IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 3, 2022, accepted April 25, 2022, date of publication April 29, 2022, date of current version May 6, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3171262

Real-Time Detection of False Readings in Smart Grid AMI Using Deep and Ensemble Learning

MOHAMMED J. ABDULAAL[1], (Member, IEEE), MOHAMED I. IBRAHEM[2,3], MOHAMED M. E. A. MAHMOUD[4], (Senior Member, IEEE), JUNAID KHALID[1], ABDULAH JEZA ALJOHANI[1,5], (Senior Member, IEEE), AHMAD H. MILYANI[1], AND ABDULLAH M. ABUSORRAH[1], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[2]Department of Cyber Security Engineering, George Mason University, Fairfax, VA 22030, USA
[3]Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Cairo 11672, Egypt
[4]Department of Electrical and Computer Engineering, Tennessee Tech University, Cookeville, TN 38505, USA
[5]Center of Excellence in Intelligent Engineering Systems (CEIES), King Abdulaziz University, Jeddah 21589, Saudi Arabia

Corresponding author: Mohamed M. E. A. Mahmoud (mmahmoud@tntech.edu)

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number IFPRC-093-135-2020 and King Abdulaziz University, Deanship of Scientific Research (DSR), Jeddah, Saudi Arabia.

ABSTRACT In the advanced metering infrastructure, smart meters are deployed at the consumers' side to regularly transmit fine-grained electricity consumption readings to the system operator (SO) for billing and real-time load monitoring and energy management. However, fraudulent consumers may compromise their meters to launch electricity-theft cyberattacks by reporting low-consumption readings to reduce their bills. These false readings not only cause financial losses but also degrade the grid's performance because they are used for energy management and load estimate. The existing solutions in the literature focus only on securing the billing, so they are not designed to detect the attacks in real time, and thus the SO may use false readings for a long period of time in load monitoring and energy management until they are identified. In this paper, we propose a general ensemble-based deep-learning detector that enables the SO to detect false readings in real time. To do that, we first train several deep learning models on samples generated from a sliding window of the readings. Then, we use the best-performing model to train several models on different ratios of false readings and use them in our ensemble-based detector. Extensive experiments are conducted, and the results indicate that comparing to the literature, our detector can detect the false readings after sending a few false readings (around 15) comparing to the existing daily and weekly detection approaches that need 144 and 1,008 readings, respectively.

INDEX TERMS Security, false readings detection, electricity theft, AMI networks, and smart power grid.

I. INTRODUCTION

The smart grid's vision aims to upgrade the existing power grid by incorporating sensing, computation, and communication in the operation of the grid to improve reliability, efficiency, and resilience [1], [2]. The advanced metering infrastructure (AMI) is an important part of the smart grid. In AMI, smart meters (SMs) are deployed at the consumers to measure and transmit fine-grained power consumption readings to the system operator (SO). These readings are used for estimating the future load, i.e., energy demand, to manage the electricity supply in real-time [3], [4]. They are also used to enable dynamic billing, where the electricity prices change during the day to encourage consumers to reduce the consumption during peak hours to balance energy supply and demand [3]. However, fraudulent consumers may compromise their meters to launch electricity theft cyberattacks by reporting low-consumption readings to reduce their electricity bills illegally [5].

The electrical utility will suffer from financial losses because of these attacks. Actually, electricity theft is a major problem in the existing power grid. It is estimated that there is a global annual loss of 89.3 billion US dollars for utility companies occurring due to electricity theft [6], of which

The associate editor coordinating the review of this manuscript and approving it for publication was Arash Asrari.

6.4 billion dollars are in the US and Puerto Rico [3], [7]. In addition to the financial losses, the false readings may also degrade the grid performance because they are used for energy management and load monitoring. Therefore, the detection of the false readings is essential to guarantee the proper operation of the smart grid.

To detect the false-reading attacks, several machine-learning models have been investigated in the literature [3], [7]–[13]. The idea of these models is that consumers have normal consumption patterns that are relevant to their activities and lifestyle. Therefore, machine learning models are trained on benign and malicious datasets to learn normal and abnormal consumption patterns, and they use the fine-grained readings reported by the consumers to evaluate the models and identify the fraudulent consumers. However, *the existing solutions in the literature focus only on securing the billing, so they are not designed to detect fraudulent consumers in real time*, and thus, the SO may use false readings for a long period in load monitoring and energy management until they are identified. Specifically, the existing solutions aim to detect electricity theft every day [3], [14]–[16] or week [11], [17] by processing the daily or weekly consumption patterns.

Therefore, in this paper, we propose a general ensemble-deep-learning detector that enables SO to detect false readings in real time to secure not only billing but also energy management and load monitoring. *Comparing to the literature, our problem is more challenging* because the existing detectors focus only on improving the accuracy while we aim to create detectors that are not only accurate but also fast in the detection of false readings. To do that, we have used a combination of approaches, such as deep and ensemble learning, training models on different ratios of false readings, hyper parameter optimization, etc.

Our approach comprises of three phases: dataset preparation and analysis, detector design, and performance evaluation. A real power consumption dataset, which is provided by the smart project [18], is used to create the benign and malicious datasets needed to train and evaluate our detector. To create the benign dataset, we have used a sliding window with 40 readings size to create benign samples. Then, to create the malicious dataset, we applied an electricity theft cyberattack on the benign samples to compute the malicious samples. To detect false readings fast with high accuracy, we train several models on different ratios of false readings and use them in our ensemble-based detector. We train a general detector that can be used for all consumers. In addition, because our detector may detect the attacks after sending a few false readings, it produces confidence scores that can be used to make more accurate load monitoring and prediction. Extensive experiments have been executed to evaluate our detector's performance in terms of accuracy and the time needed to detect the false readings. The results indicate that our detector can detect the false readings accurately and fast. Comparing to the literature, our detector can detect the false readings after sending a few false readings (around 15) comparing to the existing daily detection approaches that need 144 readings and the existing weekly detection approaches that need 1,008 readings.

This paper makes the following main contributions.

- To the best of our knowledge, this paper is the first work that focuses on the real-time detection of false readings in AMI to secure both the energy management and billing. Specifically, the existing papers in the literature aims at securing the billing, so they detect electricity theft every long time which can be a week or a day, and hence the false readings are used for a long time in energy management and load estimate, which may lead to the inefficient operation of the grid and disturbance of the supply.
- We analyzed a dataset for real energy consumption and we found that the data has time-series and periodic nature. Based on this analysis, we propose a novel deep and ensemble-based deep learning detector to improve the accuracy of the detector and detect the false readings fast. The detector also provides confidence scores which express the credibility of the readings. These scores can be used to secure the energy management system by giving more weight to the highly credible readings.
- We conduct extensive experiments to evaluate our detector's performance, and the results demonstrate that the detector accurately detects the false-readings in much less time than the existing papers in the literature.

The rest of the paper is organized as follows. The network and threat models are discussed in Section II. In Section III, we explain the preparation of the dataset produced for training our detector. Section IV presents the details of our detector used for the real-time detection of false-reading attacks. Following that, in Section V, we discuss the performance evaluations of our detector. Furthermore, in Section VI, the existing studies in the literature that investigate the detection of the electricity theft attacks in the AMI network are discussed. Finally, we conclude the paper in Section VII.

II. NETWORK/THREAT MODELS AND DESIGN GOALS

A. NETWORK MODEL

As illustrated in Fig. 1, the network model considered in this paper has smart meters forming an AMI, where $SM = SM_i, 1 \leq i \leq |SM|$, and a system operator (SO). The SMs are deployed at the consumers' side and they transmit fine-grained power consumption readings to the SO, i.e., one reading every 10 minutes. The SO uses these readings to estimate future loads, and then manage energy supply. The readings are also used for computing bills based on dynamic tariff, where the electricity prices may change during a day multiple time to balance supply and demand, i.e., the prices are high during peak hours to reduce the demand.

B. THREAT MODEL

As shown in Fig. 1, some consumers are fraudulent and they report low energy consumption to reduce their bills illegally.

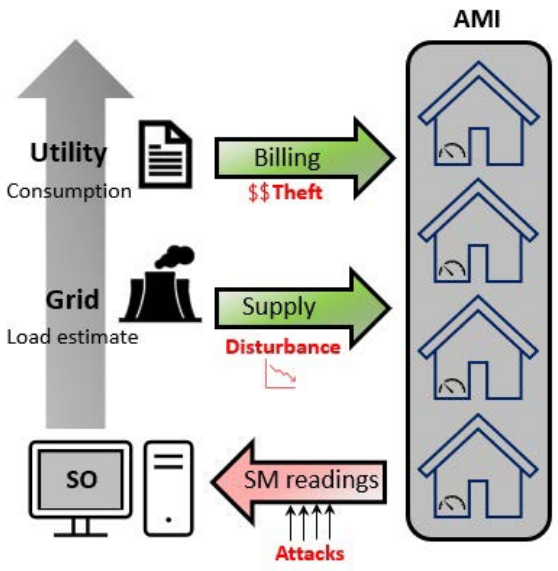

FIGURE 1. The advanced metering infrastructure (AMI) consists of many smart meters and a system operator (SO). The SO collects the smart meters' readings and computes the real-time demand to estimate the supply, and sends the consumption measurement to the utility companies for billing. The electricity theft attacks aim to reduce the readings to reduce the electricity bills, which causes financial losses and disturb the supply/demand balance.

This misbehavior may result in not only hefty financial losses to the utility but also disturbance to the energy management system by making bad decisions because the consumption readings are used for forecasting future energy demand. Practically, this attack may be launched in different ways. One way is by developing a malicious firmware that reports false readings and installing it in the SMs [19]. Several tools, such as Terminator, are being used to access the SMs and download new firmware [19]. In another way, recent research has revealed that false-reading attacks can be launched by attacking the communication network of the AMI [20]. Therefore, in this paper, a deep and ensemble learning-based detector is proposed to identify false readings in real time by processing the readings reported by the consumers' SMs.

C. DESIGN GOALS

In this paper, we aim to achieve the following goals:

- (G1) Develop a general detector that can be used for all consumers to identify false readings.
- (G2) Real-time detection of the false readings. This is measured by the number of false readings that need to be sent by the fraudulent consumer so that the detector can identify the attack.
- (G3) Confidence scores should be computed for each consumer's readings. The higher the score, the more confident the detector in the credibility of the readings. These scores can be used in load forecasting to secure the energy management system, where more weights are given to the more credible readings.
- (G4) False readings should be detected with high accuracy and low false positive.

III. DATASET PREPARATION AND ANALYSIS

A real dataset for power consumption readings issued by the Smart Project [18] is used to create the benign and malicious samples which are needed to train and evaluate the proposed detector. This dataset is available online and it includes the real power consumption readings of 114 residential apartments for 349 days in 2016, from 1st January to 14th December, with one minute granularity. Using this one-minute granularity dataset, we have derived another dataset with ten-minute granularity by aggregating the readings, which results in 144 readings for every SM in each day. Then, to create the benign dataset, we used the readings of 40 consumers for 110 days and a sliding window with the size of 40 readings, each sample corresponds to 40 readings, and by shifting the window by one reading, a new sample is generated. Note that we tried different sizes for the windows and we found that 40 readings give the best results. As a result, the cumulative number of benign samples is 633,600 ($144 \times 40 \times 110$), where each smart meter has about 15,840 samples. Fig. 2 shows the fine-grained readings of three randomly selected consumers for one year, while Fig. 3 shows the fine-grained readings of a consumer in a single day and four weeks.

To train our detector, we need both benign and malicious samples but all the given power consumption readings in the dataset are for honest consumers. Since actual malicious samples for SMs are unavailable, we used a partial reduction based electricity theft attack, proposed in [3], to mimic the behavior of fraudulent consumers and create malicious dataset. The false readings reported by the fraudulent consumers are computed as follows:

$$r_i[j] = \alpha \times tr_i[j] \tag{1}$$

where $tr_i[j]$ denotes the j^{th} true electricity consumption reading in one day for smart meter (SM_i), $r_i[j]$ denotes the corresponding reading reported by the meter, and α is a flat reduction factor. Therefore, the attacker reports the true reading reduced by the factor of α to achieve financial gains. To generate the malicious dataset, we select α randomly for each fraudulent consumer, where it is uniformly distributed in [0.3, 0.8]. Obviously, the lower α, the more financial gains the attacker can achieve.

To create the malicious dataset, we apply the attack on the 633,600 begin samples of the 40 smart meters for 110 days. After applying the attack, each smart meter has 31,680 samples including 15,840 benign samples and 15,840 malicious samples, where each sample has 40 readings with ten-minute granularity. As the number of SMs is 40, the total number of samples of the dataset is $31{,}680 \times 40 = 1{,}267{,}200$. The dataset is balanced where half of the samples are malicious and the other half are benign, i.e., 633,600 samples for each one. Then, we use these samples to create two datasets for training and testing purposes, where 80% of the dataset

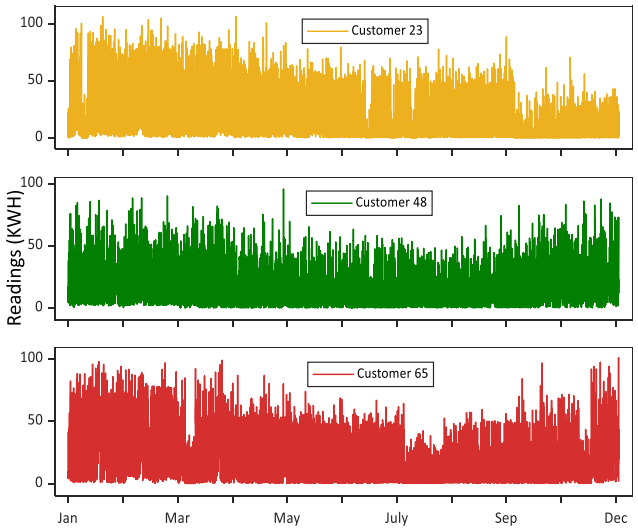

FIGURE 2. The power consumption readings of three randomly selected consumers.

(1,013,760 samples) are used for training, while 20% of the dataset (253,440 samples) are used for testing.

In addition, to design a good detector, we need to analyze the dataset first. To do that, we utilize the autocorrelation function (ACF) to assess the relation between the successive readings of the meters, i.e., the ACF offers the correlation between sequences of readings at various delay of time. Fig. 4 displays the ACFs of the readings shown in Fig. 2. The shady section of Fig. 4 represents the confidence intervals of around 95% used to assess the importance of autocorrelation at a particular time lag. We can observe from the figure that the consumption readings are time series and thus the detector should have the capability to capture and learn the time-series relation to provide accurate classifications. We can also observe from Fig. 3 that the readings have periodic patterns, so to obtain accurate classifications, the detector should learn these patterns to be able to identify anomalous patterns due to sending false readings.

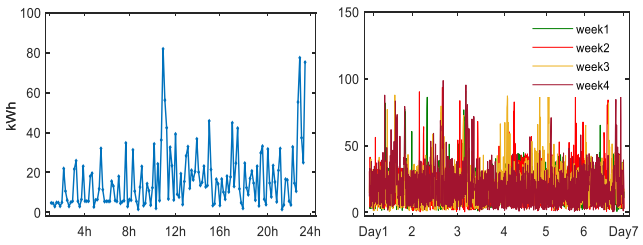

FIGURE 3. Energy consumption of consumer 23. (a) Single day consumption with 10 min granularity. (b) Consumption by week.

IV. PROPOSED DETECTOR

In this section, we first give an overview for the proposed electricity theft detector and then discuss the rationale behind the architecture of our detector. After that, we give some details on the building blocks of the detector.

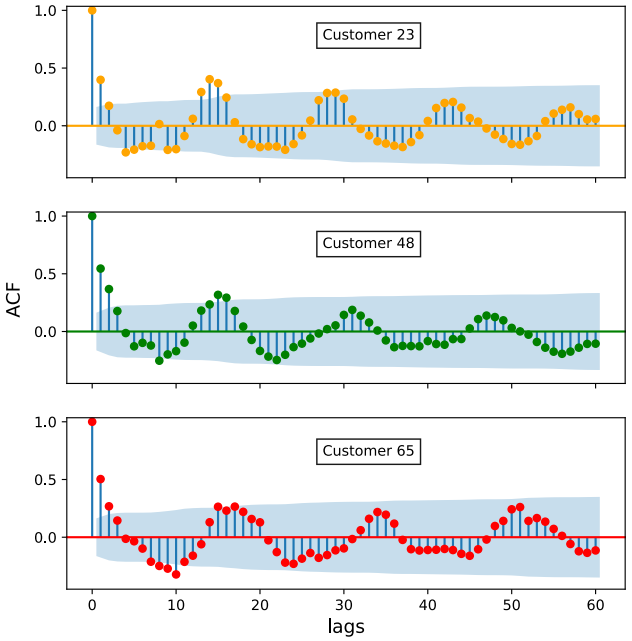

FIGURE 4. The ACFs of three randomly selected consumers.

A. OVERVIEW

Our approach comprises of three phases: dataset preparation and analysis, detector design, and performance evaluation. A real power consumption dataset, which is provided by the smart project [18], is used to create the benign and malicious datasets needed to train and evaluate our detector. To create the benign dataset, we have used a sliding window with 40 readings size to create benign samples. We tried different sizes for the window and we found that our detector gives good results when the window size is 40 readings. Then, to create the malicious dataset, we applied an electricity theft cyberattack on the benign samples to compute the malicious samples. Our analysis to the dataset indicates that the power consumption readings are time-series and have periodicity due to the periodic activities of the consumers, and thus, the detector should be able to capture and learn the time-series and periodic nature of the consumption patterns.

We first train several deep-learning models, including the feed forward neural network (FFN), the gated recurrent unit neural network (GRU), the convolutional neural network (CNN), and the long-short term memory neural network (LSTM), on the benign and malcious datasets to create binary classifiers. Then, to improve the accuracy, we use the best-performing model (GRU-based classifier) consisting of a GRU layer followed by a fully connected neural network. The GRU layer can capture the correlation between the fine-grained smart meter readings, while the fully connected neural network is used to make more accurate decisions. After that, to detect false readings fast with high accuracy, we train several models on different ratios of false readings and use them in our ensemble-based detector. Ensemble learning can make a strong model using several weak models.

Instead of training a single detector for each consumer, we train a general detector that can be used for all consumers. Single detectors need much computations to train a large number of models and they cannot detect zero-day attacks because the SO needs to collect sufficient data from each consumer to train the models. In addition, because our detector may detect the attacks after sending a few false readings, it produces confidence scores that can be used to make more accurate load monitoring and prediction. The readings are more credible when their score is high and thus more weight should be given to these readings in load forecasting to secure the energy management system.

B. RATIONALE BEHIND THE DESIGN OF OUR DETECTOR

1) MACHINE LEARNING

Several approaches that are based on game theory and state estimation to detect false readings in smart grid AMI have been presented in the literature, but machine learning-based detectors give better results than other approaches [21].

2) GENERAL DETECTOR

In the literature, there are two types of detectors; consumer-specific and general. Consumer-specific detectors are trained on the consumption readings of a specific consumer and it can only be used to detect the false readings of this consumer, while the general detectors are trained on the readings of many consumers and it can be used for detecting the false readings of all consumers. Our detector should be general because consumer-specific detectors suffer from three main problems [21]. First, because there are too many consumers and one model should be trained for each consumer, too much computation power is needed to train many models. Second, to train a model, we need to wait to collect enough data from each consumer. Third, consumer-specific detectors are vulnerable to zero-day attacks, where they cannot detect the new consumers who attack the system from the first day.

3) DEEP LEARNING

Our detector should use deep learning because it can capture complex patterns and accurately classify input data comparing to shallow learning techniques like support vector machine, decision tree, and logic regression. Recently, deep learning has been widely used in many applications and it proved that it can perform well comparing to other learning techniques [22], [23]. Moreover, we need also to use a hyper-parameter optimization technique to improve the accuracy of the model by finding the best possible parameters for the model.

4) ENSEMBLE LEARNING

Ensemble learning is a machine learning approach that uses several weak models that are usually trained on different data, to create a strong model by combining the outputs of the models to compute the final output that gives better result. Recently, ensemble learning has been widely used in many applications due to its ability to boost the accuracy of the machine learning models [24], [25].

5) CAPTURING THE TIME CORRELATIONS

Our analysis indicates that the fine-grained power consumption readings are time-series, and thus the machine learning model used should be able to capture and learn this time-series relation to provide accurate classifications.

6) BRIEF DESCRIPTION FOR OUR DETECTOR

Since our goal is not only training an accurate model but also a fast model in the detection of the false readings, we may need a combination of techniques and models to be able to achieve this goal. Based on the discussion above, our detector should be general in the sense that it can be used for every consumer, including the new consumers. The proposed detector should use deep-learning model that is able to capture the temporal correlations in the consumption readings. As shown in Fig. 5, the deep-learning model used in our detector consists of recurrent neural network (RNN) which uses a GRU layer followed by fully connected neural network. Since the smart meter's readings are time-series data with correlations among consecutive readings, the GRU architecture is used due to its capability of capturing this temporal correlation. As will be explained in Section V, we will try different deep-learning models and GRU-based model will give the best results. In training the model, hyper-parameter optimization will be used to optimize its performance. In the next subsections, we will provide more details. Finally, an ensemble learning methodology is used in order to improve the performance of the detector. As shown in Fig. 6, our detector uses multiple versions of the GRU-based models trained on different ratios of false readings. To detect the false readings attack, as shown in Fig. 7, in every time slot, we input a window of the most recent 40 readings to our detector, so, a decision is made every time slot. A brief description on the models and methods used in the proposed detector is presented in the following subsections.

C. DEEP LEARNING

Basically, deep learning models are based on neural networks that usually comprises three types of layers: input, hidden, and output [11]. Convolutional neural network (CNN), multi-layer perceptron (MLP), and recurrent neural network (RNN) are few examples of deep learning models that can be used for detecting false readings in smart grid. To train the deep learning models, the input data (or features) is fed into the model layers. Then, for a predetermined number of iterations, the features progressively are mapped into higher order abstractions through iterative update via the model's intermediate layers. Finally, these mapping abstractions are used by the output layer to make the final decision. The objective of training a model is to learn the weight and bias parameters of the model's layers and this can be achieved by defining an objective function and using an optimizer. Using the gradients of the objective function, the weights and biases

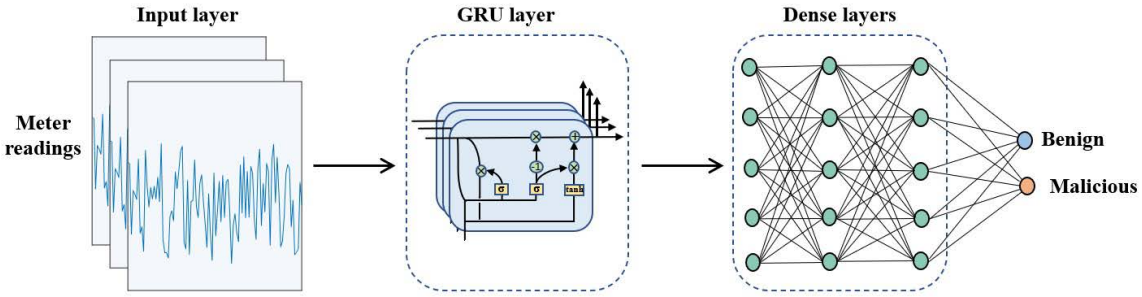

FIGURE 5. The architecture of the GRU-based classifier.

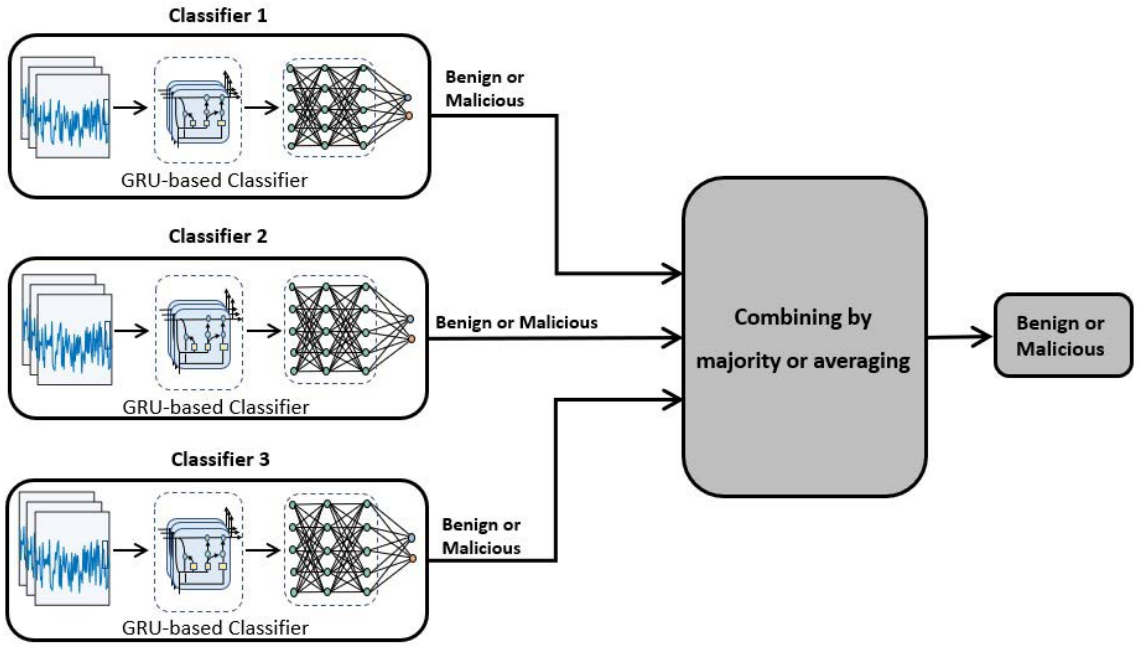

FIGURE 6. The architecture of the proposed ensemble-based detector.

of the model are updated after every iteration using back propagation algorithm, and then the corresponding output parameters of the model are observed to get the best possible results for optimizing the objective function. To minimize the objective function, which is an ultimate goal, the error is fed back into layers to modify the weights. Categorical cross-entropy cost function is widely used in the classification problems to measure the loss between the learned distribution and the true distribution.

1) ACTIVATION FUNCTIONS

In neural networks, each neuron receives inputs from the previous layer's neurons, and then it uses an activation function to compute the output (or activation) of the neuron. It is important to choose good activation functions because they have great influence on the model's convergence speed and accuracy. Among the activation functions used in the literature, the non-linear functions are widely used because they generate complex mappings between the inputs and the outputs [26]. In our experiments, we will use the following activation functions.

- Rectified Linear Unit (ReLU): The ReLU function returns zero if the input is negative, and it returns the same input value if it is positive. Therefore, the ReLU function can be implemented using *max* function as follows.

$$ReLU(x) = max(0, x) \tag{2}$$

- Softmax: For a given input vector $z = [z[1], \ldots, z[N]]$ with N elements, the output of the function is computed as follows.

$$Softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \quad for\ i = \{1, \ldots, N\} \tag{3}$$

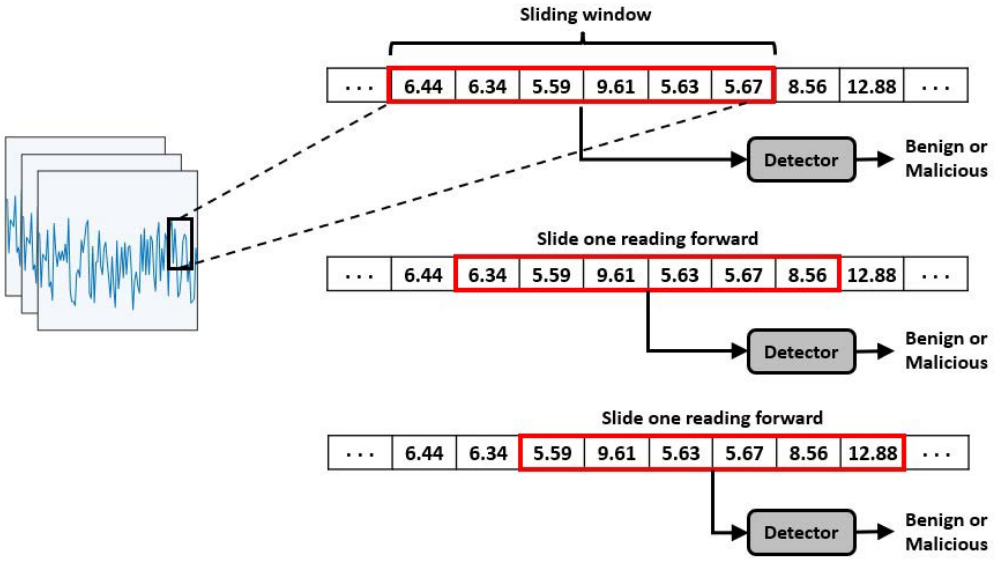

FIGURE 7. A sliding window is used to input a sample containing a number of readings to our detector.

This activation function is usually used in the output layer of the multi-class classifiers, where N is the number of classes.

2) HYPER-PARAMETER OPTIMIZATION

The performance of the machine learning models can be improved by choosing the best possible parameters of the models such as the number of layers, the number of neurons in each layer, and the type of optimizer. The tuning of the hyperparameters of a model is a highly computationally and complicated process which necessaites the use of optimization techniques to find the optimal parameters. The optimization process should calculate the optimum hyperparameters of the model that offer lowest false alarm rate (FA) and highest detection rate (DR). The false alarm rate calculates the proportion of benign samples that are wrongly identified as malicious, while the detection rate calculates the proportion of the accurately detected false readings. So, the objective of this process is to find the best hyper-parameters that can maximize the difference between detection rate and false alarm rate. For our models, we used hyperopt [27] to find the parameters that provide the best results for the models. Next, we briefly describe the deep neural network (DNN) models which will be used in our experiments discussed in Section V.

3) FEEDFORWARD NEURAL NETWORK (FFN)

This network is widely used because it shows high accuracy and can solve complex non-linear problems. It is also called multilayer perceptron (MLP) [28] and comprises three layers, named input, hidden, and output, as shown in Fig. 8. In FFN, the information travels only forward from the input nodes

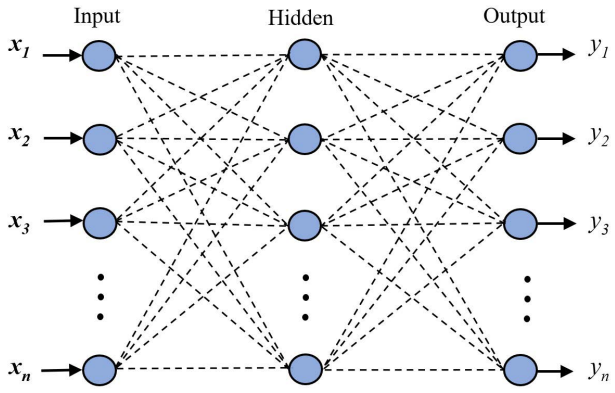

FIGURE 8. Typical FFN's architecture.

to the output nodes passing through the hidden layers. The details of these layers are as follows.

- Input Layer: The first layer receives input data by nodes, called neurons, and passes them to the succeeding layers. In the input layer, the number of neurons is equal to the number of attributes of the input data.
- Hidden Layer: The intermediate layer of an FFN is called hidden layer. The layer has multiple neurons which use the input data and an activation function to compute the output and pass it to the next layer. For simplicity, only one layer is shown in Fig. 8, but in fact, an FFN may have multiple hidden layers and the number of layers is usually an optimization parameter. In this case, every single neuron in a hidden layer is linked with all neurons of the succeeding layer and every link may have a different weight. The best weights that can optimize the model's accuracy are determined during the training process of the model.

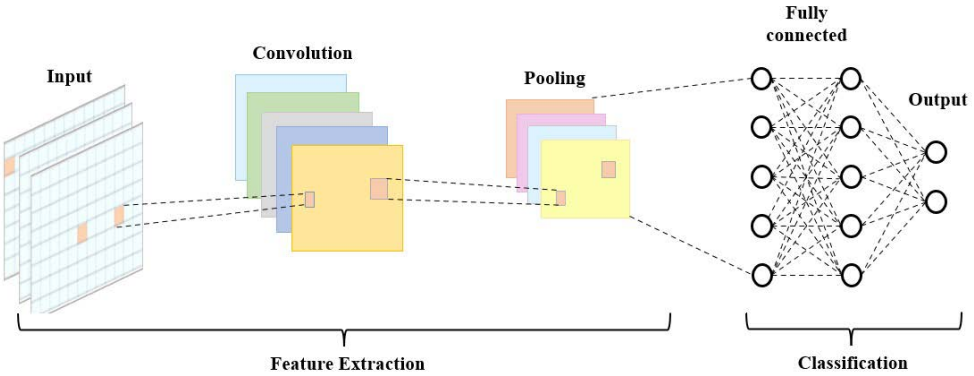

FIGURE 9. Typical CNN architecture.

- Output Layer: This final layer provides the classification of the model in case of multi-class classifiers. In our models, we will use *Softmax* activation function in this layer.

4) CONVOLUTIONAL NEURAL NETWORK (CNN)

CNN is widely used in many applications, such as image, speech, and language processing [29] due to its capability of capturing complex patterns and extracting important features from the input data. The structure of the CNN model comprises various layers namely: input, convolution, pooling, fully connected, and output, as illustrated in Fig. 9. The convolution layer extracts the important features from the input data by sliding small-size filters over the input data, and a non-linear activation function is used to allow the model to make accurate decisions and solve difficult problems. Then, the pooling layer is used to reduce the dimension of the output of the convolution layer while retaining the important information by sub-sampling the feature maps. Finally, a single or multiple fully connected layers and an output layer are used to process the output of the pooling layer and decide the classification, respectively.

5) GATED RECURRENT UNIT NETWORK (GRU)

GRU is a type of recurrent neural networks (RNN). It uses hidden states, called hidden memory, to process and learn variable-length sequences of the inputs. Due to its ability to correlate a sequence of inputs, GRU is widely used in text generation applications to predict the next word of a sentence of words by processing the previous words [30]. An illustration for a GRU is given in Fig. 10. The main component in any RNN is the transition function. For each time step t, the function takes the preceding hidden state H_{t-1} and the current time information X_t to update the current hidden state as below.

$$H_t = F(X_t, H_{t-1}) \tag{4}$$

F represents a non-linear activation function. Similarly, H_{t-1} considered the input at time $t-1$ and the state at time $t-2$ (H_{t-2}), and thus, each state considers the previous states

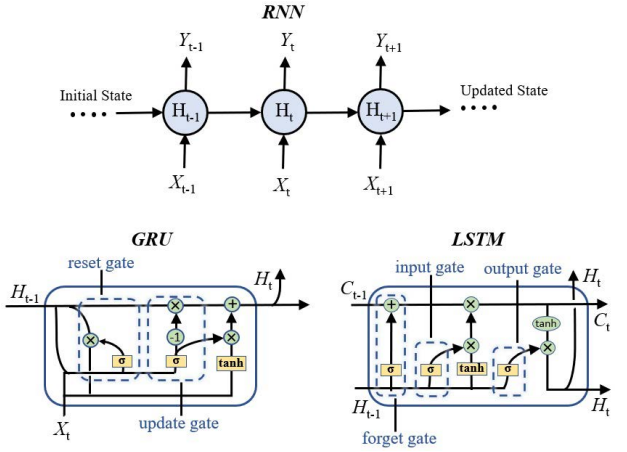

FIGURE 10. Typical architecture of RNN, GRU, and LSTM.

and inputs which enables the GRU to process a sequence of data. To decide the important information that should be kept and the information that should be ignored, GRU uses two gates, called update and reset. Due to its ability to capture the correlations among input data, we will use GRU to train our model.

6) LONG SHORT-TERM MEMORY NETWORK (LSTM)

Like GRU, LSTM is also a type of RNN. RNNs suffer from the problem of vanishing gradients and LSTM is designed to solve this problem. In this problem, the gradient becomes smaller and smaller and the parameter updates become insignificant which means no real learning is done. As illustrated in Fig. 10, LSTM comprises different memory blocks (rectangles) called cells. Memory cells are self-connected and two states, called cell state and hidden state, are transferred to the next cell. The memory block is responsible for memorizing information to support the prediction of the next value in a sequence. The temporal sequences are stored by multiplicative units, called gates. The function of the gates are as follows

- Input gate: The incorporation of the information to the cell state is controlled by the input gate.
- Output gate: The output gate extracts important features from the current cell state and produces the output values.
- Forget gate: This gate decides the useful information that should be kept for memorizing the sequence, and the information that should be removed.

D. ENSEMBLE LEARNING

Ensemble learning is a machine learning approach that uses several weak models, that are usually trained on different data, to create a strong model by combining the outputs of all the models to compute the final output that gives better result. The rationale of the ensemble learning is that every machine learning model may perform differently, i.e., works effectively on some data and less accurately on others, so the use of multiple models trained on different data can negate each other's weaknesses. In our ensemble-based detector, we use a bagging approach for combining the models. The bagging approach usually considers homogeneous weak models that learn independently from each other in parallel and then combines the outputs by averaging or voting to produce the final output.

In this paper, we will investigate an ensemble-based detector where the models used in the detector are similar to the model shown in Fig. 5. Specifically, we trained multiple GRU-based models and the output of each model is either malicious or benign, and then the decisions of all models are combined to produce a final output as shown in Fig. 6. As shown in Fig. 6, we will investigate two techniques to combine the outputs of the models. The first technique computes the final output based on the majority of the votes, i.e., a sample is benign if the majority of the individual models classify it benign; otherwise, it is malicious. The second approach averages the outputs of the individual models to compute the final output, and then determine the class based on the value of the final output.

V. EVALUATIONS

We conduct three experiments to evaluate our detector. The goal of the first experiment is to explore the use of various machine learning models to detect false readings in the case that all the readings of a sample are false. The second experiment, on the other hand, examines the performance of the best performing model in the first experiment trained on different ratios of false readings. Finally, the models trained in the second experiment are used to create our ensemble-based detector and the third experiment will evaluate the performance of this detector.

A. EXPERIMENTAL SETUP

In this paper, all experiments are carried out using one NVIDIA Tesla K80 GPU in a high-performance cluster (HPC) of the Tennessee Tech University. In order to fine-tune the models' hyperparameters, e.g., the number of units per layer, the batch size, learning rate, the activation function of each layer, etc, the hyperopt-tool [27] was used on a validation dataset during the learning phase of the models. *Adam* optimizer [31] is also used to train the models for 150 epochs while using the categorical-cross-entropy as the loss function. In addition, we have also used several Python3 libraries as follow. Our dataset is prepared with Numpy, while for training and evaluating the models, Keras [32] and Scikit-learn [33] are used, respectively. Furthermore, for data visualization, Matplotlib [34] is used.

B. EVALUATION METRICS

The detection of false readings may be regarded as a binary classification problem. The performance of our models are evaluated by the following widely-used metrics, where the true positive (TP) refers to the number of samples that are correctly sorted out as malicious, the true negative (TN) refers to the number of samples correctly categorized as honest, the false positive (FP) refers to the number of honest samples wrongly categorized as malicious, and the false negative (FN) refers to the number of malicious samples wrongly categorized as honest.

- **Accuracy** (ACC). It estimates the proportion of honest/malicious samples which are identified as honest/malicious. The following expression can be used to represent it.

$$ACC(\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100$$

- **Detection rate** (DR). It represents the proportion of malicious samples that are categorized as malicious in comparison to the total number of malicious samples included in the dataset. The following expression can be used to represent it.

$$DR(\%) = \frac{TP}{TP + FN} \times 100$$

- **False Alarm** (FA). It evaluates the proportion of honest samples that are misclassified as malicious compared to the number of honest samples in the dataset. The following expression can be used to represent it.

$$FA(\%) = \frac{FP}{FP + TN} \times 100$$

- **Highest difference** (HD). It measures the difference between the detection rate and the false alarm rate.

$$HD(\%) = DR(\%) - FA(\%)$$

C. RESULTS OF EXPERIMENT 1

In this experiment, we aim to compare the performance of four deep-learning-based detectors when all the readings of the malicious samples are false by applying the attack in Eq. (1) to all the readings of the benign samples. Specifically, the following detectors are trained: FFN, CNN, GRU, and LSTM. We used the dataset discussed in Section III to train and evaluate these detectors, where the detectors are

TABLE 1. Comparison between the performance of different detectors when all the readings are false.

Architecture	Metrics			
	ACC	DR	FA	HD
FFN	86.9	86.5	15.41	71.09
CNN	90	91	11	80
GRU	96.61	97.68	4.45	93.22
LSTM	96.29	96.74	4.17	92.57

trained on both benign and malicious samples and 20% of the dataset's samples are selected randomly for evaluation and 80% of the samples are used for training. We tried different sizes for the sliding window and we found that the size of 40 readings for each sample gives the best results. Therefore, in the three experiments, the size of all the benign and malicious samples are 40 readings, and thus, the input of all the models trained are also 40 readings.

1) RESULTS AND DISCUSSION

Table 1 gives the performance metrics of the different deep learning-based detectors. First, it can be seen from the given results that FFN-based detector achieves the lowest performance in identifying the malicious samples, while both the GRU-based and LSTM-based detectors provide better performance compared to the FFN-based and CNN-based detectors because GRU and LSTM have a good ability to capture the temporal correlation between the consecutive power consumption readings, which can enhance the detector's performance. Since GRU gives the best performance, we will use it in the design of our detector, as will be explained later.

The optimal hyper-parameters of the four detectors and the execution time of each detector are given in Table 2. As can be seen from the table, the GRU and LSTM detectors have longer execution time since they have more complex structure and perform more complicated operations than the FFN and CNN.

D. RESULTS OF EXPERIMENT 2

Unlike *Experiment 1* in which all the readings of the malicious samples are false, in *Experiment 2*, we train four GRU-based detectors on samples with different ratios of false readings. Specifically, the first detector (40M) is trained on detecting the attacks when receiving 40 false readings, i.e., the malicious samples have 40 consecutive false readings. The second detector (20M) is trained on detecting the attacks when receiving 20 false readings, and the third (10M) and fourth (5M) detectors are trained on detecting the attacks when receiving 10 and 5 false readings, respectively. Considering that the size of the sliding window is 40 readings, i.e., each sample has 40 readings, the ratio of the false readings in the malicious samples used to train the 40M, 20M, 10M, and 5M detectors are 100%, 50%, 25%, and 12.5%, respectively.

To train these detectors, we use the same benign dataset discussed in Section III, but for the malicious dataset, we have created new samples by manipulating a number of readings using Eq. (1) based on each detector. We want to detect the false readings as quickly as possible, so detecting the false readings after 5 false readings is better than detecting them after 10 and 20 readings. However, *when the number of false readings in a sample is low, it may be difficult for the detector to differentiate between benign and malicious samples*. Therefore, as will be explained in *Experiment 3*, the detectors trained in *Experiment 2* will be used in an ensemble-based architecture to detect the false readings fast with high accuracy and low false positives. As an example, the optimal hyper-parameters of the 10M GRU-based detector are given in Table 4.

1) RESULTS AND DISCUSSION

To assess the performance of the four detectors, we evaluate them using samples with different number of false readings. Fig. 11 presents the probability of identifying the malicious samples as the number of false readings increase. The given results indicate that although the 5M detector detects the false readings earlier than the other detectors, its detection rate is lower than the other models. Furthermore, Fig. 12 shows the Receiver Operating Characteristics (ROC) curves of the four detectors trained on 5, 10, 20, and 40 false readings. The area under the ROC curve (AUC) is usually used to assess the accuracy of the model's classification, where a higher AUC reflects a better performance. The figure shows that the AUC increases as the detector is trained on malicious samples with a larger ratio of false readings, i.e., 5M detector has the lowest AUC while the 40M detector has the highest AUC. The justification for these results is that as the number of false readings increases in a malicious sample, as it is easier for the detector to identify the sample.

Moreover, Table 3 presents the performance of the four detectors in terms of *ACC*, *DR*, *FA*, and *HD*. It can be seen that *as the model is trained on a higher number of false readings, as the performance is better*, i.e., higher accuracy and detection rate and lower false alarm, but *it takes a longer time to detect the false readings* as shown in Fig. 11. This is because as the number of false readings increases in a sample, as it becomes more different from the benign samples, and thus, the detector can identify the sample more successfully. As can be seen in the table, 40M detector gives much better results than the 5M detector because the malicious samples are closer to the benign samples, but with longer detection time. We can conclude that *there is an obvious tradeoff between the detection time and the accuracy of the model* and to alleviate this tradeoff, in the next experiment, we will use multiple classifiers instead of only one to make the decision.

E. RESULTS OF EXPERIMENT 3

In this subsection, we compare the performance of the ensemble-based false readings detector to the performance of the four detectors discussed in *Experiment 2*, i.e., we want to

TABLE 2. The optimal hyper-parameters of different detector architectures and their execution time.

Architecture	Hyper-parameters			Execution time
	Input layer	Hidden layers	Output layer	
FFN	(40, Linear)	(D,64,Relu),(D,256,Sigmoid),(D,128,Elu), (D,256,Sigmoid),(D,256,Relu),(D,128,Elu)	(2,Softmax)	0.002 ms
CNN	(40, Linear)	(C,64,Relu),(M,2,–),(D,128,Tanh),(D,512,Tanh)	(2,Sigmoid)	0.032 ms
GRU	(40, Linear)	(G,128,Tanh),(D,64,Relu),(D,128,Relu)	(2,Softmax)	0.18 ms
LSTM	(40, Linear)	(L,128,Tanh),(D,512,Tanh),(D,256,Sigmoid)	(2,Softmax)	0.25 ms

Note: in the hidden layer column, each (x,y,z) layer is corresponding to the following. x: type of hidden layer (D: Dense, C: Convolution, G: GRU, and L: LSTM), y: the number of units, and z: the activation function. The consecutive hidden layers are separated by ",".

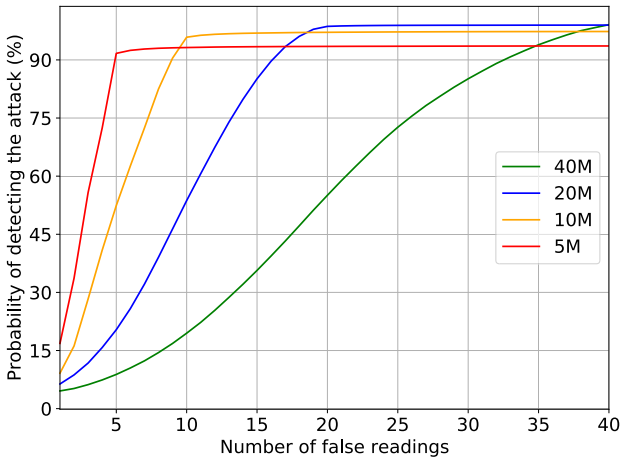

FIGURE 11. The probability of identifying the malicious samples (%) as the number of false readings increases using the four detectors 5M, 10M, 20M, and 40M.

TABLE 3. Comparison between the performance of the 5M, 10M, 20M, and 40M detectors.

Detector	Metrics			
	ACC	DR	FA	HD
40M	96.61	97.68	4.45	93.22
20M	95.8	97.14	5.52	91.62
10M	93.69	93.77	6.39	87.38
5M	89.39	90.08	11.29	78.79

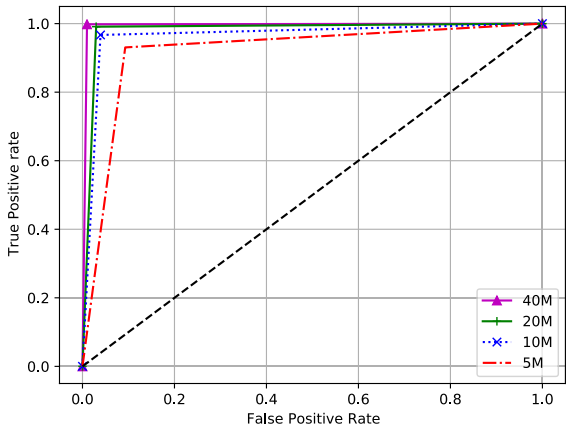

FIGURE 12. The ROC curves of four detectors trained on 5, 10, 20, and 40 false readings.

TABLE 4. The optimal hyper-parameters of the 10M detector.

Architecture	Hyper-parameters		
	Layer	Number of units	AF
GRU	Input	40	Linear
	GRU	128	Relu
	Dense	512	Relu
	Dense	128	Tanh
	Output	2	Softmax

study the impact of the ensemble learning on the performance of the detector. As shown in Fig. 6, our ensemble-based detector consists of three GRU-based classifiers trained on samples with 5, 10, and 20 false readings. We excluded the classifier trained on samples with 40 false readings discussed in *Experiment 2* because as shown in Fig. 11, it is slow in the detection of the false readings and thus using it may also slow the ensemble-based detector. To combine the outputs of the classifiers, we try both majority voting (Maj) and average voting (Avg) techniques. In the first technique, the final decision is based on the majority votes of the three detectors, e.g., the sample is malicious if at least two GRU-based detectors (out of the three detectors) classifies it as malicious; otherwise, it is benign. The second technique averages the output probability of the three GRU-based classifiers to compute the final output, and hence making a decision.

1) RESULTS AND DISCUSSION

As discussed in *Experiment 2*, Fig. 11 shows that 5M classifier can detect the false readings faster than the other classifiers, but Table 3 indicates that it has the highest false alarm. The main conclusion of *Experiment 2* is that there is a trade off between the false alarm rate and the time needed to detect the attack. In *Experiment 3*, we aim to evaluate whether the

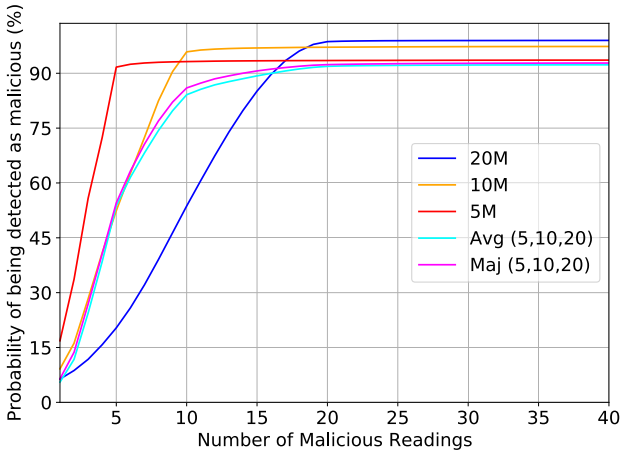

FIGURE 13. The probability of identifying the malicious samples (%) as the number of false readings increases using the the detectors 5M, 10M, and 20M, and the ensemble-based detector.

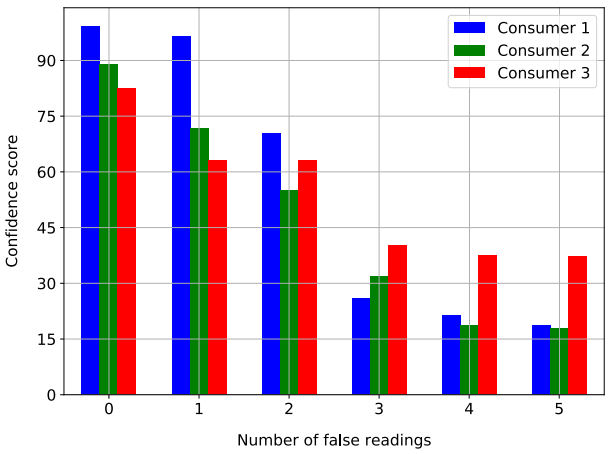

FIGURE 14. The confidence scores as the number of false readings increases in a sample.

TABLE 5. Comparison between the *FA* of the classifiers 5M, 10M, 20M, and 40M and our ensemble-based detector.

Architecture	FA
Classifier 40M	4.45
Classifier 20M	5.52
Classifier 10M	6.39
Classifier 5M	11.29
Ensemble-based detector Avg (5M, 10M, 20M)	3.69
Ensemble-based detector Maj (5M, 10M, 20M)	4.26

ensemble learning can alleviate this tradeoff. Fig. 13 gives the probability of detecting the false readings versus the number of false readings for the individual classifiers (5M, 10M, and 20M) and the ensemble-based classifier using the average and majority techniques. Table 5 gives the *FA* values of the 5M, 10M, 20M, and 40M detectors and the ensemble-based detector.

Based on the results given in Table 5, it can be concluded that the ensemble-based detector offers lower *FA* compared to all the GRU-based detectors, and the averaging technique offers better *FA* than the majority technique. Fig. 13 shows that although the 5M classifier can detect the false readings faster than the ensemble-based detector, its false alarm is 11.29 which is much higher than the false alarm rate of the ensemble-based detector (3.69), as indicated in Table 5. The superiority of the ensemble-based detector is due to the fact that it benefits from the strengths of the classifiers (i.e., the fast detection of 5M and 10M and the low *FA* of 20M classifier) and alleviate their weaknesses (i.e., the slow detection of 20M and the high *FA* of 5M and 10M classifier).

From Fig. 13, it can be seen that the probability of failing to detect a false sample is only 0.04, and therefore, the probability of failing to detect n samples is $(0.04)^n$, and thus even if the attack is not detected after one sample (which has low probability), it will eventually be detected after a number of false samples, i.e., as n increases. Note that the attackers want to reduce their bills so they need to send many false samples.

F. COMPARISON TO THE LITERATURE

As will be explained in details in Section VI, the existing papers in the literature focus on securing the billing, so some detectors such as [3], [14]–[16] are designed to detect electricity theft daily by learning and processing the daily energy consumption, while other detectors such as [11], [17] are designed to detect electricity theft weekly by processing the weekly consumption. Unlike these papers, we focus on securing both billing and energy management application, so we designed our detector to detect false readings fast by using multiple models and methods, such as deep and ensemble learning, training models on different ratios of false readings, hyperparameter optimization, etc. The results of *Experiment 3* indicate that our detector can effectively detect the false readings after sending a few false readings (15 readings) comparing to the daily detection approaches [3], [14]–[16] and the weekly detection approaches [11], [17] that need 144 and 1,008 readings, respectively, to detect the attack.

Moreover, our detector produces a confidence score that indicates how the readings are close to the benign samples, i.e., as the score increases, as there is more confidence that the readings are benign. Since a few false readings need to be sent to identify the malicious samples, to alleviate the impact of these readings, the confidence score can be used in the energy management and load predication to make accurate predictions. The idea is that more weights are given to the readings that have higher confidence scores. In Fig. 14, we give the confidence scores of the readings of three consumers versus the number of false readings. As shown in the figure, the confidence scores decrease as the number of false readings increases.

VI. RELATED WORKS

In this section, we first survey the papers in the literature that investigate the problem of electricity theft using machine learning approaches. Then, we compare our paper to the literature.

Various solutions have been proposed in the literature to detect false-reading attacks in AMI. While some of these solutions use shallow detectors [3], [10], [13], other solutions use deep learning-based detectors [9], [11], [12]. Unlike shallow detectors which need feature extraction techniques to successfully capture the behavior of the input data, the deep learning-based detectors can automatically extract these features through their deep layers. Our detector uses deep learning because it can capture complex patterns and accurately classify false reading attacks comparing to shallow learning techniques like support vector machine, decision tree, and logic regression. Recently, deep learning has been widely used in many applications and it has been proved that deep learning-based detectors outperform the shallow detectors [9], [11], [12], [22], [23].

To identify electricity theft attacks, Jokar *et al.* [3] have presented a detector that processes the energy consumption patterns of the consumers. The paper introduces six attacks and uses them in addition to real benign samples to create malicious samples. Two support vector machine (SVM) detectors are trained, including a single-class SVM that is trained only on benign samples and a multi-class SVM that is trained on both benign and malicious samples. The electricity theft is detected daily, i.e., the detector is designed to process the consumption readings of a complete day. The results indicate that the multi-class SVM exhibits better detection rate and lower false alarm rate compared to the single-class SVM detector.

Shuan Li *et al.* [14] have proposed a model for automatic electricity theft detection with hybrid convolutional neural network and random forest (CNN-RF) model. In this hybrid model, the CNN is used to learn the features of a smart meter's readings using convolution and down-sampling operations, and RF is used to classify the samples. The Irish dataset [35] that has benign SMs' readings collected by electric Ireland and sustainable energy authority of Ireland (SEAI) are used, while the malicious samples are generated by applying different cyber-attack functions to the benign samples. The proposed model is designed to detect the electricity theft by processing the consumption readings of a complete day. The given results indicate that the proposed hybrid model gives better performance compared to SVM, RF, Linear Regression (LR) and Gradient-Boosted Decision Trees (GBDT) models.

Zheng *et al.* [11] have developed a deep-learning technique that comprises both multi-layer perceptron (MLP) and CNN to detect electricity theft. To train the model, the real dataset collected by the state grid cooperation of china (SGCC) is used [36]. The dataset contains both benign and malicious samples where 9% of the consumers are fraudulent. The model is designed to detect electricity theft every week by processing the consumption readings of the week. The experimental results show better performance than RF, SVM, CNN and linear regression.

A combination of a CNN and an LSTM models are used in [17] to detect electricity theft in smart grid AMI. The dataset used to train the detector is obtained from state grid cooperation of China (SGCC) [36]. The paper proposes a new method for pre-processing the dataset to calculate the missing occurrences in the dataset. In order to resolve the class imbalance problem, the synthetic minority sampling method (SMOTE) is used to balance the dataset by creating additional data points. The detector is designed to detect electricity theft every week by processing the consumption readings of a complete week. The experimental results indicate that the proposed model gives 89% of classification accuracy.

In [15], Takiddin *et al.* have investigated a deep-learning model with vector embedding to detect electricity theft attacks. Vector embedding helps in analyzing the relationships and capturing the patterns within fine-grained power consumption readings. To improve the model's performance, a sequential grid-search hyperparameter optimization algorithm is used. The experimental results indicate an improved performance compared to the existing models.

Unlike the above papers that consider the consumption metering system, a multi-data source hybrid deep learning based electricity theft detector is presented by Badr *et al.* [16] to detect false reading attacks for net-metering systems. In the net-metering systems, each house generates energy using renewable resources, e.g., using solar panels on the rooftop. The excess power generated by the house is injected (i.e., sold) to the grid. In the consumption metering system, each smart meter reports the amount of consumption, while in the net-metering system, each smart meter reports the difference between the consumed and generated power. In addition to the smart meters' readings, the proposed model processes the irradiance and temperature data. The proposed model is hybrid and it uses CNN and GRU. The benign samples are the Ausgrid SMs' dataset [37] while the malicious dataset is created by mimicking the behavior of fraudulent consumers by introducing four attacks. The model is designed to detect electricity theft by processing the SMs' readings of one day.

Unlike the above papers that use machine learning for detecting electricity theft, Peng *et al.* [38] have presented an approach that uses a data mining technique. The idea is that a clustering technique is used to cluster the consumers' readings and local outlier factor technique is used to identify the outliers in each cluster as fraudulent consumers. However, comparing to the machine-learning approaches, the performance of data-mining-based solutions is much lower. This is because the machine learning solutions learn the consumption patterns of the consumers and use these patterns to classify the consumers, but data mining solutions can give good results if the consumption patterns of the consumers are same which cannot be ensured practically.

Meters report false readings either for an intentional act, such as compromising the meters by fraudulent consumers, or an unintentional act due to a failure or an inaccuracy by the meters. Similar to most of the papers in the literature [3], [9]–[13], [22], [23], the main focus of the paper is on the detection of reporting false readings intentionally because it is usually harder to detect comparing to the unintentional reporting of false readings. This is because the attackers try to avoid being detected by crafting readings that are difficult to detect by the utility. Moreover, the unintentional false readings can be modeled and our detector can be trained on them to identify them. Also, when a false reading is detected, the electrical utility sends a technician to inspect the meter to know if the false readings are due to the inaccuracy of the meter or due to malicious action (e.g., due to changing the malware of the meter.).

Based on our discussion in this section, the existing solutions in the literature focus only on securing the billing, so they are not designed to detect fraudulent consumers in real time, and thus the SO may use false readings for a period of time in load monitoring and energy management until they are identified. This paper aims to not only secure the billing but also the energy management application by investigating the real detection of false readings. Comparing to the literature that only focuses on developing accurate models, our problem is more challenging because we aim to develop not only accurate model but also fast in the detection of false readings. To do that, as explained earlier, we use a combination of different techniques, models, and approaches.

VII. CONCLUSION AND FUTURE WORKS

In this paper, we focus on securing both the billing and load monitoring and energy management applications. To do that, we have used a combination of approaches, such as deep and ensemble learning, training models on different ratios of false readings, hyper parameter optimization, etc. In addition, our detector produces confidence scores that can be used to make accurate load predictions. We first tried several deep learning models in *Experiment 1*, including CNN, GRU, FFN, and LSTM, and we found that GRU outperforms the other models. Then, in *Experiment 2*, we trained a GRU-based model, consisting of a GRU layer followed by a fully connected neural network, on samples with different ratios of false readings. The GRU is used to capture the correlation between the fine-grained smart meter readings, while the fully connected neural network is used to make accurate decisions. Finally, in *Experiment 3*, we used GRU-based models to create an ensemble-based detector. The results show the the ensemble learning can reduce the *FA* while detecting the false readings fast. Comparing to the literature, our detector can effectively detect the false readings after sending a few false readings (around 15 readings) comparing to the daily detection approaches that need 144 readings and the weekly detection approaches that need 1,008 readings.

In our future work, we will investigate a load forecasting model that takes into account the confidence scores produced by our ensemble-based detector to make accurate predictions even under the existing of false readings.

REFERENCES

[1] B. Chen, J. Wang, X. Lu, C. Chen, and S. Zhao, "Networked microgrids for grid resilience, robustness, and efficiency: A review," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 18–32, Aug. 2021.
[2] X. Zheng, Y. Zeng, M. Zhao, and B. Venkatesh, "Early identification and location of short-circuit fault in grid-connected AC microgrid," *IEEE Trans. Smart Grid*, vol. 12, no. 4, pp. 2869–2878, Mar. 2021.
[3] P. Jokar, N. Arianpoo, and V. C. M. Leung, "Electricity theft detection in AMI using customers' consumption patterns," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 216–226, Jan. 2015.
[4] M. Ali, M. Adnan, M. Tariq, and H. V. Poor, "Load forecasting through estimated parametrized based fuzzy inference system in smart grids," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 1, pp. 156–165, Apr. 2021.
[5] H. Zhang, B. Liu, and H. Wu, "Smart grid cyber-physical attack and defense: A review," *IEEE Access*, vol. 9, pp. 29641–29659, 2021.
[6] PRNewswire. *World Loses $89.3 Billion to Electricity Theft Annually, $58.7 Billion in Emerging Markets*. Accessed: Oct. 8, 2021. [Online]. Available: https://www.prnewswire.com/news-releases/world-loses-893-billion-to-electricity-theft-annually-587-billion-in-emerging-markets-300006515.html
[7] V. B. Krishna, C. A. Gunter, and W. H. Sanders, "Evaluating detectors on optimal attack vectors that enable electricity theft and DER fraud," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 4, pp. 790–805, Aug. 2018.
[8] M. Ismail, M. F. Shaaban, M. Naidu, and E. Serpedin, "Deep learning detection of electricity theft cyber-attacks in renewable distributed generation," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3428–3437, Jul. 2020.
[9] M.-M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gómez-Expósito, "Hybrid deep neural networks for detection of non-technical losses in electricity smart meters," *IEEE Trans. Power Syst.*, vol. 35, no. 2, pp. 1254–1263, 2019.
[10] V. Ford, A. Siraj, and W. Eberle, "Smart grid energy fraud detection using artificial neural networks," *Proc. IEEE Symp. Comput. Intell. Appl. Smart Grid (CIASG)*, Dec. 2014, pp. 1–6.
[11] 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 Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1606–1615, Apr. 2017.
[12] R. R. Bhat, R. D. Trevizan, R. Sengupta, X. Li, and A. Bretas, "Identifying nontechnical power loss via spatial and temporal deep learning," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2016, pp. 272–279.
[13] M. M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gómez-Expósito, "Detection of non-technical losses using smart meter data and supervised learning," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2661–2670, Feb. 2018.
[14] S. Li, Y. Han, X. Yao, S. Yingchen, J. Wang, and Q. Zhao, "Electricity theft detection in power grids with deep learning and random forests," *J. Electr. Comput. Eng.*, vol. 2019, pp. 1–12, Oct. 2019.
[15] A. Takiddin, M. Ismail, M. Nabil, M. M. E. A. Mahmoud, and E. Serpedin, "Detecting electricity theft cyber-attacks in AMI networks using deep vector embeddings," *IEEE Syst. J.*, vol. 15, no. 3, pp. 4189–4198, Sep. 2021.
[16] M. M. Badr, M. I. Ibrahem, M. Mahmoud, M. M. Fouda, F. Alsolami, and W. Alasmary, "Detection of false-reading attacks in smart grid net-metering system," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1386–1401, Jan. 2022.
[17] M. N. Hasan, R. N. Toma, A.-A. Nahid, M. M. M. Islam, and J.-M. Kim, "Electricity theft detection in smart grid systems: A CNN-LSTM based approach," *Energies*, vol. 12, no. 17, p. 3310, 2019.
[18] Kolter. *Residential Energy Disaggregation Dataset (REDD)*. Accessed: 2021. [Online]. Available: http://traces.cs.umass.edu/index.php/Smart/Smart
[19] V. B. Krishna, C. A. Gunter, and W. H. Sanders, "Evaluating detectors on optimal attack vectors that enable electricity theft and DER fraud," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 4, pp. 790–805, Aug. 2018.
[20] M. Zanetti, E. Jamhour, M. Pellenz, M. Penna, V. Zambenedetti, and I. Chueiri, "A tunable fraud detection system for advanced metering infrastructure using short-lived patterns," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 830–840, Jan. 2019.

[21] M. Nabil, M. Ismail, M. Mahmoud, M. Shahin, K. Qaraqe, and E. Serpedin, "Deep recurrent electricity theft detection in AMI networks with random tuning of hyper-parameters," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 740–745.
[22] C. She, C. Sun, Z. Gu, Y. Li, C. Yang, H. V. Poor, and B. Vucetic, "A tutorial on ultrareliable and low-latency communications in 6G: Integrating domain knowledge into deep learning," *Proc. IEEE*, vol. 109, no. 3, pp. 204–246, Mar. 2021.
[23] D. Wu, C. Wang, Y. Wu, Q.-C. Wang, and D.-S. Huang, "Attention deep model with multi-scale deep supervision for person re-identification," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 1, pp. 70–78, Feb. 2021.
[24] V. Kumar, D. R. Recupero, D. Riboni, and R. Helaoui, "Ensembling classical machine learning and deep learning approaches for morbidity identification from clinical notes," *IEEE Access*, vol. 9, pp. 7107–7126, 2021.
[25] H.-Y. Su and C.-R. Huang, "Enhanced wind generation forecast using robust ensemble learning," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 912–915, Jan. 2021.
[26] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," 2018, *arXiv:1811.03378*.
[27] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, "Hyperopt: A Python library for model selection and hyperparameter optimization," *Comput. Sci. Discovery*, vol. 8, no. 1, Jul. 2015, Art. no. 014008.
[28] S. Haykin, *Neural Networks and Learning Machines, 3/E*. London, U.K.: Pearson, 2010.
[29] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, M. Arbib, Ed. Cambridge, MA, USA: MIT Press, 1995, pp. 255–258.
[30] A. F. Ganai and F. Khursheed, "Predicting next word using RNN and LSTM cells: Stastical language modeling," in *Proc. 5th Int. Conf. Image Inf. Process. (ICIIP)*, Nov. 2019, pp. 469–474.
[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015, pp. 1–15.
[32] F. Chollet. (2015). *Keras*. Accessed: Oct. 8, 2021. [Online]. Available: https://github.com/fchollet/keras
[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.
[34] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, May/Jun. 2007.
[35] *Irish Social Science Data Archive*. Accessed: Sep. 2021. [Online]. Available: https://www.ucd.ie/issda/data/ commissionforenergyregulationcer/
[36] *State Grid Corporation of China*. Accessed: Sep. 2021. [Online]. Available: http://www.sgcc.com.cn/
[37] *Ausgrid's Solar Home Electricity Data*. Accessed: Sep. 2020. [Online]. Available: https://www.ausgrid.com.au/Industry/Our-Research/Data-to-share/Solar-home-electricity-data
[38] Y. Peng, Y. Yang, Y. Xu, Y. Xue, R. Song, J. Kang, and H. Zhao, "Electricity theft detection in AMI based on clustering and local outlier factor," *IEEE Access*, vol. 9, pp. 107250–107259, 2021.

MOHAMMED J. ABDULAAL (Member, IEEE) received the B.Eng. degree in mechatronic engineering from The University of Manchester, U.K., in 2012, the M.Sc. degree in mechanical engineering from King Abdullah University for Science and Technology, in 2014, and the Ph.D. degree from the School of Electrical and Electronic Engineering, The University of Manchester, in 2019. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, King Abdulaziz University, Saudi Arabia. His Ph.D. research involved design and implementation of a low-level electroencephalography recognition system and brain–computer interface. His research interests include signal processing and machine learning of biomedical systems, Hajj research, traffic control systems, cybersecurity, and various image processing applications.

MOHAMED I. IBRAHEM received the B.S. and M.S. degrees in electrical engineering (electronics and communications) from Benha University, Cairo, Egypt, in 2014 and 2018, respectively, and the Ph.D. degree in electrical and computer engineering from Tennessee Tech. University, USA, in 2021. He is currently an Assistant Professor with the Department of Cyber Security Engineering, George Mason University, USA. He was an Assistant Lecturer with Benha University. He received the Eminence Award for the Doctor of Philosophy Best Paper from Tennessee Tech University, USA. His research interests include machine learning, cryptography and network security, and privacy-preserving schemes for smart grid communication and AMI networks.

MOHAMED M. E. A. MAHMOUD (Senior Member, IEEE) received the Ph.D. degree from the University of Waterloo, in April 2011. From May 2011 to May 2012, he worked as a Postdoctoral Fellow with the Broadband Communications Research Group, University of Waterloo. From August 2012 to July 2013, he worked as a Visiting Scholar with the University of Waterloo, and a Postdoctoral Fellow with Ryerson University. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Tennessee Tech University, USA. His research interests include security and privacy preserving schemes for smart grid communication networks, mobile *ad-hoc* networks, sensor networks, and delay-tolerant networks. He has received NSERC-PDF Award. He won the Best Paper Award from IEEE International Conference on Communications (ICC'09), Dresden, Germany, in 2009. He is the author for more than 23 papers published in major IEEE conferences and journals, such as INFOCOM Conference and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *Mobile Computing*, and *Parallel and Distributed Systems*. He serves as an Associate Editor for *Peer-to-Peer Networking and Applications* (Springer). He served as a technical program committee member for several IEEE conferences and as a reviewer for several journals and conferences, such as IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and the *Journal of Peer-to-Peer Networking*.

JUNAID KHALID received the B.S. degree in electrical engineering from the University of South Asia, Lahore, Pakistan, in 2015. He is currently pursuing the M.S. degree with King Abdulaziz University, Saudi Arabia. He is also a Graduate Research Assistant with the Department of Electrical and Computer Engineering, King Abdulaziz University. His research interests include sustainable energy systems, optimization of power systems, and smart grids.

ABDULAH JEZA ALJOHANI (Senior Member, IEEE) received the B.Sc. (Eng.) degree in electronics and communication engineering from King Abdulaziz University, in 2006, and the M.Sc. and Ph.D. degrees in wireless communication from the University of Southampton, Southampton, U.K., in 2010 and 2016, respectively. He is currently a Research & Innovation Consultant at Communications and Information Technology Commission (CITC), and an Associate Professor with the Department of Electrical and Computer Engineering, King Abdulaziz University. His research interests include channel coding, cooperative communications, free-space optical communication, and MIMO systems.

AHMAD H. MILYANI received the B.Sc. (Hons.) and M.Sc. degrees in electrical and computer engineering from Purdue University, in 2011 and 2013, respectively, and the Ph.D. degree in electrical engineering from the University of Washington, in 2019. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah, Saudi Arabia. His research interests include power systems operation and optimization, renewable and sustainable energy, power electronics, electric machines, electric vehicles, and the applications of computational intelligence.

ABDULLAH M. ABUSORRAH (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Nottingham, U.K., in 2007. He is currently a Professor with the Department of Electrical and Computer Engineering, King Abdulaziz University, where he is also the Head of the Center for Renewable Energy and Power Systems. His research interests include renewable energy, smart grid, the IoT, and system analysis.