reported values of their energy consumption readings and hence reduce their bills.

A. Related Work

Simple FDIAs reduce the reported energy consumption values via partial reduction or selective bypass techniques [1]. Different machine learning approaches have been adopted to detect such attacks including shallow and deep learning-based detectors. Shallow detectors include a detector based on an auto-regressive integrated moving average (ARIMA) model, which presented a detection rate (DR) of 89% [3]. Also, a multi-class SVM-based detector provided a DR of 94% [1]. Additionally, deep detectors based on neural networks have been investigated. For example, a deep feedforward neural network-based detector offered a DR of 92% [4]. Also, deep recurrent neural networks (RNNs) detectors presented DRs of 93% [5]. Moreover, a detection mechanism based on deep vector embeddings offered a detection rate of 95% [6]. Furthermore, deep autoencoder-based anomaly detection provided detection rates of $81-95\%$ [7], [8], [9]. Other studies investigated the impact of more complex cyberattack types including evasion [10] and data poisoning attacks [11] on detectors that are either stand-alone or based on sequential ensemble learning (SEL). The SEL-based detector provided a stable detection performance of 90.1% and 92.2% when subject to high levels of evasion and data poisoning attacks, respectively [10], [11].

This publication was made possible by NPRP12S-0221-190127 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

B. Contributions

One major limitation in the studies above is that they are only tested against a single type of electricity theft attack (i.e., either simple, data poisoning, or evasion) regardless of the attack complexity. However, in reality, malicious customers may carry out more than one type of cyberattacks. Until now, reports in the literature have not investigated the robustness of machine learning-based electricity theft detectors against a comprehensive list of cyberattack types combined together. To close this gap, we carry out the following contributions.

- To mimic the attacker's behavior that is carrying out various cyberattack types, we launch three different types of attacks, namely, simple, evasion, and data poisoning attacks that consist of a comprehensive list of cyberattack functions.
- Using a dataset of real electricity reports, we generate the cyberattacks. Then, we study the impact of such attack types under different injection levels throughout a set of experiments. We start with low injection levels where malicious samples from all attacks represent 15% of the dataset. Then, we increase the injection levels to medium (25%) and high (50%) injection levels to quantify the robustness of the electricity theft detectors accordingly.
- We examine a wide variety of stand-alone electricity theft detectors that have different properties including shallow/deep structure, static/dynamic architecture, and

supervised/unsupervised training nature. These detectors are ARIMA, single and double-class SVMs, feedforward, long-short-term-memory (LSTM), and autoencoder with attention (AEA). Our simulation results imply that, with high levels of FDIAs, the stand-alone detectors suffer from performance degradation of $15.7 - 18.5\%$. We also examine the performance of an SEL-based detector that combines AEA, LSTM, and feedforward neural networks sequentially, which offers stable performance with a degradation rate of 5.3%.

The rest of this paper is organized as follows. Section II introduces the dataset and the cyberattacks. Section III presents the detectors. Section IV discusses the experimental results. Section V concludes the paper.

II. Dataset Preparation

This section presents the electricity consumption dataset used in the study. We also introduce different types of FDIAs.

A. Benign Dataset

For detector training and testing, we utilize a dataset adopted from the publicly available Irish Smart Energy Trial dataset [12]. It contains readings during a year and a half from $3,000$ smart meters installed at customer premises that report readings in 30-minute intervals. Let $E_c(d,t)$ denote an entry of matrix $\boldsymbol{E}_c$ that depicts an actual electricity consumption value for customer c on day d during time t. The dataset of [12] presents readings from honest customers, which means that $E_c(d,t)$ and the reported readings $R_c(d,t)$ are equal.

B. Malicious Dataset

The malicious portion of the dataset contains comprehensive sets of attacks that are generated from three different attack groups using the FDIA approach [1] where $R_c(d,t) \neq E_c(d,t)$. The first group contains simple attacks in which malicious customers launch either selective bypass, partial reduction, or price-based load control attacks. The second group contains evasion attacks including fast gradient sign method (FGSM), basic iterative method (BIM), and k-nearest neighbor BIM (KNN-BIM), where customers inject adversarial samples to reduce their readings and also fool the detector. The third group contains data poisoning attacks where the detector is falsely trained on mislabeled data.

1) Simple Attacks: In this group of attacks, malicious customers carry out the following six attack functions belonging to three types to reduce their electricity bills.

a) Selective Bypass: This is represented by an attack function where malicious users selectively report zero energy consumption $R_c(d,t) = 0$ during a specific time interval $[t_i(d), t_f(d)]$ and report the actual consumption $R_c(d,t) = E_c(d,t)$ outside $[t_i(d), t_f(d)]$.

b) Partial Reduction: This is represented by two attack functions where malicious customers reduce a fraction of the reported electricity consumption either by a small constant value α where $R_c(d,t) = \alpha E_c(d,t)$ or by a dynamic random value $\beta = rand(0.1, 0.8)$ such that $R_c(d,t) = \beta_c(d,t) E_c(d,t)$.

c) Price-based Load Control: This attack is applicable in cases where the tariff is different throughout the day, and is represented by three attack functions. In the first, malicious customers report a constant consumption value $\mathbb{E}[\cdot]$ during the day according to their average consumption such that $R_c(d,t) = \mathbb{E}[E_c(d)]$. Reporting a constant value may be easily spotted. Thus, the second function adopts dynamic fraction β such that = $R_c(d,t)\beta_c(d,t)\mathbb{E}[E_c(d)]$. The last function flips the readings such that higher consumption values are reported within the low tariff periods where $R_c(d,t) = E_c(d, T-t+1)$.

Each of the simple attacks is applied to $\boldsymbol{E}_c$, which results in six malicious matrices per $\boldsymbol{E}_c$. As a result, rows represent one sample of $E_c(d,t)$ with binary labels zero or one to denote benign or malicious usage, respectively.

2) Evasion Attacks: Evasion attacks are more sophisticated where malicious (adversarial) samples are crafted to reduce the energy bill and also fool the detector, i.e., being identified as benign. Herein, adversarial samples are created using three evasion attack functions, namely, FGSM [13], BIM [14], and BIM-KNN [10] using a white-box setting [15]. The attacks below reduce $\boldsymbol{E}_c$ using small perturbation values ϵ that are either constant, bounded, or unbounded.

a) FGSM Attack: The model's gradients are used for the generation of the adversarial samples in the FGSM attack. Specifically, getting ϵ for a target reading $E_c(d,t)$ requires the detection model's gradients of the loss function so that a similar adversarial sample $R_c^{\text{adv}}(d,t)$ is generated with maximum loss.[1] This process is performed according to a one-step gradient update in the same direction of the gradient's sign at each time step where

$$R_c^{\text{adv}} = E_c - \epsilon \ \text{sign}(\nabla_{E_c} J(\phi, E_c, \boldsymbol{y})). \tag{1}$$

In (1), sign denotes the signum function. ∇, J, ϕ denote the detection model's gradients, loss function, and parameters, respectively. $\boldsymbol{y}$ refers to the correct label.

b) BIM Attack: The BIM attack expands the FGSM attack through launching it across time steps using a small ε. After each iteration, BIM clips the attained time series elements [14]. This makes ε change in each iteration, so it becomes better at fooling the detector. The generated samples have dynamic ε values that are limited by a maximum perturbation value $\hat{\varepsilon}$ resulting in generating samples with matching patterns with the original reading. BIM creates adversarial samples as follows

$$R_c^{\text{adv}}(d, t+1) = \text{Clip}_{E_c, \hat{\varepsilon}} \ R_c^{\text{adv}} - \varepsilon \ \text{sign}(\nabla_{E_c} J(\phi, R_c^{\text{adv}}, \boldsymbol{y})), \tag{2}$$

where $\hat{\varepsilon} = 0.1$ and clipping is performed after each time step t. This ensures that the reported and original readings have matching patterns with a lower chance of being noticed.

c) BIM-KNN Attack: BIM-KNN extends BIM by applying it to k neighboring readings where the value of ε varies based on the mean value of a target sample $E_c(d,t)$ as well as the k neighboring readings. For example, assuming that $k = 2$, for a given series of readings, $[E_c(d, t-1), E_c(d,t), E_c(d-1, t+1)]$ with mean M_c, $\varepsilon = M_c \ E_c(d,t)$. Therefore,

[1] t and d are dropped from $E_c(d,t)$ and $R_c^{\text{adv}}(d,t)$ for simplicity.

generating samples using BIM-KNN is similar to (2) without the bounding constrain on $\hat{\varepsilon}$. This way, ε is adjusted at each reading because each reading presents different neighboring readings with distinct mean values.

3) Data Poisoning Attacks: Data poisoning refers to cases where the detector is trained with the implicit assumption that the training labels are correct [16]. However, this assumption is not always valid. Consider the case where customers conduct electricity theft and are not being detected while their data is being used by the utility company to train the electricity theft detector. In this case, the detector will be dealing with such data as if it is benign data, when in fact, it is malicious. This results in incorrect labeling, which means that a malicious sample will be falsely assigned a benign label, which leads to data poisoning attacks [11]. Data poisoning shifts the decision boundaries of the detector, which degrades its capability of differentiating between benign and malicious samples.

C. Attack Levels

To investigate the impact of FDIAs on electricity theft detectors, we generate attacks using the aforementioned attack functions and inject them at three different levels in separate experiments. In the low, medium, and high injection level cases, cyberattacks represent 15%, 25%, and 50% of the dataset, respectively. The attack types (i.e., simple, data poisoning, and evasion attacks) are injected equally in each case. For example, with low injection levels, each attack type represents 5% of the dataset.

D. Train and Test Data

For anomaly detectors trained on benign data only, data from all customers is concatenated then split into train and test sets with a $2:1$ ratio. Malicious and benign samples are then concatenated for the final test. Imbalanced data is resolved using the adaptive synthetic sampling approach (ADASYN) [17] where benign samples are over-sampled. After applying feature scaling, X_{TR} and X_{TST} denote the training and test sets, respectively. The rest of the detectors are classifiers trained on both classes. Thus, before applying feature scaling, samples are concatenated and balanced using ADASYN.

III. Design of FDIAs Detectors

A. Stand-alone Detectors

Below, we present the stand-alone shallow ARIMA and SVM detectors along with the deep feedforward, LSTM, and AEA classifier detectors.

1) Shallow Detectors: Shallow detectors employ machine learning methods that are shallow and do not entirely capture the intricate patterns forming the electricity readings profile.

a) ARIMA-based Detection: The ARIMA model is a shallow dynamic anomaly detector trained on benign data to foresee ensuing electricity usage using minimum foreseeing mean square error (MSE). If the MSE of a sample is above a specific threshold value, it is marked as malicious [3].

b) SVM-based Detection: The single class (1-SVM) model is a shallow static detector trained only on benign samples, whereas the two-class (2-SVM) detector is trained on both classes to learn and predict samples during testing.

2) Deep Detectors: Deep detectors employ deep learning techniques allowing them to apprehend the intricate patterns behind the electricity reading data.

a) Feedforward-based Detection: Feedforward is a static 2-class classifier that is based on deep neural networks where information flows in a singular direction without forming loops.

b) LSTM-based Detection: LSTM is an RNN variation that is a deep dynamic 2-class detector that presents feedback connections. The LSTM cells hold values over time intervals where input, output, and forget gates control the information flow. Hence, the LSTM model can apprehend the intricate and temporal correlations within the time-series data.

c) AEA-based Detection: Autoencoders define anomalies based on the reconstruction error Δ. Autoencoders are used for dimensionality reduction during the encoding and for data reconstruction during the decoding [7]. Δ is the difference between the input data and reconstructed data. AEA is trained on benign data to obtain the parameters that minimize Δ. The AEA model comprises an LSTM-encoder and decoder along with an attention layer L_a [9] as demonstrated in Figure 1. An electricity reading is fed into the LSTM-encoder and encoded into a hidden state. The output is then passed to L_a that assigns higher importance to time steps with higher effects on the output [18]. L_a produces a context vector acquired via an alignment scoring and softmax functions along with a multiplication layer. The LSTM-decoder receives the concatenation of the scoring function and reconstructed output.

B. Ensemble Learning-based detection

SEL-based detection works by sequentially combining different parts where the output of one part is passed on to the next one for further feature extraction and processing [10], as shown in Fig. 1. Hence, we also examine the robustness of an SEL-based model that stacks an input layer, LSTM-based AEA, additional LSTM layers, fully-connected layer, and output layer. The rationale behind this order is to conduct further processing on the AEA's reconstructed output so that more informative features are learned and the temporal correlations are better captured. This enhances the overall detection performance. Afterwards, the LSTM output is passed and reshaped by the fully connected layer so that the final decision is made at the output layer.

C. Detector training

Using iterative gradient descent optimization, the optimal values of bias and weights are learned during training with the goal of minimizing the cross-entropy cost function:

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

where ϕ denotes the model parameters, $|X_{\text{TR}}|$ represents all training samples, $\tilde{y}$ depicts the calculated label, and T denotes

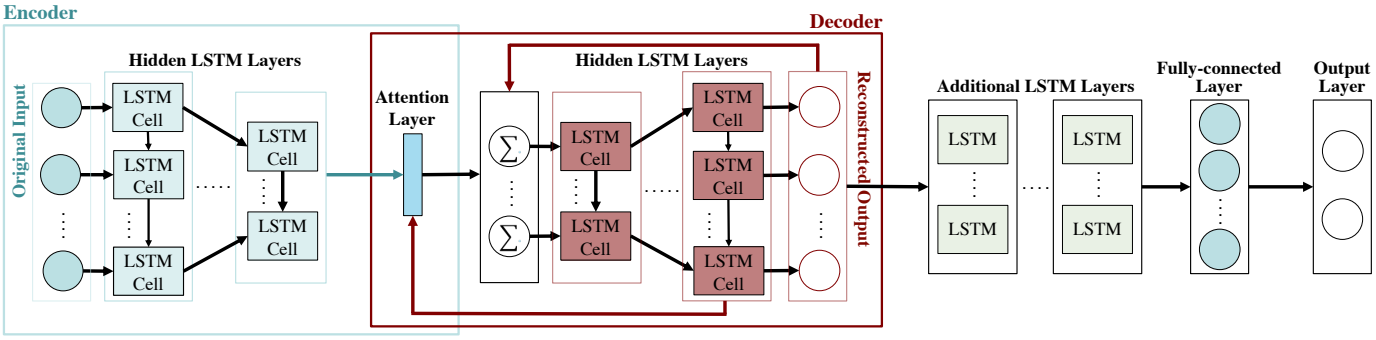

Fig. 1. Illustration of the sequential ensemble learning-based detector.

the transposition operation. We split X_{TR} into equally sized mini-batches where feedforward and backpropagation passes are executed. In the feedforward pass, the samples within mini-batch are run through all the layers. In the backpropagation pass, mini-batches are used to determine the cost function's gradient with respect to the network weights. The gradients are utilized to determine the biases and weights at each iteration.

D. Hyperparameter Tuning

Tuning the hyperparameters aids in optimizing the detection performance. Hence, we carry out sequential grid-search hyperparameter tuning in which each hyperparameter is selected in a separate stage from a predefined list until the best detection rate is achieved [19].

IV. Experimental Results

This section presents the experimental setup in terms of the used optimal hyperparameters and detection threshold values along with the numerical results of the detection performance.

A. Experimental Setup

For training, we adopt Keras sequential API with 50 epochs and a batch size of 100. Shallow and deep detectors take 1.5 and 3.5 hours to train offline, respectively. Online real-time detection takes ≤ 2 seconds.

1) Optimal Hyperparameters: After applying sequential grid-search tuning, the ensuing optimal hyperparameters are obtained. For ARIMA, the degree of differencing and moving average values are 1 and 0, respectively. For both SVM detectors, the optimal kernel and gamma are scale and sigmoid, respectively. The feedforward model has 6 layers with 500 neurons, Adamax optimizer, no dropout rate, 3 weight constraint, ReLU hidden activation function, and Sigmoid output activation function. The LSTM model has 8 layers with 300 cells, Adam optimizer, dropout rate of 0.2, and weight constraint of 5. ReLU and Softmax hidden and output activation functions are used, respectively. The AEA's encoder has 3 layers with $(500, 300, 200)$ LSTM cells that are mirrored in the decoder side, SGD optimizer, no dropout rate, 1 weight constraint. Sigmoid is utilized for the hidden and output activation function. The SEL detector has the same AEA and LSTM hyperparameters along with a fully connected layer with 500 neurons, adam optimizer, no dropout rate and 1 weight constraint, and ReLU and Sigmoid for the hidden and output activation functions, respectively.

B. Detection Threshold

For detectors trained only on benign classes, we compare the true test labels $\boldsymbol{Y}_{\text{TST}}$ with the predicted labels $\boldsymbol{Y}_{\text{PRED}}$ to get the confusion matrix. Producing $\boldsymbol{Y}_{\text{PRED}}$ requires a threshold value τ to be compared with the MSE/Δ to separate benign from malicious samples. τ is determined using the receiver operating characteristic (ROC) curve's interquartile range median. Scores greater than τ denote malicious samples. For ARIMA, 1-SVM, and AEA, the optimal τ is 0.57, 0.55, and 0.48, respectively.

C. Evaluation Metrics

A true positive (TP) is a truly detected malicious sample. A true negative (TN) is a truly detected benign sample. A false positive (FP) is a benign sample that is falsely detected as malicious. A false negative (FN) is a malicious sample that is falsely detected as benign. To evaluate the performance of the examined detectors, we use three evaluation metrics, namely, the true positive rate (TPR), false positive rate (FPR), and accuracy (ACC). TPR provides the rate of truly detected malicious samples to all samples such that TPR = TP/(TP + FN). FPR determines benign readings that are falsely detected as malicious such that FPR = FP/(FP+TN). ACC shows how well the model is able to mark benign and malicious samples such that ACC = (TP + TN)/(TP + TN + FP + FN).

D. Performance Evaluation

Table I shows the detection performance of the examined electricity theft detectors when subject to simple, data poisoning, and evasion FDIAs under different injection levels. Compared to low injection levels, the detection performance of the shallow benchmark detectors reduces by $7.5 - 8.1\%$ with medium injection levels and $17.9 - 18.5\%$ with high injection levels. Compared to shallow detectors, deep standalone detectors are around 2% more robust, where the performance reduction is $6.7 - 7.1\%$ with medium injection levels and $15.7 - 16.6\%$ with high injection levels compared to

TABLE I
IMPACT OF FDIAS ON ELECTRICITY THEFT DETECTORS.

		FDIAs Injection Levels		
Model	**Metric**	**15%**	**25%**	**50%**
ARIMA	**TPR**	71.4	63.2	52.7
	FPR	28.1	36.4	46.6
	ACC	71.3	63.5	53.0
1-SVM	**TPR**	74.8	67.1	56.8
	FPR	25.5	33.0	43.6
	ACC	75.0	67.6	57.1
2-SVM	**TPR**	76.7	69.2	59.1
	FPR	22.3	30.4	40.5
	ACC	76.7	69.7	58.6
feedforward	**TPR**	80.9	73.7	64.1
	FPR	19.6	26.9	36.1
	ACC	80.5	73.6	64.1
LSTM	**TPR**	83.2	76.2	66.1
	FPR	15.6	22.7	32.6
	ACC	83.2	76.5	65.7
AEA	**TPR**	87.1	80.5	71.3
	FPR	12.5	20.4	28.9
	ACC	86.8	81.1	71.8
SEL	**TPR**	92.7	90.4	87.6
	FPR	5.5	8.1	11.2
	ACC	92.3	90.2	87.3

the low injection levels. This enhancement is due to their deep stacked structure that apprehends the intricate patterns within the readings. Also, the LSTM and AEA present a dynamic structure that considers the temporal correlations of the readings, hence, offering better results.

The SEL-based detector offers the most stable performance among all injection levels. Superficially, compared to low injection levels, the detection performance of the SEL model reduces only by 2.3% with medium injection levels and 5.3% with high injection levels. This means that it is more robust against deep stand-alone detectors by 10%. This is mainly due to its deep and dynamic structure as well as the further processing within the different blocks.

V. CONCLUSION

This paper examined the robustness of electricity theft detectors against comprehensive sets of cyber attacks combined together. The impact of selective bypass, partial reduction, and price-based load control simple attacks along with more sophisticated FGSM, BIM, and BIM-KNN evasion attacks as well as data poisoning attacks was investigated. The attacks were injected using the FDIAs method with low, medium, and high injection levels. Several stand-alone detectors including shallow ARIMA and SVMs along with deep feedforward, LSTM, and AEA were investigated. At high FDIA levels, detection performance degrades by $15.7-18.5\%$. The impact of FDIAs on an ensemble learning-based detector that combines deep detectors sequentially was also studied. Due to its deep and dynamic structure, the ensemble detector exhibits a stable performance with 5.3% degradation at high FDIA levels. While offering promising results, further improvements will be investigated in the future to offer more stable performance to ensemble learning detectors.

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] A. Banajyoti and C. Bhende, "Performance evaluation of different machine learning techniques for detection of non-technical loss," in *Proceedings of International Conference on Communication, Circuits, and Systems*. Springer, 2021, pp. 81–87.

[3] W. H. V. Badrinath *et al.*, "ARIMA-Based modeling and validation of consumption readings in power grids," in *Critical Information Infrastructures Security*. Springer International Publishing, 2016, pp. 199–210.

[4] 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.

[5] 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.

[6] A. Takiddin, M. Ismail, M. Nabil, M. M. Mahmoud, and E. Serpedin, "Detecting electricity theft cyber-attacks in AMI networks using deep vector embeddings," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4189–4198, Oct 2020.

[7] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Deep autoencoder-based detection of electricity stealth cyberattacks in AMI networks," in *2021 International Symposium on Signals, Circuits and Systems (ISSCS)*. Iasi, Romania, 2021, pp. 1–6.

[8] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Variational auto-encoder-based detection of electricity stealth cyber-attacks in AMI networks," in *2020 28th European Signal Processing Conference (EUSIPCO)*. Amsterdam, Netherlands, Jan. 2021, pp. 1590–1594.

[9] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Deep autoencoder-based anomaly detection of electricity theft cyberattacks in smart grids," *IEEE Systems Journal*, pp. 1–12, Jan. 2022.

[10] A. Takiddin, M. Ismail, and E. Serpedin, "Robust detection of electricity theft against evasion attacks in smart grids," in *ICC 2021 - IEEE International Conference on Communications (ICC)*. Montreal, QC, Canada, Jun. 2021, pp. 1–6.

[11] 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*, vol. 12, no. 3, pp. 2675–2684, Dec 2020.

[12] "Irish Social Science Data Archive," Last accesed: Feb 2022. [Online]. Available: http://www.ucd.ie/issda/data/commissionforenergyregulationcer/

[13] 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.

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

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

[16] M. Jagielski *et al.*, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 19–35.

[17] 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.

[18] 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.

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