Multi-Layer Defense Algorithm Against Deep Reinforcement Learning-based Intruders in Smart Grids

Hossein Mohammadi Rouzbahani, Hadis Karimipour, Lei Lei

***Abstract*— High penetration of Advanced Metering Infrastructures (AMIs) and communication networks as the Internet of Energy (IoE) results in manifold security concerns, especially risk of False Data Injection Attacks (FDIAs). By fabricating the network data, FDIAs mislead power scheduling and routing strategies in IoE-based smart grids, besides monetary motivations. The conventional cyber defense systems cannot detect well-developed FDIAs, particularly once the intruder takes advantage of a Deep Reinforcement Learning (DRL)-based attack development framework that analyzes the dynamic nature of the smart grids. This paper proposes a DLR-based intruder as an active attack generator that simulates the network environment and subsequently creates unclassified FDIAs. The algorithm is initialized by various possible passive attacks, which are modeled using statistical methods. Then, a multilayer defense framework is developed using Snapshot Ensemble Deep Neural Network (SEDNN) and an adoptable Deep Auto Encoder (DAE) network to detect known and unknown threats, respectively. Performance evaluation besides a real-world simulation proves that the proposed framework can successfully detect FDIAs.**

Index Terms—False Data Injection Attack, Internet of Energy, Deep Q-Learning, Snapshot Ensemble Deep Neural Network, Deep Auto Encoder.

I. Introduction

Internet of Energy (IoE) links energy and Information and Communication Technology (ICT) to overcome emerging challenges, using modern energy management techniques and tools [1]. On the one hand, users demand to receive high-quality, reliable, and environment-friendly services with acceptable costs, guaranteeing their security and privacy. On the other hand, access to advanced real-time monitoring and controlling approaches to integrate renewable resources, maximize reliability, and minimize loss is crucial for the utilities [2].

Developing an IoE-based smart grid requires installing numerous sensors, wireless communication tools, smart appliances, and data acquisition units. While the open architecture of IoE-based networks, originating from two-way communication infrastructures and myriad internet-based entries, rises vulnerabilities against malicious activities.

False Data Injection Attack (FDIA) is one of the major and most severe threats to the network that endangers the integrity of data through bypassing the conventional bad data detection mechanisms [3]. The most vulnerable sector against FDIAs is Advanced Metering Infrastructures (AMIs) due to their scale, diversity, and complexity besides uninterrupted functionality over the communication network [4]. Three main categories of attack layouts have been introduced for FDIAs, including expert attackers, non-expert attackers, and data-driven attack models. An expert attacker is a professional adversary with complete knowledge of the nature of the system and the network topology, capable of designing an extremely complicated attack. However, even a non-expert intruder with limited information about the system can create and launch a stealth attack. Finally, da-driven attacks target the network by applying an Independent Component Analysis (ICA) to acknowledge estimated understanding about the system from the correlations of the measured data by AMIs deployed on the physical system [5].

FDIAs are commonly recognized as cyber-attacks on State Estimation (SE) in smart grids, and Bad Data Detection (BDD) methods are widely employed to detect them based on the l_2 norm between the actual and the estimated measurements [6]. Despite the fact that most of the FDIA detection techniques in the power systems focused on the SE in accordance with the line reactance data and cognizance of network topology, an attacker is still able to target the system by an FDIA in the absence of mentioned bits of knowledge. Furthermore, the fragilities of classic FDIA detection techniques become gradually prominent by facing extremely complicated attacks originating from network advancement and utilizing the gigantic quantity of AMIs and communication tools, regardless of SE data.

FDIAs have been enthusiastically investigated in terms of attack generation and detection at the same time. In [7], a linear attack generation technique with an arbitrary mean has been developed without requiring a zero-mean Gaussian distribution. The proposed attack generation framework leads to an optimal attack approach, addressing a constrained quadratic optimization problem by the Lagrange multiplier technique. Deng et al. [8] suggested an attack model aiming to launch an inexpensive technique since obtaining the system state is costly. The proposed procedure has been utilized to approximate the system states by employing a small number of power flow parameters or injection measurements. Despite the designed FDIA model in [9], which assumes that the attacker has partial knowledge of some specific measurements of the power system, the developed FDIA generation in [10] and [11] require a comprehensive understanding of different parameters.

The main shortcoming of the above-mentioned techniques is that a well-designed intelligent defense system can easily predict the modeled attacks. Besides, once the attack generation pattern is revealed, the frameworks are not capable of adapting the recent condition to create new unknown

Hossein Mohammadi Rouzbahani, Hadis Karimipour, are with University of Calgary, Calgary, AB, T2N 1N4, Canada (e-mail: hossein.mohammadirou@ucalgary.ca; hadis.karimipour@ucalgary.ca). Lei Lei is with University of Guelph, Guelph, ON, N1G 2W1, Canada (e-mail: leil@uoguelph.ca).

attacks. The optimal attack sequences have been generated by the suggested method in [12] using a dynamic game between the attacker and the network based on Reinforcement Learning (RL). Although the proposed method indicated a satisfactory performance on IEEE 39-bus systems, the attacker can be tricked by a defender, which utilizes a simulated system substitute to engage and delay the attacker. Authors [13] and [14] proposed RL-based algorithms enabling online learning and attacking. The utilized Q-Learning algorithm suffers from the lack of scalability and generalization besides the curse of dimensionality, which makes the algorithm extremely inefficient.

FDIA related investigations in the literature are not focused on the attack generation side, and many studies have been conducted on attack detection methods. Using the ex-ante admittance perturbation strategy, a hidden moving target defense approach has been proposed in [15], which the attackers cannot detect. This strategy presumes that the transmission line admittance changes at every SE interval. Liu et al. [15] presented a subsequent admittance perturbation strategy based on the differences between the column space of the measurement and attack matrices. Although the aforementioned strategies can precisely detect stealthy FDI attacks, they still rely on all network states that may not be estimated correctly due to meters placement and network topology. A joint admittance perturbation and meter protection method has been proposed in [16], aiming to increase the accuracy of estimated states under stealthy FDI attacks.

Physical protection of all utilized assets in the network is expensive and impractical, especially in large-scale systems [17]. In fact, limited network information is always available that opens a gate for malicious activities. Authors in [18] show that complete real-time knowledge is not approachable for an attacker in a real case due to inadequate access to most grid facilities. Consequently, most FDIAs occur while network topology and transmission-line admittance values are not utterly clear to the attacker.

Recently FDIAs models exploit the transmitted data over communication links among nodes and data centers that lead to generating highly complex big data. Accordingly, machine learning techniques are extensively considered as an attack detection solution since conventional methods are not capable of feature engineering and finding complex patterns [19]. Supervised and semi-supervised learning algorithms based on Support Vector Machine (SVM) have been employed in [20] to develop an attack detection procedure that has been examined on various IEEE test systems. The results show the superiority of the proposed methods (to detect both known and unknown attacks) over techniques that employ state vector estimation. Lee et al. [21] proposed a cyber threat detection approach based on the difference between True Positive (TP) and False Positive (FP) rates. The outcome demonstrates that a combination of event profiling for data preprocessing and Deep Neural Network (DNN) algorithms, including Convolutional Neural Network (CNN) and Long-Short-Term Memory (LSTM), is capable of detecting FDIAs with 6% higher accuracy than conventional machine-learning methods. Moreover, electricity theft which is a primary concern for utilities, has been investigated using different machine learning techniques. Although authors in [22] demonstrate the superiority of Artificial Neural Network (ANN) over Decision Tree (DT) and Random Forest (RF) for detecting electricity theft as an FDIA, in [23] and [25], it has been shown that CNN-based methods performed a better attack detection rate by a considerable difference.

Neither the studies mentioned above nor other related works in the context of FDIA detection in IoE-based smart grids present a framework to develop an intelligent intruder who designs attacks adapting to the dynamic environment of the smart grid. Moreover, a multilayer attack detection structure is required to detect passive and active threats simultaneously.

Motivated to address the above-mentioned concerns, the main contributions of this paper are summarized as follows.

1- An intelligent intruder is trained using Deep Q-Learning (DQL) to target the network, taking advantage of online learning by simulating a dummy power system. Moreover, various possible FDIAs are mathematically modeled to initialize the attacker algorithm.

2- As the first layer of the proposed framework, a Snapshot Ensemble Deep Neural Network (SEDNN) algorithm is developed employing the Cosine annealing technique by taking a snapshot once the model hits a local minimum before altering the learning rate. An ensemble of developed snapshots enhances the attack detection performances while reducing the risk of overfitting and computational cost.

3- A Deep Autoencoder-based network with an adaptable reconstruction error threshold is introduced as the active cyber defense to detect future unknown attacks based on the real-time information of the network. Although FDIAs are becoming more complex and intelligent, this active cyber defense makes the proposed framework more reliable in an unsupervised manner.

The remainder of this article is organized as follows. Section II presents the system model. In Section III, the DRL-based attack generation framework is introduced, initialized by the mathematical modeled possible attacks. Section IV presents the structure and algorithms of the proposed attack detection framework. The proposed model and framework are simulated in section V. Finally, section VI concludes this article.

II. SYSTEM MODEL

One of the principal characteristics of an IoE-based smart grid is to provide real-time control and monitoring of physical components anytime and anywhere [25]. As Figure 1 illustrates, the architecture of the network model consists of three main layers, including Micro Area (MA), Neighborhood Area (NA), and Wide Area (WA). Several smart meters, sensors, data concentrators, and AMI headends are placed into MAs over a local bidirectional wireless communication network. Then, an aggregator collects consumption data of all energy entities and sends the gathered information to the

attack detection unit. Lastly, the control entity takes an appropriate action based on the status of the detection unit that shows whether the system is under attack or not.

A group of MAs forms a NA, exchanging electrical energy based on their contract. A neighborhood data aggregator collects overall data of every participated MA. Next, the utilized attack detection module examines data correctness and declares the attack status to the next unit. The same process takes place at the WA level, considering collected information from two or many NAs. All the embedded sensors report a network parameter according to their assignments. This model takes consumed power as the reported measure with a specific sampling rate in a MA.

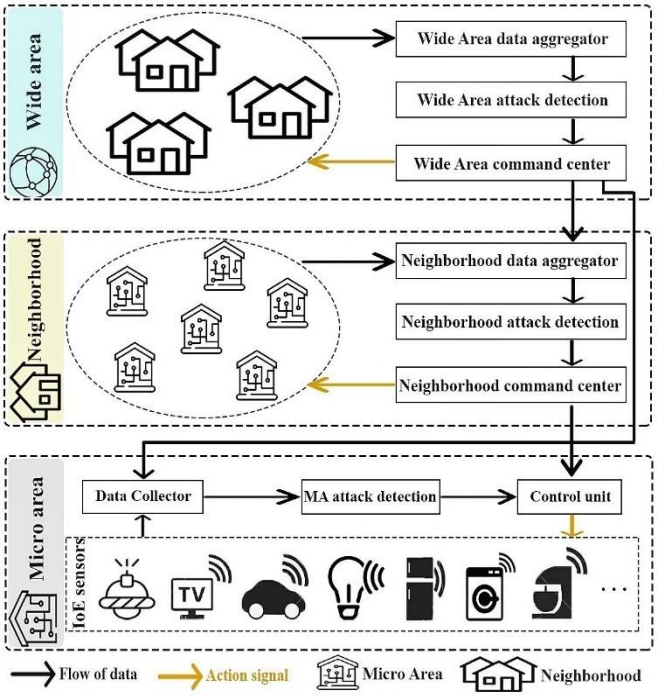

Figure 1.The proposed framework of the IoE-based network

All the embedded sensors report a network parameter according to their assignments. This model takes consumed power as the reported measure with a specific sampling rate in a MA. The logic is extendable for other parameters and NA and WA in the same way. Equation (1) defines the matrix of actual power consumption $P^{Act} \in |\mathbb{R}|^{n\times m}$, where n and m are the number of time slots (e.g., if reading is reported every 15 minutes, then n=96) and the number of energy components, respectively. The vector $C_j = (c_{1j}, c_{2j}, \cdots, c_{nj})^T$ denotes reported daily consumption of appliance j in different time slots, where c_{ij} indicates reported consumed power by sensor j at the specific time slot i.

$$P^{Act}(c) = \sum_{i=1}^{m} C_i = \begin{bmatrix} c_{11} & \cdots & c_{1m} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nm} \end{bmatrix} \quad (1)$$

Generally, an intruder compromises the integrity of the information by injecting a fake data vector $\alpha \in \mathbb{R}^{n\times m}$. Mathematically, conventional FDIAs are formulated as in (2), where P^{False} is the falsified matrix [26].

$$P^{False}(c) = P^{Act}(c) + \alpha \quad (2)$$

This research takes the capability of node selection in all different locations for the attacker into account. Also, the intruder can schedule the attack on continuous or many discrete time slots. Accordingly, the formulation of FDIAs is modified in (3), where $f(c_{ij}) = \psi_1 c_{ij}^{\beta} + \psi_2 c_{ij}^{\beta-1} + \cdots + \psi_k$ denotes applying function by the attacker on the matrix of measurements, also ψ, and β are constants and $k \in \mathbb{R}$.

$$P^{False}(c) = \begin{bmatrix} f_{11}(c_{11}) + \alpha_{11} & \cdots & f_{1m}(c_{1m}) + \alpha_{1m} \\ \vdots & \ddots & \vdots \\ f_{n1}(c_{n1}) + \alpha_{n1} & \cdots & f_{nm}(c_{nm}) + \alpha_{nm} \end{bmatrix} \quad (3)$$

III. The Proposed DQL-based Attack Generation Framework

This section introduces different parts of the designed framework, including sample library (i.e., initial attacks and normal samples), adversarial attack generator, simulated environment, and actual environment, as indicated in Figure 2.

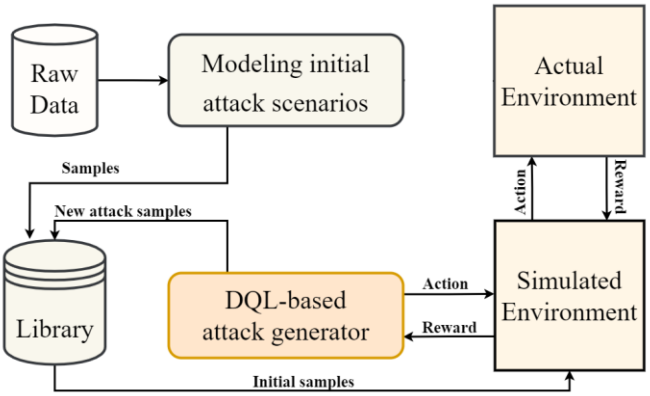

Figure 2. The framework of the proposed attack generation method

The entire process of training the attack generator algorithm is as follows.

1) Step 1: Mathematical modeling of possible FDIA scenarios

Five statistically different FDIA scenarios are modelled to store in the library as classified attacks. These attack scenarios are employed for initializing the training process of the DQL algorithm.

1.1 Node-based attack scenario

In this scenario, the attacker chooses one or multiple components and targets them regardless of time. Subsequently, corresponding columns of under attack nodes in $P^{Act}(c)$ are changed. For instance, if the first node is selected by the intruder, then the first column of $P^{Act}(c)$ is changed from $(c_{11}, \dots, c_{n1})^T$ to $(f(c_{11}), \dots, f(c_{n1}))^T$. Equation (4) demonstrates alterations of the j^{th} array in the first column of $P^{Act}(c)$ after an attack.

$$P^{False}{}_{j1} = \psi_{i1} C_{i1}^{\beta} + \psi_{i2} C_{i1}^{\beta-1} + \cdots + \psi_{ik} + \alpha_{i1} \quad (4)$$

The varying coefficients are defined based on the Joint Probability Distribution Function (JPDF) for each node considering time. Consumption and time that are denoted by C and T, are the variables and $f_{CT}: \mathbb{R}^2 \to \mathbb{R}$ is a nonnegative function so that the JPDF is defined for any set of $\mathbb{Q} \in \mathbb{R}^2$ as in (5), where $\{a, b\} \in \mathbb{Q}$.

$$P\{a < C < a + da, b < T < b + db\} = \int_{b}^{b+db} \int_{a}^{a+da} f_{CT}\ da.db \approx f(a,b)da\ db \quad (5)$$

Then, maximum and minimum JPDF for every possible pair of C and T are calculated. Finally, varying coefficients including ψ, β, and α are determined, satisfying the inequality in (6).

$$Min\ P \leq \psi_{i1} C_{i1}^{\beta} + \psi_{i2} C_{i1}^{\beta-1} + \cdots + \psi_{ik} + \alpha_{i1} \leq Max\ P \quad (6)$$

1.2 Time-based attack scenario

The second scenario occurs once all nodes are targeted at continuous or multiple discrete time slots. Consequently, the first array of the j^{th} time slot changes as indicated in (7).

$$Z_{1j} = \psi_{1j} C_{1j}^{\beta} + \psi_{2j} C_{2j}^{\beta-1} + \cdots + \psi_{kj} + \alpha_{1j} \quad (7)$$

The coefficients are calculated the same as in the previous attack scenario.

1.3 Joint node-time-based scenario

This scenario is a combination of node-based and time-based scenarios, and the attacker considers both objectives simultaneously. The coefficients are set to avoid normality test failure.

1.4 Shifting scenario

In this setup, the attacker only shifts the time of the reported consumption, one or multiple time slots. Typically, the main aim of this type of attack is bypassing high-priced tariffs during peak hours. No dummy vector is injected in the consumption matrix and just ${P^{False}}_{i(j+\Delta)} = {P^{Act}}_{ij}$, where Δ stands for the number of shifts in the time slot number.

1.5 Blind attack scenario

Blind attacks usually arise by amateur attackers intending electricity theft. The attacker has no expertise and randomly injects fake vectors. Predominantly, most injected values are zero to minimize the electricity bill amount.

2) Step 2: Training the environment simulator

It is challenging to design an optimal attack strategy in the absence of preceding knowledge. The solution is assessing the environment by trial and error, while it is crucial to attack and learn stealthily. Consequently, a dummy environment is developed to avoid revealing during the finding optimal attack strategy.

The process is started with randomly selecting samples from the highly imbalanced library with only 9% attack examples. A Long Short-Term Memory (LSTM) is designated to form the dummy environment, considering the nature of the actual environment that samples power consumption of all energy components every 15 minutes. Then a DQL algorithm is utilized to simulate the actual environment by estimating the parameters. The update rule of conventional Q-Learning is indicated in (8), where $Q(s,a)$ represents the value of action a_t in state s_t, s' is the next state by the probability of transferring from state s with action a, γ is discount factor, and r denotes the reward

$$Q_{t+1}(s,a) \leftarrow Q_t(s,a) + \alpha.\left(r + \gamma \max_{a'} Q(s',a') - Q_t(s,a)\right) \quad (8)$$

Since the process is extraordinarily slow and costly due to the required memory and time, Deep Neural Network (DNN) takes the crucial role as a function approximator in DQL, where the inputs are the states. The Q-values are calculated as the outputs, focusing on minimizing the loss function as in (9), where μ is the experience buffer containing, and θ represents the parameters of the policy.

$$L(\theta_t) = \mathrm{E}_{\mu}\left[\left(Q(s,a;\theta_t) - r_{t+1} - \gamma \max_{a} Q(s',a;\theta_t)\right)^2\right] \quad (9)$$

The selected sample is input into the simulated environment as the agent, while the $\max_{a'} Q(s',a')$ is predicted as the output. Then, the RL-environment (i.e., the actual environment, in this stage) receives the current state and the corresponding action to generate the reward. This process continues to minimize the loss function while predicting the parameters, constraints, and network topology of the simulated environment.

3) Step 3: Generating innovative FDIAs

After eliminating the risk of revealing, the second DQL-algorithm acts as an attack generator. Accordingly, the reward function is modified in this stage to distinguish the newly created attack from the previously modeled ones. Furthermore, since the attack generation section freely targets different sections of the dummy environment on various time slots, the attacker information is not limited to the local information or specific time slots. Also, there is no restriction in the cooperation and communication with the dummy environment, and providing feedback allows the attacker to define the optimal policy and improve it constantly.

The process is briefed in Algorithm 1, where Pr^{mis} denotes the probability of mis-scored, D is reply buffer, and α^{lr} indicates the learning rate.

Algorithm 1: Attack generation algorithm process

Input *D to capacity* C^{rep}, *minibatch* k^{rep}, α^{lr}
Initialize *the parameter of the dummy environment*
Inputs $S, A, \gamma, n, \epsilon$
for episode = 1, M **do**
  randomly generate a sample of sates
  initialize sequences ${S_1}^i$
  store transition in D at each episode
  get the classification result from the dummy environment
  compare the classification labels l
  if l: $True$:
    set $reward = 2$
  else:
    set $reward = 0$

```
        return reward
    Compare Pr^mis
    if Pr^mis_{t+1} > Pr^mis_t:
        set reward = −1
    else:
        set reward = 3
        return reward
    set y_j, then calculate the error
    perform gradient descent
end
```

IV. The Proposed Multilayer Attack Detection Framework

Mathematical modeling of different attack scenarios illustrated that a professional intruder could design a series of attacks that can pass conventional FDIA detection frameworks. Moreover, a comprehensive attack detection is required to detect overlooked threats since the network environment is exceptionally dynamic, and adversaries are capable of planning progressively complex and intelligent attacks.

This paper proposed a multilayer attack detection framework that combines supervised and unsupervised learning algorithms. As figure 3 shows, real-time reported information is analyzed by a SEDNN attack detection algorithm to find any malicious activities using the predefined and classified attack models in a library. Then, normal data is inserted into a Deep Auto Encoder (DAE) based unsupervised classifier to discover any possible abnormality. The developed DAE network takes advantage of an adaptable reconstruction error threshold. After detecting an attack, the library is updated to reduce detection time and cost in the future.

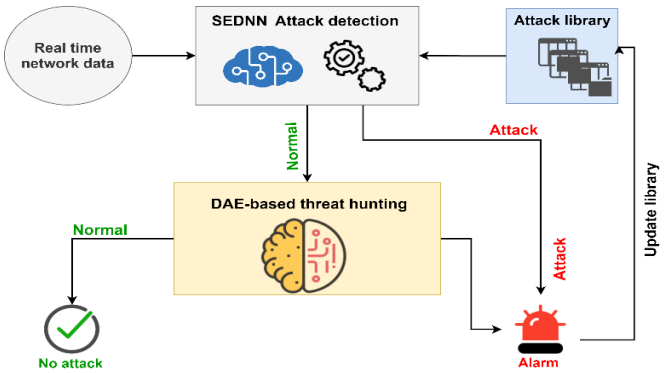

Figure 3. A schematic of the proposed attack detection algorithm

1) *The proposed Snapshot Ensemble Deep Neural Network (SEDNN) algorithm to detect passive attacks*

Ensemble learning expresses the method of training and combining multiple machine learning algorithms aiming to enhance predictive performance. Ensemble architecture of neural networks is more precise and robust than a single model due to the abilities stemming from this method, including overfitting avoidance, concept drifting, and dimensionality reduction.

The main disadvantage of the ensemble method is that training multiple DNN models is a costly process due to the extensive computational burden. Also, the best model among all trained models usually beats the ensemble method. Consequently, a snapshot ensemble that develops multiple models from a single training process is introduced as the solution. This technique combines different models' predictions while saving models during the training phase and employing them to create an ensemble setup [27]. Furthermore, the learning rate used during the training stage is aggressively altered using the Cosine annealing technique defining the initial learning rate and the number of training epochs to avoid similarity among models. In DNNs' training process, the learning rate is generally decreased after several epochs, resulting in validation loss reduction. Hence, the risk of overfitting is remarkably increased, which needs to be addressed.

The Cosine annealing method fluctuates the learning rate from a maximum to approximately zero, letting the algorithm converge to a different solution. Equation (10) formulates the learning rate α in the Cosine annealing procedure, where α_0 denotes initial learning rate; t is the iteration number, T stans for the total iteration number, and M denotes the number of cycles [28].

$$\alpha(n) = \frac{\alpha_0}{2}\left(\cos\left(\pi \times \left[\frac{T}{M}\right]^{-1} \times mod\left(t, \left[\frac{T}{M}\right]\right) + 1\right)\right) \quad (10)$$

Once the model hits a local minimum considering the validation loss, a snapshot of the model is taken, and the parameters are saved. Then, the learning rate is increased, as mentioned above, to start the training cycle of the second snapshot. An ensemble model can be developed after training N models while the number of snapshots is defined based on the total training time of all models. Equation (11) defines the process of selecting N (i.e., the number of snapshots), where $T_i^{Snapshot}$, and $T_{standard}^{DNN}$ define the training time of snapshot number i and the training time of a standard DNN, respectively.

$$T_N^{Snapshot} = T_{standard}^{DNN} - \sum_{i=0}^{N-1} T_i^{Snapshot} \quad (11)$$

2) *Developing a Deep Auto Encoder (DAE) network to detect active attacks*

Even though the previous layer is trained with numerous attack samples created by the DQL-based attack generator, there might still be unknown attacks that are capable of passing the passive attack detection layer. Accordingly, a threat hunting layer is required to enhance the detection rate. Furthermore, since the algorithm needs to detect unknown attacks, the model must be developed by unsupervised techniques.

Deep autoencoders are feed-forward multilayer neural networks consisting of an input layer, one or multiple hidden layers, and an output layer, aiming to learn data reconstructions. As a data-compression model, DAE maps the original data into a reduced dimension representation and rebuild the data from compressed information via a pair of encoder and decoder. In addition, the ability to discover correlations among data features makes DAEs capable of detecting FDIAs in an unsupervised manner.

Equation (12) shows how the encoder maps the original d

dimensional vector $(x_1, x_2, \dots, x_n)^T$ to λ number of neurons in the hidden layer h, reducing the dimension ($\lambda < n$), where h_i is the activation of the i^{th} neuron; W denotes the encoder weight matrix, b and σ stand for input bias vector and nonlinear transformation function, respectively [29].

$$h_i = \sigma\left(\sum_{k=1}^{n}(W_{ik} \times x_k) + b_i\right) \quad (12)$$

The decoder in (13) reconstructs back the hidden layer to the original space.

$$y_i = \sigma\left(\sum_{k=1}^{n}(W_{ik} \times h_k) + b_i\right) \quad (13)$$

The critical point in this model is minimizing reconstruction error, which is given in (14).

$$error = argmin\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{d}(x_i - y_i)^2 \quad (14)$$

A flat reconstruction error threshold may result in a vulnerable detection structure or even false alarms due to the dynamic nature of the attacks created by the DQL-based attack generator. The procedure of developing the adaptable DAE layer is demonstrated in Figure 4.

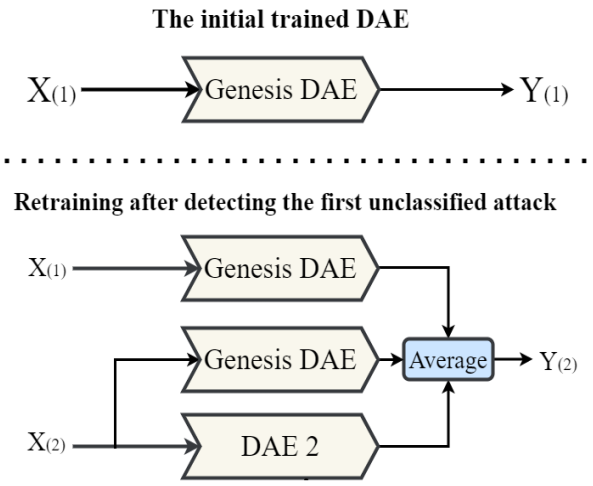

Figure 4. Schematic of DAE network

After training each training stage, the residuals $r_k = |x_k - y_k|$ are calculated to estimate the probability distribution of the outputs and the residuals, using the Radial Basis Function (RBF) kernel. Then, the marginal distribution $M(r, y_i)$ is determined as shown in (15), where $P(y, r)$ denotes the joint probability distribution.

$$P(y_i, r) = M(r, y = y_i) \times \int_{-\infty}^{+\infty} P(y_i, r).\, dr \quad (15)$$

Next, a critical point is estimated for each y_i considering the upper and lower levels of y, where $y^{uper} = 1.15 \times y$ and $y^{lower} = 0.85 \times y$. The process is done after defining a critical function and making it constant between the defined upper and lower levels.

The proposed multilayer FDIA detection framework is summarized in Figure 5.

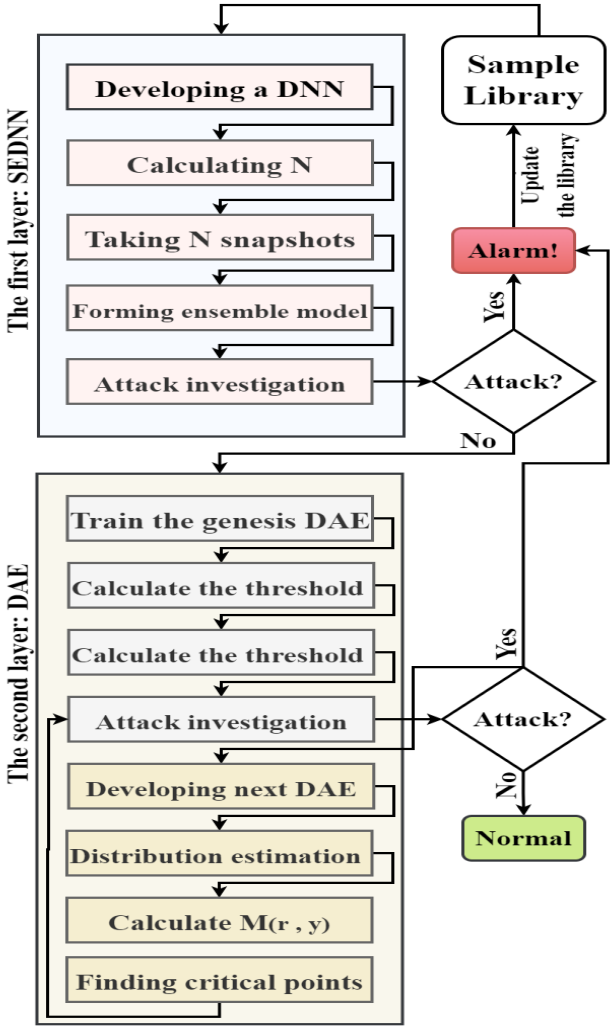

Figure 5. The procedure of the proposed attack detection layer

V. Results And Evaluations

This section first investigates the quality of the generated attacks by the DQL-based attack generation, indicating that the modeled attack scenarios can pass the presented algorithms in the literature. Then, the performance of both the active and passive layers is evaluated. All experiments are performed on a subset of the Pecan Street dataset, which is available in the Non-Intrusive Load Monitoring Toolkit (NILMTK) format [30]. Finally, a simulation examination demonstrates the feasibility, necessity, and practical outcome of the proposed FDIA detection algorithms.

1) Qualification of the developed attack generator

Three proposed FDIA detection frameworks published in top-tier journals during the last three years are selected to show their performances against the proposed attack generator framework. Artificial Neural Network (ANN), Decision Tree (DT), and Random Forest (RF) have been employed in [22] to determine attacks and anomalies in IoT sensors. Additionally, two different CNN-based mechanisms have been developed in [31] and [24], focusing on FDIAs. After some minor justifications to make the codes compatible with the dataset based on the proposed attack scenarios, the simulations show

that the generated attacks can pass the detection systems. Table 1 demonstrates that the preceding defense frameworks cannot detect the proposed attack models with a reasonable performance. Since the dataset is highly imbalanced, the accuracy does not reflect the performance of the algorithm. Accordingly, three important metrics (i.e., Recall, Precision, and F-1 score), which are not affected by the asymmetry of the dataset, are reported to illustrate the preciseness of the model. Recall is the number of correctly positive detected attacks (TP) divided by the sum of TP and the number of samples that falsely labeled as normal (FN). Precision is the ratio of TP and the sum of TP and False Positive (FP). Finally, the f-1 score is formulated in (16) as,

$$f1-score = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{16}$$

Consequently, a new attack detection is required to detect all possible attack scenarios besides hunting unknown threats.

TABLE 1
PERFORMANCE OF THE RECENTLY DEVELOPED FDIA DETECTION FRAMEWORKS AGAINST THE PROPOSED ATTACK SCENARIOS

Model	f1-score	Precision	Recall
Combined CNN [31]	0.4942	0.4548	0.5412
Wide and Deep CNN [24]	0.5186	0.5119	0.5255
SVM [24]	0.3004	0.2945	0.3066
ANN [22]	0.4627	0.4704	0.4554
DT [22]	0.2924	0.2852	0.2998
RF [22]	0.3248	0.3214	0.3284

2) Performance of the first layer: SEDNN

An ensemble of ten single models is developed using the Cosine annealing technique. The proposed SEDNN is developed using 60%, 15%, and 25% of data for training, validation, and test, respectively. Three hidden layers are defined for each model where the number of epochs is 120, and Cosine annealing learning rate cycling is 5. The batch size and learning rate are set at 256 and 0.01, respectively. This model uses the ReLU activation function while the drop-out rate is 0.3. Also, an SGD optimizer with a momentum of 0.90 is used in the model. Although the final attack detection accuracy of the first layer of the proposed framework is 96.9%, since the dataset is highly imbalanced with just 9% attack samples, f1-score, Precision, and Recall are reported to clarify the algorithm's performance.

TABLE 2
COMPARISON PERFORMANCE OF DIFFERENT METHODS

Model	f1-score	Precision	Recall
The proposed SEDNN	0.9566	0.9631	0.9502
CNN-LSTM [23]	0.8967	0.9044	0.8892
WDCNN [24]	0.8953	0.9001	0.8906
EDNN [23]	0.91879	0.9372	0.9011
RF [23]	0.7339	0.7424	0.7256

Table 2 summarizes the results and compares the performance of the developed SEDNN and other techniques, including CNN_LSTM, random bagging Ensemble of DNN (EDNN), Wide Deep CNN (WDCNN), and RF. Additionally, the superiority proposed algorithm in terms of f1-score, Precision, and Recall is investigated, making a comparison with works in [23] and [24].

3) Performance of the second layer: DAE

The normal data that pass through the first layer is then injected into the second layer, aiming to detect any unknown threat. In this stage, the data splitting procedure assigns 65% and 15% of the entire dataset to training and validation stages, while 20% of data is remained to test the developed model. Four hidden layers are embedded while the number of neurons is reduced layer by layer based on the comparison factor. Drop-out is also utilized at the rate of 0.15, mitigating the risk of overfitting and improving generalization error. An Adam optimizer is utilized to compile the DAE, and the learning rate and batch size are set at 0.001 and 512, respectively. The algorithm calculates validation errors for the first training round to define a threshold. The threshold is set as shown in (17), where IQR stands for interquartile range. The model sends an attack signal once the test error exceeds the reconstruction error threshold. Then, the threshold is adjusted as mentioned in the previous section.

$$\tau = Median + \frac{3 \times IQR}{2} \tag{17}$$

The threat hunting layer is trained 500 epochs while the validation test is monitored to avoid overfitting. Figure 6 shows the reconstruction error of 475,450 observations. The least FP rate obtains when τ is 0.815×10^{-3}. The same setup is then trained to utilize the adoptable reconstruction error.

Later, the model is tested by launching the total number of 43,480 FDIAs at diverse time slots during midnight, morning off-peak hours, midday and afternoon peak hours, and mid-load hours, with various false data injection magnitudes. The FP rate is a critical metric of attack detection in smart grids due to the severe economic and forensic consequences of a mistaken alarm. The FP rate of the proposed threat hunting layer is 0.097, along with the model accuracy of 98.82%, indicating outstanding performance.

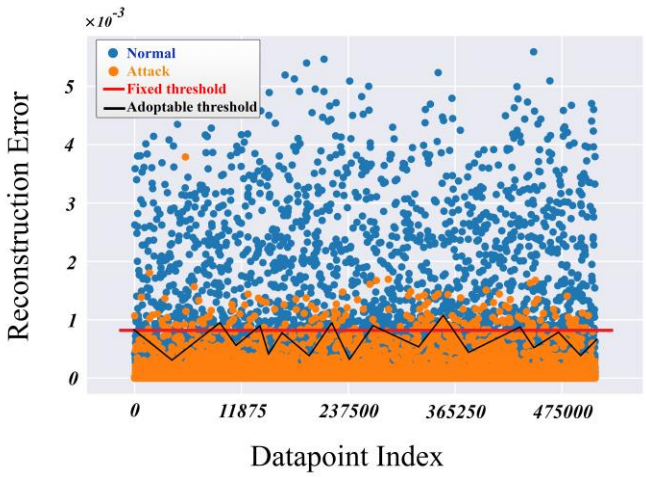

Figure 6. Reconstruction error distribution

Finally, table 3 makes a performance comparison among the second layer of the proposed method, with a flexible Bayes Classifier [32], and two well-known anomaly detection algorithms, including One-Class Support Vector Machine (OCSVM) and Isolation Forest (IF).

TABLE 3
PERFORMANCE COMPARISON AMONG ABOVE-MENTIONED METHODS

Model	Accuracy	FP rate (%)
The proposed DAE (Adoptable)	0.9882	0.97
The proposed DAE (Fixed)	0.9433	1.45
Flexible Bayes classifier	Not reported	1.92
OCSVM	0.8249	9.21
IF	0.7516	13.44

4) Network Parameters

In this section, a real-world network simulation is investigated to indicate the network's performance that operates with the developed frameworks.

The communication network and smart grid structure are modeled using ns-3 and GridLAB-D, respectively, while the Framework for Network Co-Simulation (FNCS) operates as an integrator between both simulators. Furthermore, various necessary communication and grid configurations are outlined and appended into a preprocessing module.

All created attack scenarios are scheduled and stored in a library specifying their target. As the heart of the simulator, a model engine manages and executes all the processes. Moreover, after developing the simulator, the proposed attack detection framework is embedded into the model engine to discover its performance in a real-world environment.

Figure 7 shows the architecture of the simulator and simulation parameters. Furthermore, two neighborhoods are created as a NA, ensuring the scalability of the system. The first neighborhood contains 7 MA, and the rest of 5 belongs to the second one.

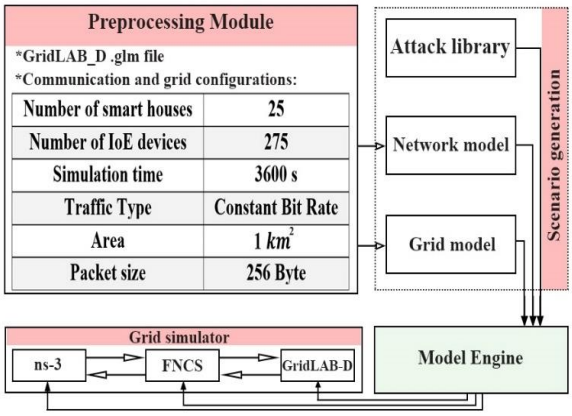

Figure 7. The structure of the GridAttackSim simulator

Two identical network topologies are also created using the developed algorithms in [23] and [24] as the pair model to compare network performances, including throughput and delay. Network throughput is a metric that indicates the amount of successfully transmitted data between transceivers in a timespan. Additionally, the average time of receiving entire information at the end node is network delay.

As Table 4 demonstrates, since the proposed method employs a DQL-based attack generation engine, most of the possible FDIAs have been classified as the detection framework at the training stage, resulting in better network performance.

TABLE 4
NETWORK PARAMETERS IMPROVEMENT WITH DIFFERENT ALGORITHMS

Method	Data rate (pkts/sec)	Throughput (kbps)	Delay (ms)
The proposed Model	2	252	126
	6	271	164
	9	364	197
CNN-LSTM [23]	2	185	258
	6	231	308
	9	262	361
WDCNN [24]	2	192	212
	6	219	278
	9	269	349

Hitherto, we indicate that the model performs properly from attack detection and network performances points of view. However, under the same attack scenarios, without an intelligent FDIA protection system, the system was targeted over 24 hours, resulting in average reducing the electricity bill for the attacker to 43%, which is not noticeable by the conventional inspections. Consequently, the net profit of the power supplier dropped 86%. The absence of the proposed framework results in chaos in power scheduling and routing, especially in the neighborhood area, affecting peer-to-peer electricity trading among the end-users.

VI. CONCLUSION

In this paper, a DQL-based FDIA generator has been developed using various possible attack scenarios that were mathematically modeled. Moreover, a two-layer attack detection framework was developed to identify both known and unknown attacks. The first layer used a SEDNN that presented better performance than other machine learning-based techniques, including EDNN, DNN, and RF, where the accuracy and f1-score of the model were 98.02% and 95.99%, respectively.

Threat hunting responsibility was assigned to the second layer using a DAE model that indicated an outstanding result where the FP rate was remarkably low by 2.9%. Also, compared to two other anomaly detection techniques, including OCSVM and IF, the designed model performed better by accuracy and f-1score of 98.81% and 95.66%, respectively.

Ultimately, the proposed attack modeling and detection framework were simulated using a combination of ns-3, FNCS, and GridLAB-D simulators. Additionally, the same setup was modeled based on two different developed algorithms to make a comparison between the performances. The result showed the superiority of the proposed framework in terms of network throughput and end-to-end delay.

VII. References

[1] H. Hua, J. Cao, G. Yang, and G. Ren, "Voltage control for uncertain stochastic nonlinear system with application to energy Internet: Non-fragile robust H∞ approach," *Journal of Mathematical Analysis and Applications*, vol. 463, no. 1, pp. 93–110, Jul. 2018, doi: 10.1016/J.JMAA.2018.03.002.

[2] H. M. Rouzbahani, H. Karimipour, and L. Lei, "A review on virtual power plant for energy management," *Sustainable Energy Technologies and Assessments*, vol. 47, p. 101370, Oct. 2021, doi: 10.1016/J.SETA.2021.101370.

[3] H. Karimipour and V. Dinavahi, "On false data injection attack against dynamic state estimation on smart power grids," *2017 5th IEEE International Conference on Smart Energy Grid Engineering, SEGE 2017*, pp. 388–393, Sep. 2017, doi: 10.1109/SEGE.2017.8052831.

[4] A. Hansen, J. Staggs, and S. Shenoi, "Security analysis of an advanced metering infrastructure," *International Journal of Critical Infrastructure Protection*, vol. 18, pp. 3–19, Sep. 2017, doi: 10.1016/J.IJCIP.2017.03.004.

[5] V. Krishnan and F. Pasqualetti, "Data-Driven Attack Detection for Linear Systems," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 671–676, Apr. 2021, doi: 10.1109/LCSYS.2020.3005102.

[6] H. Karimipour and V. Dinavahi, "Extended Kalman Filter-Based Parallel Dynamic State Estimation," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1539–1549, May 2015, doi: 10.1109/TSG.2014.2387169.

[7] H. Guo, J. Sun, and Z.-H. Pang, "Stealthy false data injection attacks with resource constraints against multi-sensor estimation systems," *ISA Transactions*, Mar. 2022, doi: 10.1016/J.ISATRA.2022.02.045.

[8] R. Deng, P. Zhuang, and H. Liang, "False Data Injection Attacks Against State Estimation in Power Distribution Systems," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2871–2881, May 2019, doi: 10.1109/TSG.2018.2813280.

[9] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, Jun. 2011, doi: 10.1145/1952982.1952995.

[10] A. Anwar, A. N. Mahmood, and M. Pickering, "Modeling and performance evaluation of stealthy false data injection attacks on smart grid in the presence of corrupted measurements," *Journal of Computer and System Sciences*, vol. 83, no. 1, pp. 58–72, Feb. 2017, doi: 10.1016/J.JCSS.2016.04.005.

[11] G. Dán and H. Sandberg, "Stealth attacks and protection schemes for state estimators in power systems," *2010 1st IEEE International Conference on Smart Grid Communications, SmartGridComm 2010*, pp. 1–6, 2010, doi: 10.1109/SMARTGRID.2010.5622046.

[12] Z. Ni and S. Paul, "A Multistage Game in Smart Grid Security: A Reinforcement Learning Solution," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 9, pp. 2684–2695, Sep. 2019, doi: 10.1109/TNNLS.2018.2885530.

[13] Y. Chen, S. Huang, F. Liu, Z. Wang, and X. Sun, "Evaluation of reinforcement learning-based false data injection attack to automatic voltage control," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 2158–2169, Mar. 2019, doi: 10.1109/TSG.2018.2790704.

[14] S. Paul and Z. Ni, "A Study of Linear Programming and Reinforcement Learning for One-Shot Game in Smart Grid Security," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2018-July, Oct. 2018, doi: 10.1109/IJCNN.2018.8489202.

[15] C. Liu, J. Wu, C. Long, and D. Kundur, "Reactance Perturbation for Detecting and Identifying FDI Attacks in Power System State Estimation," *IEEE Journal on Selected Topics in Signal Processing*, vol. 12, no. 4, pp. 763–776, Aug. 2018, doi: 10.1109/JSTSP.2018.2846542.

[16] C. Liu, H. Liang, T. Chen, J. Wu, and C. Long, "Joint Admittance Perturbation and Meter Protection for Mitigating Stealthy FDI Attacks against Power System State Estimation," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1468–1478, Mar. 2020, doi: 10.1109/TPWRS.2019.2938223.

[17] H. M. Rouzbahani, H. Karimipour, and L. Lei, "An Ensemble Deep Convolutional Neural Network Model for Electricity Theft Detection in Smart Grids," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, vol. 2020-October, pp. 3637–3642, Oct. 2020, doi: 10.1109/SMC42975.2020.9282837.

[18] M. A. Rahman and H. Mohsenian-Rad, "False data injection attacks with incomplete information against smart power grids," *GLOBECOM - IEEE Global Telecommunications Conference*, pp. 3153–3158, 2012, doi: 10.1109/GLOCOM.2012.6503599.

[19] D. Xue, X. Jing, and H. Liu, "Detection of false data injection attacks in smart grid utilizing elm-based ocon framework," *IEEE Access*, vol. 7, pp. 31762–31773, 2019, doi: 10.1109/ACCESS.2019.2902910.

[20] M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni, and H. V. Poor, "Machine Learning Methods for Attack Detection in the Smart Grid," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 8, pp. 1773–1786, Aug. 2016, doi: 10.1109/TNNLS.2015.2404803.

[21] J. Lee, J. Kim, I. Kim, and K. Han, "Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles," *IEEE Access*, vol. 7, pp. 165607–165626, 2019, doi: 10.1109/ACCESS.2019.2953095.

[22] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet of Things*, vol. 7, p. 100059, Sep. 2019, doi: 10.1016/J.IOT.2019.100059.

[23] M. Nazmul Hasan, R. N. Toma, A. al Nahid, M. M. Manjurul Islam, and J. M. Kim, "Electricity Theft Detection in Smart Grid Systems: A CNN-LSTM

Based Approach," *Energies 2019, Vol. 12, Page 3310*, vol. 12, no. 17, p. 3310, Aug. 2019, doi: 10.3390/EN12173310.

[24] Z. Zheng, Y. Yang, X. Niu, H. N. Dai, and Y. Zhou, "Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606–1615, Apr. 2018, doi: 10.1109/TII.2017.2785963.

[25] M. Rana, "Architecture of the internet of energy network: An application to smart grid communications," *IEEE Access*, vol. 5, pp. 4704–4710, 2017, doi: 10.1109/ACCESS.2017.2683503.

[26] H. Karimipour and V. Dinavahi, "Robust Massively Parallel Dynamic State Estimation of Power Systems Against Cyber-Attack," *IEEE Access*, vol. 6, pp. 2984–2995, Dec. 2017, doi: 10.1109/ACCESS.2017.2786584.

[27] H. M. Rouzbahani, A. H. Bahrami, and H. Karimipour, "A Snapshot Ensemble Deep Neural Network Model for Attack Detection in Industrial Internet of Things," *AI-Enabled Threat Detection and Security Analysis for Industrial IoT*, pp. 181–194, 2021, doi: 10.1007/978-3-030-76613-9_10.

[28] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot Ensembles: Train 1, get M for free," *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, Apr. 2017, doi: 10.48550/arxiv.1704.00109.

[29] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," *Wireless Telecommunications Symposium*, vol. 2018-April, pp. 1–5, May 2018, doi: 10.1109/WTS.2018.8363930.

[30] "Dataport – Pecan Street Inc." https://www.pecanstreet.org/dataport/ (accessed Mar. 09, 2022).

[31] D. Yao, M. Wen, X. Liang, Z. Fu, K. Zhang, and B. Yang, "Energy Theft Detection with Energy Privacy Preservation in the Smart Grid," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7659–7669, Oct. 2019, doi: 10.1109/JIOT.2019.2903312.

[32] M. Cui, J. Wang, and B. Chen, "Flexible Machine Learning-Based Cyberattack Detection Using Spatiotemporal Patterns for Distribution Systems," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1805–1808, Mar. 2020, doi: 10.1109/TSG.2020.2965797.