International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 06 | June - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

DEEP LEARNING DETECTION OF ELECTRICITY THEFT CYBER-ATTACKS IN RENEWABLE DISTRIBUTED GENERATION

VISWA.S

COMPUTER SCIENCE ENGINEERING & SETHU INSTITUTE OF TECHNOLOGY

Abstract - The 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 behaviour. 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. It integrating various data from the DG smart meters, meteorological reports, and SCADA metering points in the training of a deep convolutional-Artificial Neural Network is performed and analyse the cyber-theft on electrical data. The machine learning and feed neural network algorithm of Artificial Neural Network is applied and find the cyber-attack in benign and malicious Electrical data and predict the data. The prediction result is based on in the form of accuracy.

1.INTRODUCTION

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 develop a deep learning-based electricity theft detection system, we have investigated the application of deep feed forward, and deep artificial neural networks. The detector is trained using benign and malicious datasets. Hyper parameter 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. 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. Electrical theft leads to enormous losses to the utilities in the power sector. The major cause of these thefts is the illegal use of electricity by the consumers through tapping. To detect the malicious consumers that intentionally purloin the electricity.

2. 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 tech

niques that classify customers based on their load profifile 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 profifiles of the customers, which reported a classifification accuracy up to 70%.

An SVM-based classififier 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 classifification3 accuracy of 92.5%. The work in [1] relies on an SVM-based classififier and presents a wide range of electricity theft cyberattacks, which improved the classifification 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) classififier based on a gated recurrent unit (GRU) that is able to capture the temporal correlation within the customer's load profifile, 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%.

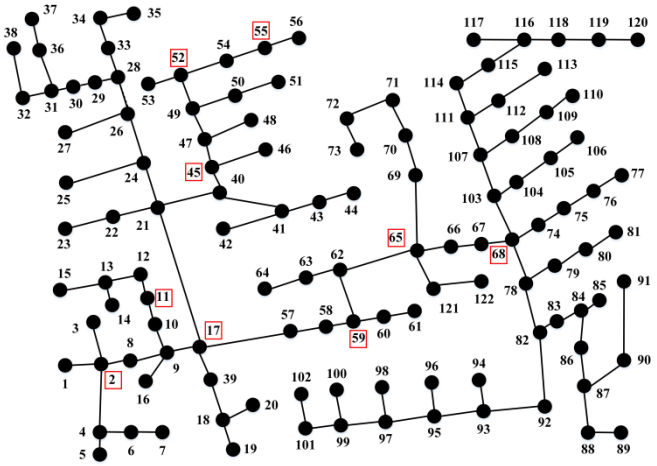
3 . DATA PREPARATION

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 profifiles 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 classififier.

*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 flflow within an IEEE distribution test system. Figure 1 presents the utilized 3-phase IEEE 123-bus test system.



customers, which represents a practical scenario with a mixture of residential and nonresidential units. The fifirst 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 profifile per residential household, real smart meter data from Ontario Canada is utilized [19]. The dataset presents load profifiles for customers over the four seasons of the year with a consumption reading reported every 60 minutes. The real load profifiles 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/m2) 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 defifined 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/m2 irradiance, *K*v and *K*i are the voltage and current temperature coeffificients, 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 profifile for each panel type, and hence for each residential customer, can be determined. Defifine the following terms at a specifific day $d \in D$ and specifific hour $t \in T$ 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 *F F* is the fifill factor of the PV panel. Hence, the generated power *P* PV *k,t,d* can be

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^{PV}. The aggregate generated power P^{PV} on

$$P\,\text{PV}\,i,p,t,d = \text{X}\,k\,P\,\text{PV}\,k,t,d \times N\text{PV}\,k,i,p\,S\text{base}$$

B. Malicious dataset

One of the challenges that face this research work is the absence of data that is needed to develop the desired classififier. 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 profifile. 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 profifile is to claim higher injected energy to the power grid. We introduce the four cyber-attack functions listed in Table II. The fifirst cyber-attack function $f1(Et,d)$ implements a partial increment attack, where a malicious customer reports an incremental percentage $(1+\alpha)$ of the actual generated energy Et,d (e.g., reporting 120% of the actual generation). The second function $f2(Et,d)$ 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 $f3(Et,d)$ 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 $f4(Et,d)$ 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 profifile matrix **E**, which results in four malicious matrices. The concatenation of the benign and malicious solar energy generation profifiles presents the entire dataset **X** where each row gives a sample energy generation profifile 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 **X** is denoted by . As we have four times malicious data than the benign one,

the trained classififier 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 **X** presents a zero mean and unit variance and is associated with the labeling vector **Y**. The data is then split into two disjoint sets with ratio 2:1, namely train data **X**TR with label **Y**TR and test data **X**TST with label **Y**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	$f_4(E_{t,d}) = \max(E_{t,d}, E_{t-1,d})$

4. DESIGN OF ELECTRICITY THEFT DETECTOR

In this section, we aim to develop a classififier that can detect cyber-attacks targeting the integrity of the readings about the 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.

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 **X**, the input layer consists of 24 neurons that are fed with the readings of the generated energy over the day, i.e., $xd \in \mathbf{X}$. 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 Wl defifines the weight $wl\ nn0$ of the connection from neuron $n0$ in layer $l-1$ to neuron n in layer l. The bias vector of layer l, bl , defifines the bias bln of neuron n in layer l. Let $zn = \mathrm{P}\ n0\ wl\ nn0\ al{-}1\ n0 + bln$ denote the weighted sum of inputs to neuron n, where $al = \sigma(zl)$ and $\sigma(\cdot)$ is an activation function. The training of the detector aims to fifind the model parameters Wl and bl denoted by Θ, which isachieved by minimizing the cross-entropy

$$\min_{\Theta} C = \frac{-1}{|\mathbf{X}_{\mathrm{TR}}|} \sum_{\mathbf{X}_{\mathrm{TR}}} \{y(x_d)\ln(a_n^L) + (1-y(x_d))\ln(1-a_n^L)\},$$

where $|\mathbf{X}\mathrm{TR}|$ denotes the number of training samples and $y(xd)$ denotes the label corresponding to sample xd. Iterative gradient descent is used to solve the minimization in (4). Let 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 networkbased 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: *5 a* represents partial derivative with respect to a, $\sigma 0$ ($zl(x)$) denotes the reciprocal of the partial derivative of zl with respect to al, 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: $\Delta^L(x) = \nabla_a C(x) \odot \sigma'(z^L(x))$ and $\Delta^l(x) = ((w^{l+1})^{\mathrm{T}}\Delta^{l+1}(x)) \odot \sigma'(z^l(x))$ $\forall l = L-1, \ldots, 2$
    end for
    Weight and bias update:
    $w^l = w^l - \frac{\eta}{K}\sum_x \Delta^l(x)(a^{l-1}(x))^{\mathrm{T}}$ 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:

spite 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 profifile represents a time-series data that is best handled using a recurrent neural network (RNN)-based classififier, 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 classififier consists of 24 neurons that are fed with the daily energy generation profifile $xd \in \mathbf{X}$. 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 ol with $o1 = xd$. A

hidden GRU layer l defifines 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

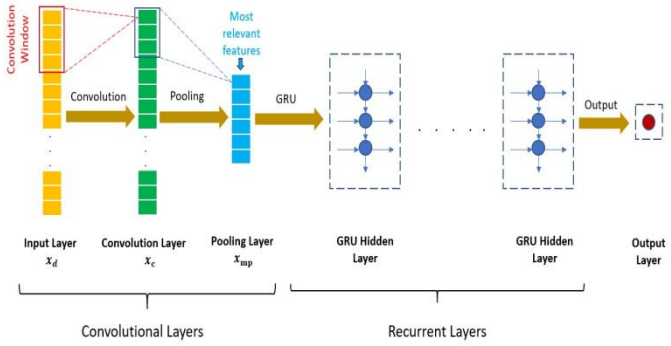
Initialization: Weights $U^l_{(\cdot)}$, $W^l_{(\cdot)}$, $V^l_{(\cdot)}$, and bias $b^l_{(\cdot)}$
$\forall l,\ i = 1$
while $i \neq I$ **do**
  Initialize: $m = 1$
  while $m \neq M$ **do**
    for *each training example* x_d *in mini-batch* m **do**
      Feed forward:
      for *each recurrent layer* 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: $\nabla_{U^l_{(\cdot)}} C(x_d)$, $\nabla_{V^l_{(\cdot)}} C(x_d)$, $\nabla_{W^l_{(\cdot)}} C(x_d)$, and $\nabla_{b^l_{(\cdot)}} C(x_d)$
    end for
    Weight and bias update:
    $U^l_{(\cdot)} = U^l_{(\cdot)} - \frac{\eta}{K} \sum_x \nabla_{U^l_{(\cdot)}} C(x_d)$
    $V^l_{(\cdot)} = V^l_{(\cdot)} - \frac{\eta}{K} \sum_x \nabla_{V^l_{(\cdot)}} C(x_d)$
    $W^l_{(\cdot)} = W^l_{(\cdot)} - \frac{\eta}{K} \sum_x \nabla_{W^l_{(\cdot)}} C(x_d)$
    $b^l_{(\cdot)} = b^l_{(\cdot)} - \frac{\eta}{K} \sum_x \nabla_{b^l_{(\cdot)}} C(x_d)$
  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 slt and it represents the memory that is computed using the hidden state $slt-1$ of the same layer at the previous time step.

• Update gate at time step t: This is denoted by zl and represents a combination of the new input $ol-1$ t and the previous memory slt -1 , given by $zlt = \sigma(ol-1\ t\ Ulz + slt-1Wlz + blz)$, Ulz and Wlz are learnable weight matrices, blz is a bias vector, and $\sigma(\cdot)$ is an activation function.

• Reset gate at time t: This is denoted by rlt and it specifiies the contribution of the memory $slt-1$ to the next state hlt We have $rlt = \sigma(ol-1\ t\ Ulr + sl\ -1Wrl + blr)$ and $hlt = \tanh(ol-1\ t\ Ulh+(-1\ rl)Wlh+blh)$, Ul , Wrl , Ulh , and Wlh are weight matrices and blr and blh are bias vectors. The next state is then calculated as $slt+1 = (1-zl)\ hl + zl\ slt$ and the output is given by $olt+1 = (Volslt+1 + blo)$, where Vol and blo are learnable weight matrix and bias, respectively. The objective of the detector's training stage is to learn the parameters $Ul(\cdot)$, $W(l\cdot)$, $V(l\cdot)$, and $bl(\cdot)$ that lead to the desirable output $y(xd)$ for input xd. 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



5 SIMULATER 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 flflow 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 flflow

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: $L = \{2, 3, 4\}$, $N = \{64, 128, 256\}$, $A = \{$Relu, Elu, Tanh, Sigmoid$\}$, $O = \{$RMSProp, ADAM, SGD, AdaGrad, AdaDelta, AdaMax, NADAM$\}$.

B. Single Data Sources Models

This subsection investigates the detector's training using a single data source, namely, PV generation profifile. 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 classifification based on an SVM model and time-series anomaly detection

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 + GRU	1 4	64 64	Rmsprop Rmsprop	Relu Tanh	Sigmoid

based on an ARIMA model. For the SVM-based classification benchmark, the classififier is trained on both benign and malicious PV energy generation profifile 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 profifile to learn the ARIMA model parameters that can predict the future generation profifile 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 profifile 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 classififier yields $C = 10$ and RBF Kernel. Table IV summarizes the detection performance of the deep learning-based classififiers following the optimal hyperparameters 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 profifiles) 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 classififier

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

Model	Hyper-parameters				
	L	N	O	A_H	A_O
M1: CNN + GRU	1 4	64 64	Rmsprop Rmsprop	Relu Tanh	Sigmoid
M2: CNN + GRU + Dense	1 6 3	64 64 64	Rmsprop Rmsprop Rmsprop	Relu Tanh Sigmoid	Sigmoid
M3: CNN + GRU + Dense	1 4 3	64 64 64	Adam Adam Adam	Relu Sigmoid Sigmoid	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 theshallow 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 profifile, solar irradiance data, and SCADA meter readings, which results in further improvement in the detection performance.

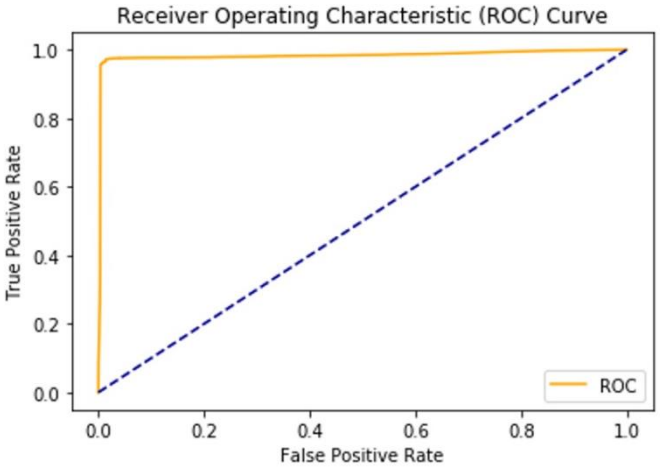
D. Robustness of the Detection Scheme

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

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%



introduced. In the fifirst 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 $f1(Et,d)$. In the testing phase, the detector's performance is examined against all malicious and benign PV generation profifiles. 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 cyberattack functions, namely, $f1(Et,d)$ and $f2(Et,d)$, to create the dataset of the training phase while the detector's performance is tested against all malicious and benign PV generation profifiles. The last case (C3) considers three cyberattack functions, namely, $f1(Et,d)$, $f2(Et,d)$, and $f3(Et,d)$, to create the malicious dataset of the training phase, while the detector's performance is tested against all malicious and benign PV generation profifiles. 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 cyberattacks 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 profifile and its relationship with solar irradiance data and SCADA meter readings, which is then used to detect new (unseen) cyber-attacks.

3. CONCLUSIONS

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 electricutility 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	FI
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 detecion performance among different deep learning-based models. Optimal selection of hyper-parameters is investigated using a random grid searchapproach. Our studies also demonstrated that the detection performance can be significantly enhanced if

multiple data sources are integrated while training the detector. In specifific, the integration of the PV generation profifile, 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 smartmeters 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 Effificient Electric Power Systems, second edition, *John Wiley & Sons Inc*, 2013.