Deep Learning Detection of Electricity Theft Cyber-attacks in Renewable Distributed Generation

Muhammad Ismail, *Senior Member, IEEE*, Mostafa F. Shaaban, *Member, IEEE*, Mahesh Naidu, and Erchin Serpedin, *Fellow, IEEE*

***Abstract*—Unlike the existing research that focuses on detecting electricity theft cyber-attacks in the consumption domain, this paper investigates electricity thefts at the distributed generation (DG) domain. In this attack, malicious customers hack into the smart meters monitoring their renewable-based DG units and manipulate their readings to claim higher supplied energy to the grid and hence falsely overcharge the utility company. Deep machine learning is investigated to detect such a malicious behavior. We aim to answer three main questions in this paper: a) What are the cyber-attack functions that can be applied by malicious customers to the generation data in order to falsely overcharge the utility company? b) What sources of data can be used in order to detect these cyber-attacks by the utility company? c) Which deep machine learning-model should be used in order to detect these cyber-attacks? Our investigation revealed that integrating various data from the DG smart meters, meteorological reports, and SCADA metering points in the training of a deep convolutional-recurrent neural network offers the highest detection rate** (99.3%) **and lowest false alarm** (0.22%)**.**

***Index Terms*—Distributed generation, electricity theft, deep machine learning, hyper-parameter optimization.**

I. Introduction

Electricity theft represents a pressing problem that has brought enormous financial losses to electric utility companies worldwide. In the United States alone, $6 billion worth of electricity is stolen annually [1]. Traditionally, electricity theft is committed in the consumption domain via physical attacks that includes line tapping or meter tampering.

The smart grid paradigm opens the door to new forms of electricity theft attacks [2], [3]. First, electricity theft can be committed in a cyber manner. With the advanced metering infrastructure (AMI), smart meters are installed at the customers' premises and regularly report the customers' consumption for monitoring and billing purposes. In this context, malicious customers can launch cyber-attacks on the smart meters to manipulate the readings in a way that reduces their electricity bill [1]. Second, the smart grid paradigm enables customers to install renewable-based distributed generation (DG) units at their premises to generate energy and sell it back to the grid operator and hence make a profit. This includes photovoltaic (PV) solar cells and wind turbines. In this context, two approaches are adopted when renewable DG units are integrated in the power grid, namely, the net metering system and the feed-in tariffs (FITs) policy. In the net metering system, the excess generation from the DG can be stored as future credit for customers. On the other hand, in the FIT policy, which is referred to as clean energy cashback, customers sell all their generated energy to the grid and get paid in exchange. The incentives offered by the FIT programs are more effective compared with net metering for promoting renewable energy. As such, this program is adopted in several countries worldwide including UK, Canada, Japan, China, Australia, etc. [4]. Hence, FIT requires two meters to be installed in the customer premises, one meter is a selling meter that monitors the energy generated from the DG unit, which is directly injected (sold) to the grid, and the other meter is a buying meter that monitors the consumption. Thus, consumption and generation can be charged independently. In this two-metering system, malicious customers can manipulate the integrity of the reported energy generation data to claim higher supplied energy to the grid and hence falsely overcharge the electric utility company. Such a malicious act is possible due to the weak authentication firmware that is installed in the majority of smart meters deployed worldwide. In specific, a malicious customer gains entry to the firmware via the ANSI optical port of the smart meter using software tools such as Termineter [5] - [7]. While several research works have investigated electricity theft cyber-attacks at the consumption domain, such a research problem is not well investigated in the DG domain and requires a better attention.

Limited research works in literature investigate electricity theft detection in the generation domain [6], [8], while various research works focus on detecting electricity thefts in the consumption domain [1], [9] - [18]. Among existing techniques, machine learning-based methods offer promising detection performance [1], [11] - [18]. Adopting a data-driven approach to detect such malicious attacks in the generation domain is highly motivated as the utility companies are only aware of the DG unit capacity and not necessarily the panel/turbine type and its relevant characteristics [20]. However, it should be highlighted that detecting electricity theft in the DG domain differs from detecting it in the consumption domain as the cyber-attack functions applied at the consumption domain aim to reduce the electricity bill of the malicious customers. On the other hand, the cyber-attack functions applied at the DG units aim to claim a higher supplied energy to the grid.

M. Ismail is with the Department of Computer Science, Tennessee Tech University, Cookeville, TN 38505, USA (email: mismail@tntech.edu).

M. F. Shaaban is with the Department of Electrical and Computer Engineering, University of Waterloo, Canada (e-mail: mostafa.shaaban@uwaterloo.ca).

M. Naidu and E. Serpedin are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA (e-mail: mah12man@tamu.edu, eserpedin@tamu.edu).

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

https://doi.org/10.1109/TSG.2020.2973681

Unfortunately, the existing research work on electricity theft in DG units either do not present such cyber-attack functions [8] or assume that the attacker is aware of the detection mechanism in place to launch its attacks [6]. Hence, novel cyber-attack functions that do not assume the knowledge of the detection mechanism need to be introduced in order to develop a malicious dataset that mimics the theft behavior at the DG's side. Furthermore, the problem in hand offers rich data sources that can be used in order to detect the theft behavior. Various data sources can be used to detect electricity theft at renewable-based DGs including the energy generation profile, meteorological data, and the readings from the supervisory control and data acquisition (SCADA) metering points that monitor various electrical parameters in distribution systems. Further investigations are required to assess the performance improvement in detecting electricity theft in the generation domain when all such data sources are integrated since this is not well studied in existing research, e.g., [6] and [8]. Finally, while time-series-based techniques are adopted for electricity theft detection in the consumption and generation domains, these are usually presented in the context of developing anomaly detectors, e.g., based on least squares with moving time-windows [8] or auto-regressive integrated moving average (ARIMA) [6] models, that are trained only on the benign dataset. On the other hand, when developing classifiers that are trained on both benign and malicious samples to detect electricity theft in the consumption domain, support-vector-machine (SVM) and other shallow classifiers are usually adopted, which does not account for the time series nature of the data. Hence, further investigations are required to develop deep learning-based classifiers that capture the complex patterns and temporal correlation within the generation profile, meteorological data, and SCADA meter readings to yield better detection performance for electricity theft in the generation domain.

In this paper, we aim to answer three main questions relevant to electricity theft detection in renewable-based DG units, namely, a) What are the cyber-attack functions that can be applied to the generation data in order to falsely overcharge the electric utility company? b) What sources of data can be used in order to detect these cyber-attacks by the utility company? c) Which deep machine learning-model should be used in order to detect these cyber-attacks? To address these questions, this paper presents the following contributions:

- We propose a set of cyber-attack functions that manipulate the benign data of the DGs' smart meters in a manner that mimics electricity theft by malicious customers. We focus our attention in this paper on solar energy-based DG units. The extension of the cyber-attack functions and the rest of the analysis to consider other renewable energy sources, e.g., wind energy, is straight forward.
- We investigate the adoption of various data sources to detect electricity theft cyber-attacks in the solar panels. These data sources include in addition to the DG's smart meter data, meteorological (solar irradiance) data, and SCADA metering data. In order to establish a dataset for the benign data, we simulated an IEEE 123-bus test system using practical load and irradiance data for 1 year. Then, the proposed cyber-attack functions are applied on the benign dataset to develop a malicious dataset.
- In order to develop a deep learning-based electricity theft detection system, we have investigated the application of deep feed forward, deep recurrent, and deep convolutional-recurrent neural networks. The detector is trained using benign and malicious datasets. Hyperparameter optimization is applied to define the optimal architecture for the detector. The detector developed herein is a general detector trained using datasets obtained from all the DGs in the system, and hence, the detector can be used to detect the presence of electricity theft cyber-attack for any DG unit in the system. Moreover, we investigate the integration of various data sources (i.e., DG smart meter readings, irradiance data, and SCADA meter readings) to further enhance the detection performance. The proposed detection architecture achieves detection rate of 99.3% and false alarm rate of 0.22%.

The rest of this paper is organized as follows. Section II reviews the related work. The preparation of benign and malicious datasets is discussed in Section III. The detection approach is explained in Section IV. Simulation results and discussions are presented in Section V. Finally, conclusions are drawn in Section VI.

II. Related Work

Limited research work exists on electricity theft detection at the generation domain. Specifically, [8] investigates the detection of electricity theft in PV solar panels by developing an anomaly detector based on the least squares approach and a moving time window. Furthermore, [6] presents a set of optimal cyber-attack functions on the DG units while assuming that the attacker is aware of the detection mechanism. The developed detectors in [6] are based on ARIMA models, Kullback-Leibler divergence (KLD), and principle component analysis (PCA). Most of the relevant works address electricity theft detection in the energy consumption domain.

Data driven solutions have been popular in detecting electricity theft in the consumption domain because of the vast streams of data that are obtained from the smart meters deployed at the customers premises. Many of these works rely on commonly used data-driven machine learning techniques that classify customers based on their load profile into honest and malicious customers. For instance, in [12], a feed forward neural network with single hidden layer is adopted for electricity theft detection using the load profiles of the customers, which reported a classification accuracy up to 70%. An SVM-based classifier is developed in [13] with a fuzzy inference system as a post-processing stage, resulting in a detection accuracy of 72%. In [14], an electricity theft detector is proposed based on an SVM, which results in a detection accuracy of 86.43%. The electricity theft detector in [15] adopts a graph-based approach that implements optimum path forest with a reported detection accuracy of 89%. The work in [16] adopts a two-step approach based on decision trees and SVM to detect electricity thefts, leading to a classification accuracy of 92.5%. The work in [1] relies on an SVM-based classifier and presents a wide range of electricity theft cyber-attacks, which improved the classification accuracy to 94% with 11% false alarm rate, leading to a highest difference of 83%. The aforementioned works utilize shallow machine learning techniques and thus cannot fully capture the various consumption patterns observed in the complex structure of the power metering data. To further enhance the detection accuracy, deep machine learning techniques can be adopted. The work in [17] adopts a deep recurrent neural network (RNN) classifier based on a gated recurrent unit (GRU) that is able to capture the temporal correlation within the customer's load profile, resulting in detection rate of 92.5% and false alarm rate of 5%, improving the highest difference to 87.5%. Furthermore, [18] investigates stealth false data injection (FDI) attacks for electricity theft in the consumption domain, where a stack of restricted Boltzmann machines (RBMs) is implemented in order to detect such malicious FDI attacks, which results in a detection accuracy up to 96%.

III. Data Preparation

This section describes how realistic benign and malicious datasets are developed. Since this data is not publicly available, realistic synthetic data is created. Real load profiles and solar irradiance data are utilized to obtain the benign data, then a set of cyber-attack functions are applied on the benign dataset to obtain the malicious dataset. The benign and malicious datasets will then be used to train the classifier.

A. Benign Dataset

One of the goals of this work is to investigate the integration of different data sources in the training process of the deep learning-based detector. These various data sources include the readings from DG smart meters, meteorological data (solar irradiance), and SCADA metering points. In order to create the benign dataset that incorporates these readings, we simulate the power flow within an IEEE distribution test system. Figure 1 presents the utilized 3-phase IEEE 123-bus test system.

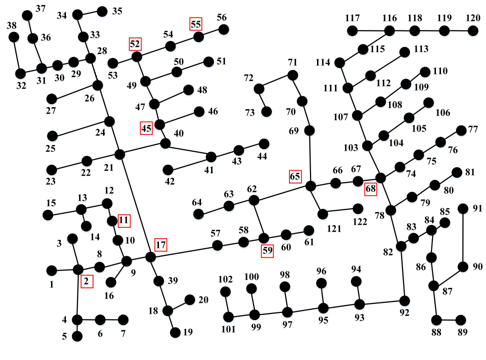

Fig. 1. IEEE 123 bus test system. Highlighted buses in red boxes are monitored with SCADA metering points. The overall 9 allocated SCADA meters provide maximum observability of the system [21].

It should be noted that all the residential customers are considered to be located on buses 52 to 123; on the other hand, the buses from 1 to 51 are dedicated for nonresidential customers, which represents a practical scenario with a mixture of residential and nonresidential units. The first step is to specify the number of residential units, which is determined based on average peak demand of 5 kW in the 3-phase test system. Without loss of generality, only residential customers are considered to have PV panels installed on their roof tops. In order to present a realistic load profile per residential household, real smart meter data from Ontario Canada is utilized [19]. The dataset presents load profiles for customers over the four seasons of the year with a consumption reading reported every 60 minutes. The real load profiles are utilized per residential household such that the aggregate load per phase per bus does not exceed the peak demand of the IEEE 123-bus test system.

To incorporate renewable energy-based DG units within the system, a penetration level of 30% is considered (i.e., 30% of the residential customer peak demand). Historical irradiance data from weather station, located in Ontario Canada, is utilized. The solar irradiance readings (in kW/m^2) are reported at intervals of 60 minutes for 365 days. To specify the number of panels installed per residential unit, an average PV capacity that is randomly selected between 0.5 and 1.5 kW is considered per residential household, without loss of generality. The PV capacity per residential household is divided by the panel capacity to determine the number of installed panels per household. To simulate a realistic environment, 5 different types of PV panels are considered, without loss of generality. Each residential unit that is considered to install solar PV panels is randomly assigned one panel type. Table I summarizes the characteristic parameters of each panel type [4]. The PV panel parameters in Table I are under standard test conditions (25^o C) and are defined as follows: V^{MPP} and I^{MPP} are voltage and current at the maximum power point, respectively, I^{SC} and V^{OC} are the PV panel short circuit current and open circuit voltage at operation conditions, respectively, T_{NOC} stands for the nominal cell operating temperature, which presents the temperature reached by solar cells under nominal conditions of 20^o C and 0.8 kW/m^2 irradiance, K^{v} and K^{i} are the voltage and current temperature coefficients, respectively, and the PV capacity C_{PV} is the maximum power generated by the PV panel. Given the panel-related characteristics and the solar irradiance values, the corresponding solar energy generation profile for each panel type, and hence for each residential customer, can be determined. Define the following terms at a specific day $d \in \mathcal{D}$ and specific hour $t \in \mathcal{T}$ for a given panel type k: T^{cell} is the cell temperature of the PV panel, T^{A} is the ambient temperature, S^{IR} is the solar irradiance, and FF is the fill factor of the PV panel. Hence, the generated power $P^{\text{PV}}_{k,t,d}$ can be calculated as follows [4]:

$$
\begin{aligned}
T^{\text{cell}}_{k,t,d} &= T^{\text{A}}_{t,d} + S^{\text{IR}}_{t,d} \times \frac{T_{\text{NOC},k} - 20}{0.8}, \\
I^{\text{SC}}_{k,t,d} &= S^{\text{IR}}_{t,d} \times (I^{\text{SC}}_k (1 + K^{\text{i}}_k (T^{\text{cell}}_{k,t,d} - 25)/100)), \\
V^{\text{OC}}_{k,t,d} &= V^{\text{OC}}_k (1 + K^{\text{v}}_k (T^{\text{cell}}_{k,t,d} - 25)/100), \\
FF_k &= \frac{V^{\text{MPP}}_k \times I^{\text{MPP}}_k}{V^{\text{OC}}_k \times I^{\text{SC}}_k}, \\
P^{\text{PV}}_{k,t,d} &= FF_k \times V^{\text{OC}}_{k,t,d} \times I^{\text{SC}}_{k,t,d}.
\end{aligned}
\tag{1}
$$

TABLE I
CHARACTERISTIC PARAMETERS OF SOLAR PV PANELS

Panel Type	1	2	3	4	5
V^{MPP} (V)	72.9	30.2	49.2	40.2	47
I^{MPP} (A)	5.97	8.11	1.78	6	2.88
V^{OC} (V)	85.6	37.8	61	50.7	61.3
I^{SC} (A)	6.43	8.63	1.98	6.7	3.41
K^{v} (% ^{o}K)	-0.0027	-0.0033	-0.0027	-0.003	-0.003
K^{i} (% ^{o}K)	0.05	0.06	0.04	0	0.07
T_{NOC} (^{o}C)	45	46	45	47	45
C_{PV} (kW)	0.435	0.245	0.0875	0.23	0.135

Given the load and generation profiles for each bus, the IEEE 123-bus test system is simulated to specify the power flows and voltages, which present the readings provided by the SCADA metering points. The objective here is to capture the relationship between the SCADA meter readings and the PV energy generation profile, which will be used later for theft detection. The SCADA readings in the form of voltage, current, and power are affected by the injection from the PV units installed in the downstream. Denote the total number of PV panels of type k installed by the customers on bus i and phase p as $N^{\mathrm{PV}}_{k,i,p}$. The aggregate generated power $P^{\mathrm{PV}}_{i,p,t,d}$ on bus i and phase p at time t and day d is given by

$$P^{\mathrm{PV}}_{i,p,t,d} = \sum_{k} \frac{P^{\mathrm{PV}}_{k,t,d} \times N^{\mathrm{PV}}_{k,i,p}}{S_{\mathrm{base}}}, \tag{2}$$

where S_{base} is the base power for the system in kVA and $P^{\mathrm{PV}}_{k,t,d}$ is given by (1). The non-linear power flow equations are described as follows

$$\begin{aligned}
&P^{\mathrm{G}}_{i,p_1,t,d} + P^{\mathrm{PV}}_{i,p_1,t,d} - P^{\mathrm{L}}_{i,p_1,t,d} = \\
&\sum_{j}\sum_{p_2}\{V_{i,p_1,t,d}V_{i,p_2,t,d}Y_{i,j,p_1,p_2}\times \\
&\cos(\theta_{i,j,p_1,p_2} + \delta_{i,p_2,t,d} - \delta_{i,p_1,t,d}) - V_{i,p_1,t,d}V_{j,p_2,t,d}\times \\
&Y_{i,j,p_1,p_2}\cos(\theta_{i,j,p_1,p_2} + \delta_{i,p_2,t,d} - \delta_{i,p_1,t,d})\} \quad \forall\ i, p_1, t, d, \\
&Q^{\mathrm{G}}_{i,p_1,t,d} - Q^{\mathrm{L}}_{i,p_1,t,d} = \sum_{j}\sum_{p_2}\{V_{i,p_1,t,d}V_{i,p_2,t,d}Y_{i,j,p_1,p_2}\times \\
&\sin(\theta_{i,j,p_1,p_2} + \delta_{i,p_2,t,d} - \delta_{i,p_1,t,d}) - V_{i,p_1,t,d}V_{j,p_2,t,d}\times \\
&Y_{i,j,p_1,p_2}\sin(\theta_{i,j,p_1,p_2} + \delta_{i,p_2,t,d} - \delta_{i,p_1,t,d})\} \quad \forall\ i, p_1, t, d,
\end{aligned} \tag{3}$$

where i and j are bus indices, p_1 and p_2 are phase indices, $P^{\mathrm{G}}_{i,p_1,t,d}$ and $Q^{\mathrm{G}}_{i,p_1,t,d}$ are the generated active and reactive power in per unit at bus i for phase p_1 in time t at day d, respectively, $P^{\mathrm{PV}}_{i,p_1,t,d}$ is the generated PV active power, $P^{\mathrm{L}}_{i,p_1,t,d}$ and $Q^{\mathrm{L}}_{i,p_1,t,d}$ are the load active and reactive power in per unit, respectively, $V_{i,p_1,t,d}$ and $\delta_{i,p_1,t,d}$ are the magnitude in per unit and angle of the voltage at bus i and phase p_1, respectively, and Y_{i,j,p_1,p_2} and θ_{i,j,p_1,p_2} are the magnitude in per unit and angle of the admittance element in the branch admittance matrix, respectively.

In a practical setting, not every bus in the power grid is monitored by a SCADA metering point. Only a subset of buses is monitored. The number and location of the SCADA metering points are determined with the objective of achieving maximum observability for the entire system. By adopting the SCADA metering optimal allocation approach described in [21], the number of SCADA metering points that achieves maximum observability is found to be 9 meters whose locations are highlighted in Figure 1.

From the discussion above, the following matrices present the benign dataset: the solar energy generation profile $\mathbf{E}$ as described by (1), the solar irradiance profile $\mathbf{R}$ as given by $S^{\mathrm{IR}}_{t,d}$, and the SCADA meter power flow readings for the selected 9 buses $\mathbf{P}$ as given by (3). Each of these matrices presents a reading every 60 minutes over 365 days. For each matrix, the rows present a full day sample and the columns present a time-instant (60 minutes-separated) within the day.

B. Malicious dataset

One of the challenges that face this research work is the absence of data that is needed to develop the desired classifier. In the previous subsection, we have implemented a realistic simulation environment to create a benign dataset that represents various data sources. In this subsection, a set of cyber-attack functions will be applied on the benign dataset in order to create the malicious dataset. The cyber-attack functions manipulate the benign data in a way that mimics the malicious customer behavior. As the malicious customer does not have access to the solar irradiance data and the SCADA metering data, the cyber-attack functions are applied only on the solar energy generation profile. The customer has access to the smart meter attached to the solar panel, which is not the case for the weather station that reports the solar irradiance data and the SCADA metering points monitoring the buses.

The objective of the cyber-attack functions that manipulate the reported energy generation profile is to claim higher injected energy to the power grid. We introduce the four cyber-attack functions listed in Table II. The first cyber-attack function $f_1(E_{t,d})$ implements a partial increment attack, where a malicious customer reports an incremental percentage $(1+\alpha)$ of the actual generated energy $E_{t,d}$ (e.g., reporting 120% of the actual generation). The second function $f_2(E_{t,d})$ presents also a partial increment attack, however, the incremental percentage changes from time instant to another and from day to day $(1 + \alpha_{t,d})$. The third attack function $f_3(E_{t,d})$ represents a minimum generation attack, where a malicious customer sets a minimum reporting value $(\beta_{t,d})$ for the generated energy (for instance, $\beta_{t,d} = 20\%$ of the peak generation is reported whenever the actual generated energy equals zero). The fourth cyber-attack function $f_4(E_{t,d})$ is a peak generation attack, where a malicious customer reports only the highest energy generation value once reached. It should be highlighted that the aforementioned attack functions are generic regardless of the type of renewable energy source.

Each cyber-attack function is applied on the solar energy generation profile matrix $\mathbf{E}$, which results in four malicious matrices. The concatenation of the benign and malicious solar energy generation profiles presents the entire dataset $\hat{\mathbf{X}}$ where each row gives a sample energy generation profile over the day. Each sample is associated with a label that equals '0' if the sample is benign and equals '1' if the sample is malicious. The label column vector associated with $\hat{\mathbf{X}}$ is denoted by $\hat{\mathbf{Y}}$. As we have four times malicious data than the benign one, the trained classifier will be a biased one. To avoid this, the minor (benign) class is over-sampled using the adaptive synthetic sampling approach (ADASYN) [22]. The balanced dataset is then normalized in order to bring the values of all the features to a common scale. The normalized dataset $\mathbf{X}$ presents a zero mean and unit variance and is associated with the labeling vector $\mathbf{Y}$. The data is then split into two disjoint sets with ratio 2:1, namely train data $\mathbf{X}_{\text{TR}}$ with label $\mathbf{Y}_{\text{TR}}$ and test data $\mathbf{X}_{\text{TST}}$ with label $\mathbf{Y}_{\text{TST}}$.

TABLE II
PROPOSED CYBER-ATTACK FUNCTIONS FOR ELECTRICITY THEFT ON RENEWABLE-BASED DG UNITS.

Attack Type	Mathematical Representation
Partial Increment Attack	$f_1(E_{t,d}) = (1+\alpha)E_{t,d}$
Partial Increment Attack	$f_2(E_{t,d}) = (1+\alpha_{t,d})E_{t,d}$
Minimum Generation Attack	$f_3(E_{t,d}) = \beta_{t,d} + E_{t,d}$
Peak Generation Attack	$f_4(E_{t,d}) = \max(E_{t,d}, E_{t-1,d})$

IV. Design of Electricity Theft Detector

In this section, we aim to develop a classifier that can detect cyber-attacks targeting the integrity of the readings about the amount of generated energy. The detector design is based on deep neural networks that can capture complex representative patterns within the data. Three structures are investigated for the detector, namely, deep feed forward, deep recurrent, and deep convolutional-recurrent neural networks.

A. Training Stage

1) Deep Feed Forward Neural Network-based Detector: The deep feed forward neural network presents the simplest implementation of the detector and offers the lowest computational complexity. It consists of an input layer, a set of hidden layers, and an output layer. Using $\mathbf{X}$, the input layer consists of 24 neurons that are fed with the readings of the generated energy over the day, i.e., $x_d \in \mathbf{X}$. The hidden layers present L layers each with N neurons. The output layer has 1 neuron to represent the two classes, i.e., benign sample $y =$ '0' or malicious sample $y =$ '1'.

The weight matrix W^l defines the weight $w^l_{nn'}$ of the connection from neuron n' in layer $l-1$ to neuron n in layer l. The bias vector of layer l, b^l, defines the bias b^l_n of neuron n in layer l. Let $z^l_n = \sum_{n'} w^l_{nn'} a^{l-1}_{n'} + b^l_n$ denote the weighted sum of inputs to neuron n, where $a^l = \sigma(z^l)$ and $\sigma(\cdot)$ is an activation function. The training of the detector aims to find the model parameters W^l and b^l denoted by Θ, which is achieved by minimizing the cross-entropy

$$\min_{\Theta} C = \frac{-1}{|\mathbf{X}_{\text{TR}}|} \sum_{\mathbf{X}_{\text{TR}}} \{ y(x_d)\ln(a^L_n) + (1-y(x_d))\ln(1-a^L_n)\}, \tag{4}$$

where $|\mathbf{X}_{\text{TR}}|$ denotes the number of training samples and $y(x_d)$ denotes the label corresponding to sample x_d. Iterative gradient descent is used to solve the minimization in (4). Let I denote the number of iterations. The entire training set is divided into equal-sized M mini-batches. Algorithm 1 describes the training stage of the feed forward neural network-based detector assuming a stochastic gradient descent (SGD) optimization. In Algorithm 1, two stages are implemented in each iteration. The feed forward stage determines the predicted output vectors. The back propagation stage then determines the gradient of the cross-entropy of (4) as a function of an error term Δ [23]. The gradient then is used to update the weights and bias values in each iteration. The following symbols are used in Algorithm 1: $\bigtriangledown_a$ represents partial derivative with respect to a, $\sigma'(z^l(x))$ denotes the reciprocal of the partial derivative of z^l with respect to a^l, $\odot$ is the Hadamard product, and T represents the transpose operation.

Algorithm 1: Deep Feed Forward-based Detector Training

```
Initialization: Weights $W^l$ and biases $b^l$ for all $l$,
  $i = 1$
  while $i \neq I$ do
    Initialize: $m = 1$
      while $m \neq M$ do
        for each training example $x_d$ in mini-batch $m$
          do
            Feed forward:
              Compute: $z^l(x) = w^l a^{l-1}(x) + b^l$ and
              $a^l(x) = \sigma(z^l(x))\ \forall l = 2, \ldots, L$
            Back propagation:
              Compute: $\Delta^L(x) = \bigtriangledown_a C(x) \odot \sigma'(z^L(x))$
              and
              $\Delta^l(x) = ((w^{l+1})^{\text{T}} \Delta^{l+1}(x)) \odot \sigma'(z^l(x))$
              $\forall l = L-1, \ldots, 2$
        end for
        Weight and bias update:
          $w^l = w^l - \frac{\eta}{K}\sum_x \Delta^l(x)(a^{l-1}(x))^{\text{T}}$ and
          $b^l = b^l - \frac{\eta}{K}\sum_x \Delta^l(x)$
    end while
end while
Output: Optimal parameters $W^l$ and $b^l$ for all layers
```

2) Deep Recurrent Neural Network-based Detector: Despite offering lower computational complexity, the deep feed forward neural network-based detector does not exploit the temporal correlation present in the input data. The energy generation profile represents a time-series data that is best handled using a recurrent neural network (RNN)-based classifier, which can further enhance the detection performance. To overcome the vanishing gradient problem while learning temporal correlation over long intervals, a variant of the RNN, namely, a gated recurrent unit (GRU)-based RNN, is utilized [23]. The input layer of the GRU-based classifier consists of 24 neurons that are fed with the daily energy generation profile $x_d \in \mathbf{X}$. The input layer is followed by L hidden GRU layers, and each layer presents N neurons (units). Except for the last GRU layer, each layer accepts and produces a sequence vector. The output layer presents 1 neuron: $y =$ '0' and $y =$ '1' for a benign and a malicious sample, respectively.

Each layer l presents an output vector o^l with $o^1 = x_d$. A hidden GRU layer l defines the following parameters:

- Input at time step t: This is denoted by o^{l-1}_t and results from the previous layer $l-1$.

Algorithm 2: Deep GRU RNN-based Detector Training

```
Initialization: Weights $U^l_{(\cdot)}$, $W^l_{(\cdot)}$, $V^l_{(\cdot)}$, and bias $b^l_{(\cdot)}$ $\forall l$, $i = 1$
while $i \neq I$ do
    Initialize: $m = 1$
    while $m \neq M$ do
        for each training example $x_d$ in mini-batch $m$ do
            Feed forward:
            for each recurrent layer $l$ do
                for each time step $t$ do
                    $z^l_t = \sigma(o^{l-1}_t U^l_z + s^l_{t-1} W^l_z + b^l_z)$
                    $r^l_t = \sigma(o^{l-1}_t U^l_r + s^l_{t-1} W^l_r + b^l_r)$
                    $h^l_t = \tanh(o^{l-1}_t U^l_h + (s^l_{t-1} \odot r^l) W^l_h + b^l_h)$
                    $s^l_t = (1 - z^l) \odot h^l + z^l \odot s^l_{t-1}$
                    $o^l_{t+1} = (V^l_o s^l_t + b^l_o)$
                end for
            end for
            Back propagation:
            Compute: $\bigtriangledown_{U^l_{(\cdot)}} C(x_d)$, $\bigtriangledown_{V^l_{(\cdot)}} C(x_d)$, $\bigtriangledown_{W^l_{(\cdot)}} C(x_d)$, and $\bigtriangledown_{b^l_{(\cdot)}} C(x_d)$
        end for
        Weight and bias update:
        $U^l_{(\cdot)} = U^l_{(\cdot)} - \frac{\eta}{K} \sum_x \bigtriangledown_{U^l_{(\cdot)}} C(x_d)$
        $V^l_{(\cdot)} = V^l_{(\cdot)} - \frac{\eta}{K} \sum_x \bigtriangledown_{V^l_{(\cdot)}} C(x_d)$
        $W^l_{(\cdot)} = W^l_{(\cdot)} - \frac{\eta}{K} \sum_x \bigtriangledown_{W^l_{(\cdot)}} C(x_d)$
        $b^l_{(\cdot)} = b^l_{(\cdot)} - \frac{\eta}{K} \sum_x \bigtriangledown_{b^l_{(\cdot)}} C(x_d)$
    end while
end while
Output: Optimal $U^l_{(\cdot)}$, $W^l_{(\cdot)}$, $V^l_{(\cdot)}$, and $b^l_{(\cdot)}$ $\forall l$.
```

- Hidden state at time step t: This is denoted by s^l_t and it represents the memory that is computed using the hidden state s^l_{t-1} of the same layer at the previous time step.
- Update gate at time step t: This is denoted by z^l_t and represents a combination of the new input o^{l-1}_t and the previous memory s^l_{t-1}, given by $z^l_t = \sigma(o^{l-1}_t U^l_z + s^l_{t-1} W^l_z + b^l_z)$, U^l_z and W^l_z are learnable weight matrices, b^l_z is a bias vector, and $\sigma(\cdot)$ is an activation function.
- Reset gate at time t: This is denoted by r^l_t and it specifies the contribution of the memory s^l_{t-1} to the next state h^l_t. We have $r^l_t = \sigma(o^{l-1}_t U^l_r + s^l_{t-1} W^l_r + b^l_r)$ and $h^l_t = \tanh(o^{l-1}_t U^l_h + (s^l_{t-1} \odot r^l) W^l_h + b^l_h)$, U^l_r, W^l_r, U^l_h, and W^l_h are weight matrices and b^l_r and b^l_h are bias vectors.

The next state is then calculated as $s^l_{t+1} = (1 - z^l) \odot h^l + z^l \odot s^l_t$ and the output is given by $o^l_{t+1} = (V^l_o s^l_{t+1} + b^l_o)$, where V^l_o and b^l_o are learnable weight matrix and bias, respectively. The objective of the detector's training stage is to learn the parameters $U^l_{(\cdot)}$, $W^l_{(\cdot)}$, $V^l_{(\cdot)}$, and $b^l_{(\cdot)}$ that lead to the desirable output $y(x_d)$ for input x_d. This is achieved by minimizing the cross-entropy function presented in (4). The solution of such a minimization follows a similar approach as described for the feed forward neural network, however, the back propagation here is essentially a back propagation through time (BPTT). The training process is described in Algorithm 2.

3) Deep Convolutional-Recurrent Neural Network-based Detector: The RNN (GRU)-based architecture captures the temporal correlation within the data. However, to further enhance the detection performance, the RNN-model can be fed with more relevant features rather than just raw data as described in the previous subsection. Towards this objective, a hybrid convolutionl neural network (CNN) and RNN is tested for the detector's architecture. This structure consists of 1-dimensional (1-D) convolutional and max-pooling layers followed by a stack of layers of GRUs. The convolutional layer consists of number of learnable filters that are used to extract features from the input. Such filters are small in size and extract features with strong dependency patterns. The max-pooling layer then compresses the output from the convolutional layer in order to retain the most relevant features. In that sense, the 1-D convolutional and max-pooling layers act as a trainable feature extractor and the GRU layers further learn the temporal relationship within the extracted features. The architecture is know as C-RNN.

The input layer applies 1-D convolution on the energy generation profile $x_d \in \mathbf{X}$ that presents 24 readings per sample day. The output of this convolutional layer is $x_c = x_d F_c + b_c$, where F_c and b_c are learnable square filter matrix and bias, respectively. This output is then fed to the max-pooling layer, resulting in $x_{\text{mp}} = \text{maxpool}(x_c)$, which is then fed to the GRU layer. Hence, the input to the first GRU layer $o^1 = x_{\text{mp}}$. Then, L hidden GRU layers, each consists of N neurons (units), are introduced followed by an output layer that consists of 1 neuron representing benign $y =$ '0' or malicious $y =$ '1'. Figure 2 illustrates the detector's architecture.

The objective of the detector's training stage is to learn the parameters F_c and b_c of the convolutional layer along with the parameters of the GRU layers described in the previous subsection, namely, $U^l_{(\cdot)}$, $W^l_{(\cdot)}$, $V^l_{(\cdot)}$, and $b^l_{(\cdot)}$, that lead to the desirable output $y(x_d)$ for input x_d. This is achieved by minimizing the cross-entropy function presented in (4), which can be done following an Algorithm similar to Algorithm 2 that also incorporates the convolutional layer ($x_c = x_d F_c + b_c$ and $x_{\text{mp}} = \text{maxpool}(x_c)$) in the feed forward stage right before the For-loop of the RNN.

4) Performance Evaluation of the Detection Performance: Let TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. TP means that the sample is malicious and it is detected as malicious, TN means that the sample is benign and detected as benign, FP means that the sample is benign and detected as malicious, while FN means that the sample is malicious and detected as benign. The following metrics are defined to evaluate the performance of the detector: (a) The detection rate (DR), which is also referred to as recall or sensitivity, measures the fraction of correctly detected malicious samples. Hence, DR = TP/(TP + FN); b) The false alarm (FA) measures the fraction of benign samples that are falsely identified as being malicious. Thus, FA = FP/(FP + TN); c) The highest difference HD = DR – FA; d) The precision (PR) measures the fraction of correctly detected malicious samples from all samples. Hence, PR=

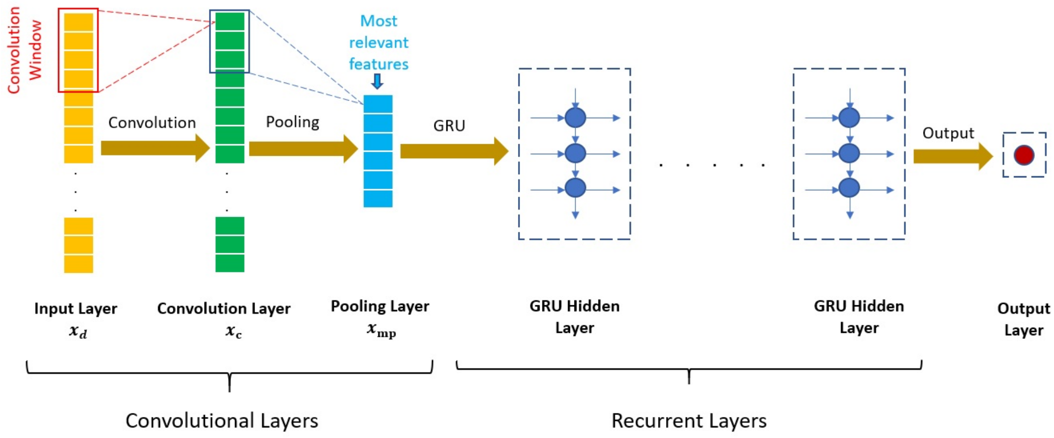

Fig. 2. Illustration of the C-RNN-based detector's architecture.

TP/(TP+FP); e) F1 score, F1 = (2 × PR × DR)/(PR + DR); f) Receiver operating characteristic (ROC) curve, which plots the TP versus the FP and hence the detection performance is demonstrated by the area under the curve (AUC).

B. Optimization of Hyper-parameters

The previous algorithms specify the parameters of the detector in terms of the weight matrices and bias values. Other hyper-parameters remain to be specified such as the number of hidden layers L, number of neurons within each hidden layer N, the type of optimizer used in determining the parameters O, and the activation functions A_{H} and A_{O}, for hidden and output layers, respectively. Optimal choice of such hyper-parameters can significantly improve the detection performance. To avoid the computational complexity of an exhaustive grid search, in this paper, a random grid search is used instead to reach sub-optimal hyper-parameters efficiently. Let $\mathcal{L}$, $\mathcal{N}$, $\mathcal{O}$, and $\mathcal{A}$ denote uniform distributions of number of hidden layers, number of neurons, optimizer, and activation functions. Algorithm 3 presents random search of hyper-parameters using the uniformly distributed sets over the training data $\mathbf{X}_{\mathrm{TR}}$ using K-fold cross validation.

Algorithm 3: Random Search-based Hyper-parameter Optimization

```
Initialization: Weight and bias values, $i = 1$
while $i \neq I$ do
    $L[i] \leftarrow \mathcal{L}, N[i] \leftarrow \mathcal{N}, O[i] \leftarrow \mathcal{O}, A[i] \leftarrow \mathcal{A}$
    for each $\hat{\mathbf{X}}_{\mathrm{TR}}$ and $\hat{\mathbf{X}}_{\mathrm{TST}}$ in K-folds($\mathbf{X}_{\mathrm{TR}}$) do
        Solve Algorithm 1 for deep feed forward-based
          detector or Algorithm 2 for deep
          GRU-RNN-based detector or the modified
          Algorithm 2 for deep C-RNN using the
          sampled hyper-parameters and record the
          DR[$i$] and FA[$i$].
    end for
    Record the average DR and FA values across all
      folds.
end while
Output: Optimal hyper-parameter values that yield
  highest DR and FA performance.
```

C. Integration of Multiple Data Sources in the Training

The detection models discussed in the previous subsections are trained using only the smart meter data reporting the PV generation profile per household. To further enhance the detection performance, various data sources can be integrated to train a more efficient detector. This is especially true when the integrated data sources are beyond the access of malicious users, and hence, the samples from such data sources are always benign. The data sources considered in this paper include meteorological (solar irradiance) data that are made available from weather stations, and SCADA metering points monitoring the buses within the power grid. Hence, the trained detector will learn the correlation between the reported PV generation profile samples and the solar irradiance and SCADA metering data. For instance, it is naturally expected that the PV generation profile of solar PV panels follows the solar irradiance profile. Hence, there should exist a relationship between these two temporal sequences that can be exploited to better inform the theft decision. Similarly, SCADA meter readings can be utilized to enhance the detection performance by capturing the temporal correlation between the PV generation profile and the net power flow within the grid.

To reuse the detection model developed in previous subsections using single data source (and to determine the percentage performance improvement), the embedding obtained from such a model is merged (concatenated) with the solar irradiance data ($\mathbf{R}$). A set of dense layers are then stacked to capture the relationship between the embedding of the PV generation profile and the solar irradiance data, and hence, classify the input PV generation profile as benign or malicious sample. To include the SCADA meter readings, principle component analysis (PCA) is first applied on the readings from the 9 meters ($\mathbf{P}$) in order to capture the most relevant features in a one-column vector ($\hat{\mathbf{P}}$). Then, the embedding obtained from the PV generation profile is merged (concatenated) with irradiance data ($\mathbf{R}$) and SCADA meter reading ($\hat{\mathbf{P}}$) followed

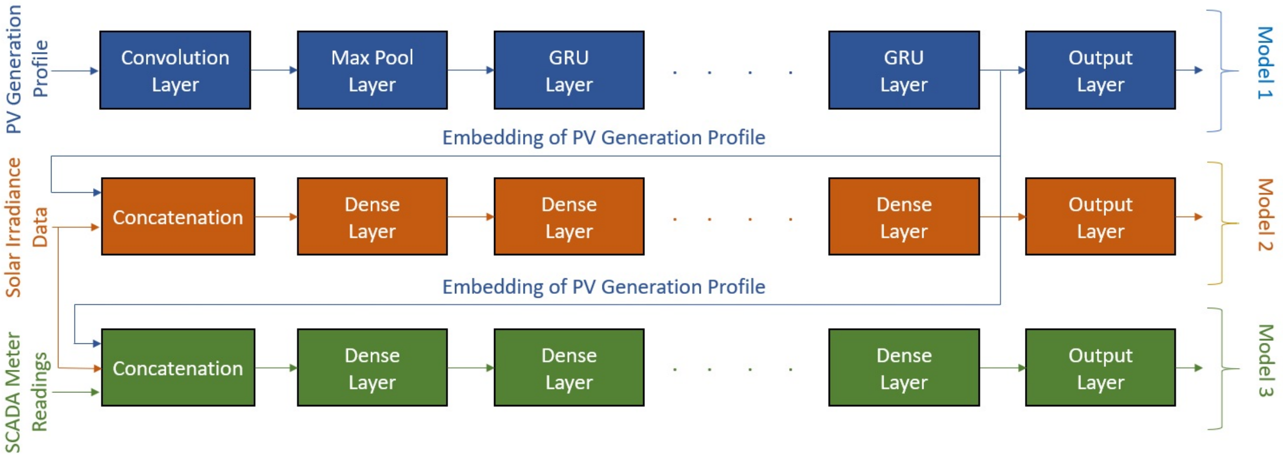

Fig. 3. Illustration of integration of different data sources within the model's training.

by a set of stacked dense layers to capture the relationship between the embedding of the PV generation profile, solar irradiance data, and the SCADA meter readings to classify the PV generation profile as benign or malicious sample. The complete architecture is illustrated in Figure 3. Model 1 (M1) in Figure 3 incorporates a single data source, namely, the PV generation profile. Model 2 (M2) integrates both the embedding of the PV generation profile and the solar irradiance data. Model 3 (M3) integrates the embedding of the PV generation profile, the solar irradiance data, and the SCADA meter readings. Hyper-parameter optimization is also applied on the extra added layers for M2 and M3.

V. Simulation Results

A. Implementation Details

For data preparation, the IEEE 123 bus test system discussed in Section III is implemented using a simulation environment that integrates both MATLAB and GAMS to solve the unbalanced power flow of the IEEE 123-bus for a period of one year. A for loop is included in the MATLAB to provide the load and generation data at each hour of the day to the GAMS program that solves the non-linear power flow equations. For the training and testing of the machine learning models, keras sequential API [24] is utilized when a single data source is used. Keras functional API [25] is utilized when various data sources are integrated. For hyper-parameter optimization, the following parameters are used in Algorithm 3: $\mathcal{L} = \{2, 3, 4\}$, $\mathcal{N} = \{64, 128, 256\}$, $\mathcal{A} = \{\text{Relu}, \text{Elu}, \text{Tanh}, \text{Sigmoid}\}$, $\mathcal{O} = \{\text{RMSProp}, \text{ADAM}, \text{SGD}, \text{AdaGrad}, \text{AdaDelta}, \text{AdaMax}, \text{ NADAM}\}$.

B. Single Data Sources Models

This subsection investigates the detector's training using a single data source, namely, PV generation profile. The objective is to judge which of the deep learning models presented in Section IV.A offers the best detection performance. Furthermore, the performance of the proposed deep learning-based detection schemes is compared with shallow classification based on an SVM model and time-series anomaly detection based on an ARIMA model. For the SVM-based classification benchmark, the classifier is trained on both benign and malicious PV energy generation profile and presents a class label as the output. For the ARIMA-based anomaly detection scheme, the model is trained only on the benign PV energy generation profile to learn the ARIMA model parameters that can predict the future generation profile while minimizing the error between the predicted and actual values. Then, the anomaly detector is tested on both benign and malicious datasets. Whenever the error between the predicted and reported generation profile is larger than a threshold, a malicious sample is detected. Table III presents the results of hyper-parameter optimization for the different deep learning detection models, using Algorithm 3. Hyper-parameter optimization of the SVM classifier yields $C = 10$ and RBF Kernel.

TABLE III
Optimal Hyper-parameters of the Neural Network Models

Model	Hyper-parameters				
	L	N	O	A_{H}	A_{O}
DNN	8	128	Nadam	Sigmoid	Sigmoid
GRU	4	64	Adagrad	Tanh	Sigmoid
CNN +	1	64	Rmsprop	Relu	
GRU	4	64	Rmsprop	Tanh	Sigmoid

Table IV summarizes the detection performance of the deep learning-based classifiers following the optimal hyper-parameters in Table III. As demonstrated in Table IV, the hybrid C-RNN detector offers the best detection performance among the other architectures. This is due to the fact that the C-RNN detector is trained on the most relevant features as extracted by the CNN while the GRU learns the temporal correlation within the data that distinguishes benign and malicious samples. Detection errors occur since we have various panel types (hence, various forms of benign PV generation profiles) and cyber-attack functions (hence, various forms of malicious samples). These factors could confuse the detector in discriminating benign from malicious samples. However, the reported detection and false alarm rates by the proposed detector demonstrate high detection performance. Furthermore, comparison results with a shallow classifier

TABLE IV
DETECTION PERFORMANCE OF THE PROPOSED DETECTORS IN COMPARISON WITH SVM AND ARIMA-BASED DETECTORS

Model	Test Results				
	DR	FA	HD	PR	F1
DNN	90%	2%	88%	97.8%	93.8%
GRU	91%	1.6%	89.4%	98.3%	94.4%
C-RNN	94.6%	2.6%	92%	98.7%	96.2%
SVM	88.3%	3.4%	84.9%	96.4%	92%
ARIMA	83%	22%	61%	75.5%	80%

TABLE V
OPTIMAL HYPER-PARAMETERS OF THE MODELS IN FIGURE 3

Model	Hyper-parameters				
	L	N	O	A_H	A_O
M1: CNN + GRU	1	64	Rmsprop	Relu	
	4	64	Rmsprop	Tanh	Sigmoid
M2: CNN + GRU + Dense	1	64	Rmsprop	Relu	Sigmoid
	6	64	Rmsprop	Tanh	
	3	64	Rmsprop	Sigmoid	
M3: CNN + GRU + Dense	1	64	Adam	Relu	Sigmoid
	4	64	Adam	Sigmoid	
	3	64	Adam	Sigmoid	

(SVM) and time-series anomaly detection (ARIMA) reveals performance improvement in detection rate from $83-88\%$ to 94.6% (improvement up to $11.6-6.6\%$). This is mainly due to the fact that deep machine learning techniques can better capture the complex patterns within the data, which further yield better detection performance. The high false alarm rate in the ARIMA model, and thus the lower detection performance compared with all other models, is due to the fact that the ARIMA model is trained only on the benign dataset while all other models including the shallow SVM classifier is trained on both benign and malicious datasets.

C. Integration of Multiple Data Sources

Since the hybrid C-RNN detector presents the best performance among other architectures, the C-RNN model is then tested for the integration of multiple data sources. The optimal hyper-parameters of the three models, namely M1, M2, and M3, illustrated in Figure 3 are summarized in Table V. Using such hyper-parameters, the detection performance of the three models is presented in Table VII. The ROC curve for the model with best detection performance (M3) is given in Figure 4. It is observed that the integration of the solar irradiance data within the model's training enhanced the HD from 92% to 98.2%. The incorporation of the SCADA meter reading further enhanced the HD to 99.08%. Such improvement in results is due to the fact that the detector has successfully learnt the relationship between the PV generation profile, solar irradiance data, and SCADA meter readings, which results in further improvement in the detection performance.

D. Robustness of the Detection Scheme

This subsection investigates the robustness of the proposed detection scheme against new cyber-attack functions. We consider in this subsection model M3 as it presents the highest detection performance. Three train and test cases are introduced. In the first case (C1), the detector is trained on benign PV generation data, solar irradiance data, SCADA meter readings, and the malicious dataset is based only on a single cyber-attack function, namely $f_1(E_{t,d})$. In the testing phase, the detector's performance is examined against all malicious and benign PV generation profiles. Hence, this case represents a scenario where the detector is tested against three new cyber-attack functions that are not part of the training dataset. The second case (C2) considers two cyber-attack functions, namely, $f_1(E_{t,d})$ and $f_2(E_{t,d})$, to create the malicious dataset of the training phase while the detector's performance is tested against all malicious and benign PV generation profiles. The last case (C3) considers three cyber-attack functions, namely, $f_1(E_{t,d})$, $f_2(E_{t,d})$, and $f_3(E_{t,d})$, to create the malicious dataset of the training phase, while the detector's performance is tested against all malicious and benign PV generation profiles. The performance results are summarized in Table VII. Such results demonstrate the robustness of the proposed detection scheme as the detector maintains a high detection performance even when new cyber-attacks are introduced in the testing stage. This is because the detector managed to generalize its learning experience to capture the main distinctive patterns in the benign PV generation profile and its relationship with solar irradiance data and SCADA meter readings, which is then used to detect new (unseen) cyber-attacks.

TABLE VI
DETECTION PERFORMANCE OF THE MODELS IN FIGURE 3.

Model	Test Results				
	DR	FA	HD	PR	F1
M1	94.6%	2.6%	92%	97.4%	96.2%
M2	99.1%	0.9%	98.2%	99.13%	99%
M3	99.3%	0.22%	99.08%	99.77%	99.55%

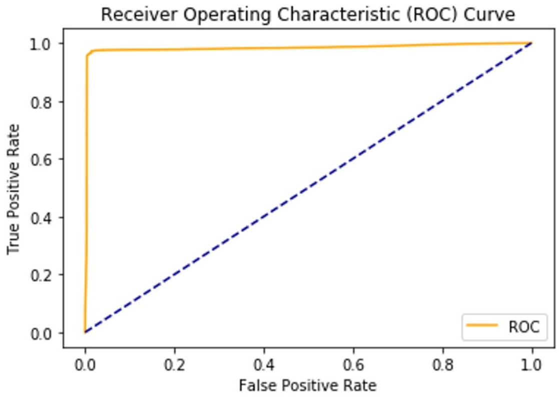

Fig. 4. ROC curve for model M3 presented in Figure 2.

VI. CONCLUSION

This paper investigated electricity theft detection in renewable energy-based DG units. A set of cyber-attack functions were introduced to manipulate the integrity of the readings of the injected power from the DG units in order to falsely overcharge the electric utility company. These cyber-attack functions include partial increment, minimum generation, and

TABLE VII
DETECTION PERFORMANCE OF M3 IN FIGURE 3 AGAINST NEW (UNSEEN) CYBER-ATTACKS.

Case	Test Results				
	DR	FA	HD	PR	F1
C1	97.38%	2.8%	94.58%	97.9%	97.6%
C2	97.7%	0.9%	96.8%	99.1%	98.4%
C3	98.4%	0.7%	97.7%	99.3%	98.8%

peak generation attacks. Our investigations revealed that a hybrid C-RNN deep learning architecture offers the best detection performance among different deep learning-based models. Optimal selection of hyper-parameters is investigated using a random grid search approach. Our studies also demonstrated that the detection performance can be significantly enhanced if multiple data sources are integrated while training the detector. In specific, the integration of the PV generation profile, irradiance data, and SCADA meter readings presented a detection rate of 99.3% and false alarm of only 0.22%. Furthermore, the robustness of the proposed detector is demonstrated against new cyber-attacks that were not present in the detector's training stage.

REFERENCES

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

[2] T. R. Sharafeev, O. V. Ju, and A. L. Kulikov, "Cyber-security problems in smart grid: cyber attacks detecting methods and modelling attack scenarios on electric power systems," *International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, pp. 1-6, 2018.

[3] Y. Tang, C. W. Ten, and K. P. Schneider, "Inference of tampered smart meters with validations from feeder-level power injections," *IEEE PES Innovative Smart Grid Technologies Conference (ISGT)-Asia*, pp. 1-5, 2019.

[4] G. M. Masters, Renewable and Efficient Electric Power Systems, second edition, *John Wiley & Sons Inc*, 2013.

[5] S. McIntyre, Termineter, Jan. 2018. [Online]. Available: https://github.com/securestate/termineter/blob/master/README.md

[6] V. B. Krishna, C. A. Gunter, and W. H. Sanders, "Evaluating detectors on optimal attack vectors that enable electricity theft and der fraud," *IEEE Journal on Selected Topics in Signal Processing*, vol. 12, no. 4, pp. 790-805, Aug. 2018.

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

[8] X. Yuan, M. Shi, and Z. Sun, "Research of electricity stealing identification method for distributed PV based on the least squares approach," *5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT)*, pp. 2471-2474, 2015.

[9] A. A. Cárdenas, S. Amin, G. Schwartz, R. Dong and S. Sastry, "A game theory model for electricity theft detection and privacy-aware control in AMI systems," *50th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1830-1837, 2012.

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

[11] T. Zhan et al, "Non-technical loss and power blackout detection under advanced metering infrastructure using a cooperative game-based inference mechanism," *IET Generation, Transmission, Distribution*, vol. 10, no. 4, pp. 873-882, Oct. 2015.

[12] A. Nizar, Z. Dong, and Y. Wang, "Power utility nontechnical loss analysis with extreme learning machine method," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 946-955, Aug. 2008.

[13] J. Nagi, K. Yap, S. Tiong, S. Ahmed, and F. Nagi, "Improving SVM-based nontechnical loss detection in power utility using the fuzzy inference system," *IEEE Transactions on Power Delivery*, vol. 26, no. 2, pp. 1284-1285, April 2011.

[14] E. Ângelos, O. Saavedra, C. O. Cortes, and A. Souza, "Detection and identification of abnormalities in customer consumptions in power distribution systems," *IEEE Transactions on Power Delivery*, vol. 26, no. 4, pp. 2436-2442, Oct. 2011.

[15] C. Ramos, A. de Sousa, J. Papa, and A. Falcao, "A new approach for nontechnical losses detection based on optimum-path forest," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 181-189, Feb. 2011.

[16] A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar and S. Mishra, "Decision tree and SVM-based data analytics for theft detection in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 1005-1016, June 2016.

[17] 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," *24th International Conference on Pattern Recognition (ICPR)*, Aug. 2018.

[18] Y. He, G. Mendis, and J. Wei, "Real-time detection of false data injection attacks in smart grid: a deep learning-based intelligent mechanism," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2505-2516, Sept. 2017.

[19] M. M. Othman, H. M. A. Ahmed, M. H. Ahmed, and M. M. A. Salama, "A techno-economic approach for increasing the connectivity of photovoltaic distributed generators," *IEEE Transactions on Sustainable Energy*, Sept. 2019.

[20] http://www.ieso.ca/en/Get-Involved/microfit/-/media/files/ieso/document-library/microfit/version-4/microFIT-Contract-version-4-1.pdf

[21] M. F. Shaaban, A. H. Osman, and F. M. Aseeri, "A multi-objective allocation approach for power quality monitoring devices," *IEEE Access*, vol. 7, pp. 40866-40877, 2019.

[22] H. He, Y. Bai, E. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," *IEEE World Congress on Computational Intelligence*, pp. 1322-1328, June 2008.

[23] I. Goodfellow, Y. Bengio, and A. Courville, 'Deep Learning, *MIT Press*, 2016.

[24] The Sequential Model API: https://keras.io/models/sequential/

[25] Getting Started with the Keras Functional API: https://keras.io/getting-started/functional-api-guide/