Electrical Power and Energy Systems 134 (2022) 107345

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

Coordinated data falsification attack detection in the domain of distributed generation using deep learning

Narayan Bhusal [a], Mukesh Gautam [a], Raj Mani Shukla [b], Mohammed Benidris [a,*], Shamik Sengupta [b]

[a] Department of Electrical and Biomedical Engineering, University of Nevada, Reno, Reno, NV 89557, United States
[b] Department of Computer Science and Engineering, University of Nevada, Reno, Reno, NV 89557, United States

ARTICLE INFO

Keywords:
Coordinated data attacks
Distributed generation
Deep learning
Residual neural network
Multi-label classification

ABSTRACT

This paper proposes a deep learning-based multi-label classification approach to detect coordinated and simultaneously launched data falsification attacks on a large number of distributed generators (DGs). The proposed approach can detect coordinated additive, deductive, and combination of additive and deductive (attackers use the combination of additive and deductive attacks to camouflage their attacks) types of power output manipulation and falsification attacks on DGs. In training the proposed classifier, readings from DG meters and data from supervisory control and data acquisition (SCADA) systems along with meteorological data are used as input and class labels (additive, deductive, and combination) are used as output. The output class labels are developed based on the comparison between normal and compromised outputs of DGs. Two parallel data falsification classifiers with separate class labels are developed to increase the detection accuracy. The proposed approach is demonstrated on several systems including a 240-node real distribution system (based in the USA) and the IEEE 123-node distribution test system. The results show that the proposed approach can detect low margin coordinated attacks (as low as 5% of actual DG readings) with up to 99.9% accuracy. The performance of the proposed work is compared with multi-layer perceptron (MLP), convolutional neural network (CNN), and residual neural network. All of the developed source codes (including unbalanced quasi-static power flow in OpenDSS-MATLAB environment and deep learning in Python) of the proposed solution are publicly available at GitHub.

1. Introduction

Although the integration of advanced communication and automation technologies with distributed generators (DGs) can improve the operation and control of power grids, DGs could be exposed to multiple attack vectors through unsecured and unencrypted communications. The IEEE 1547–2018 standard mandates that all DGs must have communication capabilities to provide grid support functions during both normal conditions and contingencies [1], which can make them accessible by a number of different entities (e.g., manufacturers, utilities, aggregators, and consumers) and thus susceptible to a diverse of cyber-attacks. Therefore, sophisticated tools will be required to harden DGs against cyber-attacks through developing and employing real-time detection and classification algorithms. While there can be different types of cyber-attacks on DGs, this paper mainly focuses on coordinated data falsification attacks that can be initiated by manipulating DG's measurements and transmission infrastructure (e.g., advance metering infrastructure or AMI).

Several methods have been proposed in the literature to address the challenge of smart meter manipulations (e.g., electricity theft to reduce energy bills). In [2], electricity theft detection based on relative entropy has been proposed for smart grids with AMI. In [3], a combination of decision tree (DT) and support vector machine (SVM) is proposed to detect electricity thefts in smart grids. In [4], the typical coordinated attack scenarios have been proposed and analyzed based on bilevel optimization. In [5], an ensemble machine learning models have been proposed for the detection of electricity theft in power grids. A convolutional neural network (CNN) has been proposed in [6] for the detection of energy theft in smart grids.

In [7], wide and deep CNNs have been used for detecting electricity

* Corresponding author.
E-mail addresses: bhusalnarayan62@nevada.unr.edu (N. Bhusal), mukesh.gautam@nevada.unr.edu (M. Gautam), rajshukla@nevada.unr.edu (R.M. Shukla), mbenidris@unr.edu (M. Benidris), ssengupta@unr.edu (S. Sengupta).

https://doi.org/10.1016/j.ijepes.2021.107345
Received 25 October 2020; Received in revised form 25 May 2021; Accepted 30 June 2021
Available online 28 July 2021

thefts in smart grids. The method proposed in [7] has been implemented in two steps: in the first step, raw data are passed through a pre-processing unit to restore outliers and erroneous values; and in the second step, pre-processed data are fed to a wide and deep CNN framework for detecting electricity thefts. Two data mining techniques, which are maximum information coefficient (MIC) and clustering, have been used in [8] for detecting electricity thefts in smart grids. A deep recurrent neural network (RNN) based technique has been proposed in [9] to detect electricity thefts in which hyper-parameters are evolutionarily tuned. A statistical consensus-based two-tier approach has been proposed in [10] to detect orchestrated data falsification attacks on smart metering infrastructure for additive, deductive, and camouflage attacks. The first-tier determines whether the ratio between the harmonic to arithmetic mean of aggregated daily power consumption data is within a pre-determined safe margin; and the second-tier uses the sum of residuals between the ratios and the safe margin over multiple days to confirm attacks.

Developing methods to detect data falsification attacks on DGs have been gaining significant momentum in recent years. Spatial and temporal correlations between multiple solar farms have been utilized in [11] to detect data falsification attacks. Several methods have been investigated in [11] including the vector autoregressive model (VAR), deep neural network (DNN), long short-term memory neural (LSTM) networks, inverse principle component analysis (iPCA) technique, and deep auto-encoders (DAE). In [12], a least square approach and moving time window have been proposed to identify electricity theft in distributed photovoltaic (PV) systems. Detection of data attacks on individual DGs has been proposed in [13] using auto-regressive integrated moving average (ARIMA) models, Kullback–Leibler divergence (KLD), and PCA. Sandia National Laboratory in association with the U.S. Department of Energy (DOE)'s solar energy technologies office (SETO) has addressed the cybersecurity challenges at solar inverter levels in [14,15]. In [16], a deep learning-based approach has been proposed to detect data falsification attacks on PV systems. Historical DG meter readings, supervisory control and data acquisition (SCADA) measurements, and the solar irradiance data have been utilized in [16] to detect data falsification attacks on a single renewable DG.

Although energy consumption reading meters and DG meters have similar architectures for data transmission, loads and DGs are different in terms of their cyber-attack vectors. Most of existing data falsification detection methods, for both DG domain and energy consumption domain, generally take time (days) to detect data falsification attacks. Also, most of existing methods and algorithms tackle additive only or deductive only attacks on a single DG or a smart meter. Detecting a combination of additive and deductive attacks that are simultaneously launched on multiple DGs or smart meters in a coordinated manner is a challenging task [10]. Furthermore, most of existing methods are valid only for data manipulation attacks that are above 20% of the actual energy consumption/generation, which limits their detection reliability especially for combined additive and deductive attacks. Moreover, the majority of these methods have not utilized the prominent environmental parameters, which can boost the performance of data falsification detection methods. Therefore, it is critical to develop computationally efficient approaches that can detect and mitigate coordinated data falsification attacks on DGs in real-time with high accuracy for a wide range of attack intensities (i.e., to detect attacks with small changes in the system's overall output power). This speaks to a recent (February 2020) Funding Opportunity Announcement by the U.S. Department of Energy (DOE), Solar Energy Technologies Office (SETO), in which one of the most immediate and pressing concerns revolved around developing solutions to detect coordinated cybersecurity attacks on PV systems and other DGs [17].

This paper proposes a deep learning based multi-label classification approach to detect coordinated data falsification attacks in the DG domain, specifically, on PV and wind turbines. In the proposed approach, additive, deductive, and the combination of both attacks are tackled. The coordinated attacks are initiated on a large number of DGs simultaneously in such a way that it is difficult to detect using simple mean and proximity-based approaches. Several environmental parameters (e.g., ambient temperature and solar irradiance in case of PVs and wind speed in case of wind turbines) along with the readings from DG meters and SCADA measurements are taken into consideration and utilized to detect the attack in real time. Deep learning methods including multi-layer perceptron (MLP), convolutional neural network (CNN), and convolutional layer-based architecture of Residual neural network (ResNet) are investigated to test their performance to detect coordinated data attacks. Classical machine learning models including k-nearest neighbor, decision tree, and logistic regression are also investigated to test their performance to detect coordinated data falsification attacks. The experimental validation is performed on a 240-node real distribution system and the standard IEEE 123-node distribution test system. The results show that the proposed model can detect coordinated low margin (as low as 5% of actual DG readings) attacks with as high as 99.9% accuracy. Although the work presented in this paper is applied to attacks on PV and wind turbine generators, the proposed work can be adapted to detect coordinated data falsification attacks on other types of DGs.

The contribution of this paper can be summarized as follows.

1. Describing different types of attackers and their intention to attack distributed generation domains.
2. Developing and describing types of attacks that are possible in distributed generation domain.
3. Proposing a multi-label classification-based approach to detect and distinguish coordinated data falsification attacks in DG domain, specifically PVs and wind turbines. Accurately determining the nature of attacks (additive or deductive) in a DG is important since it provides power system operators with clear information to take further remedial actions.
4. Validating the proposed approach through several case studies on the United States-based 240-node real distribution system and the IEEE 123-node standard distribution test system.

The rest of the paper is organized as follows. Section 2 describes DG threat model. Section 3 provides a coordinated attack detection mechanism, the architecture of deep learning models, and evaluation metrics. Section 4 presents data generation preliminaries for this study. Section 5 examines the proposed approach through simulation studies. Section 6 provides concluding remarks.

2. Cyber-attaks on DGs

Threat modeling refers to the process of identification and classification of the prospective attacks or threats in order to make the system highly defensive. The threat modeling results in the development of attack vectors. While developing attack vectors, we need to think from a attackers' point of view. That's why the development of appropriate attack vectors is viewed as a very challenging task. In relation to DGs, various attackers with their profiles, motivations, and probable attack vectors are described in the following sections.

2.1. Types of attackers

Motivated by the work presented in [18–20] for electricity theft and cyber-attack in energy consumption domain, the type of attackers in DG domain can be categorized based on the level of expertise and motivations as follows—articles [21,17] have also described some of the unique cybersecurity challenges and attack scenario on DGs.

2.1.1. Low-tier individual attackers

These attackers have very little knowledge about computing and networking capabilities of AMI. They generally use vulnerable

authentication points using software tools such as Terminator [13] to launch attacks. Individual greedy DG owners who wish to increase their revenue fall under this category. These attackers initiate an additive attack to increase monetary burden to electric utilities. Also, some of individual attackers may initiate an attack for fun and learning.

2.1.2. Mid-tier coordinated attackers

These attackers may have coordinated multi-member network for business or learning purposes. These attackers usually have some knowledge of computing, communication, and networking capabilities of AMI. Using their expertise, they can discover new types of threats and can create and launch hardware and software tools to compromise DGs connected AMI infrastructures. This type of attackers may crack certain design aspects of AMI and utilize it for some monetary or non-monetary benefits. As an example of non-monetary benefit, utility company 'A' may initiate a coordinated attack on DG owners of company 'B' resulting in the deterioration of the reputation and credibility of company 'B', which may impel the customers of company 'B' to switch utility company. Criminal organizations (with its diverse multi-member commitment) who is trying to learn how to launch cyber-attacks also fall under this category.

2.1.3. Sophisticated coordinated attackers

These attackers have a highly coordinated and organized team with diverse expertise. They may be driven by various political motives. They have no motivation to attack a single DG unit, but rather they use cracked meters to initiate an integrated attack on whole power network, nuclear stations, etc. This type of attackers may belong to competing nation-states and terrorist organizations.

The main focus of this work is to detect coordinated data falsification attack launched by mid-tier adversaries for business purpose. These attackers try to attack a large number of DG owners without being easily detected from simple proximity and statistical consensus-based detection methods.

2.2. Data falsification attack functions

There can be several ways of developing coordinated data falsification attack functions. Basically, attack functions have been categorized into (a) additive attack, (b) deductive attack, and (c) combination of both additive and deductive attacks (camouflage), in energy consumption domain [22,10,23,24]. We have adopted similar attack functions with some adjustments for coordinated data attacks in DG domain. These attack functions are described as follows.

Let $i \in \{1, 2, 3, \ldots, N\}$ be a meter, where N is the total number of meters. Let $P^i_{DG,t}$ be power reported by a DG meter i to a utility at any instant t. For the unbiased DG meter, the actual power produced, $P^i_{DG,t}(\text{act})$, must be equal to the reported power to the utility (i.e., $P^i_{DG,t}(\text{act}) = P^i_{DG,t}$), while compromised DG meter could report any of the following falsified power generation data.

2.2.1. Additive attack

In this case, an attacker reports $P^i_{DG,t} = P^i_{DG,t}(\text{act}) + \Delta P^i_t$. The additive generation bias, ΔP^i_t, could be constant partial incremental (e.g., increasing by 20% all the time except at the time of zero generation), time-varying partial incremental, and changing power in such a way to report $P^i_{DG,t}$ as peak generation all the time. An additive attack could be launched by a single DG owner or organized criminal group or competing utility on DG meters of its rival company to manipulate readings of the DG meters for various purposes. The motive of individual DG owners and criminal groups could be self monetary benefits while that of competing utility companies could be to add monetary burden to push their rival companies toward bankruptcy.

2.2.2. Deductive attack

In this case, an attacker reports less output power than the actual amount, i.e., $P^i_{DG,t} = P^i_{DG,t}(\text{act}) - \Delta P^i_t$, where the deductive generation bias, ΔP^i_t, could be developed similar to that developed for additive attacks. Deductive attacks could be launched by a competing utility company on its rival company's premises to reduce the revenue of DG owners of the victim utility. Although this type of attack could increase the short-term monetary benefit to the victim utility, in the long run, it may induce loss of business confidence for DG owners.

2.2.3. Combined attack

This type of attack is launched in such a way that makes them extremely difficult to detect. For example, simultaneously applying the additive mode of attack to half of the meters and deductive mode of attack to other half of the meters. This mode of attack increases revenue to one set of DG owners with the cost of reducing revenue to other sets of DG owners. This type of attack may not be noticeable by mean-based statistical consensus methods because overall there can be little or no deviations in the total power generated by DGs under attack [10]. This type of attack could be launched by competing utility to induce the loss of the business confidence of DG owners of a victim company.

3. Coordinated data falsification attack detection model

This section describes the proposed attack detection mechanism and provides a brief description of various deep learning models, model training attributes, and evaluation metrics.

3.1. Attributes of the proposed attack detection mechanism

Most of existing attack detection mechanisms in the DG domain have been based on binary classification problems which detect whether there is an attack or not (1 or 0) on a single DG. However, because of the similar architecture of DG meter readings and smart meters in the energy consumption domain, multiple DGs are susceptible to coordinated attacks with different levels of data manipulations. The proposed solution classifies an attack on a DG into three categories: normal, additive, or deductive attack. For a power system with N DGs, we employ an N-label-based approach where each label is 0, -1, or 1 for normal, deductive, or additive attacks respectively.

From the perspective of machine learning-based methods, detecting an attack on a single DG is a single-label classification problem whereas classifying a large number of DGs at the same time with multiple labels is a multi-label classification problem. Although machine learning techniques have achieved significant progress in the single-label classification problem, they are still facing several challenges in solving multi-label classification problems. Different from single-label classification, multi-label classification problems are often evaluated with a conflicting multitude of quality measures [25,26]. Also, single-label balanced methods, such as down-sampling, cannot be applied on multi-label problems because the latter is extremely unbalanced in nature. Therefore, deliberate attention is needed to design multi-label classification problems.

Fig. 1 shows the block diagram of the proposed method. It can be seen from the figure that the proposed method takes environmental parameters (e.g., solar irradiance and ambient temperature for PV system), smart meter data, and SCADA measurements as inputs, and it outputs the classified label for each DG. In Fig. 1, there are two separate parallel branches for classifying additive and deductive attacks. The upper classifier is trained to detect additive attacks while the lower classifier is trained to detect deductive attacks. The outputs of the two classifiers combined gives the result of combined attacks.

Before developing this attack detection architecture, we have used a single multi-class classifier to detect both additive and deductive attacks. However, we have found out that the performance of the single

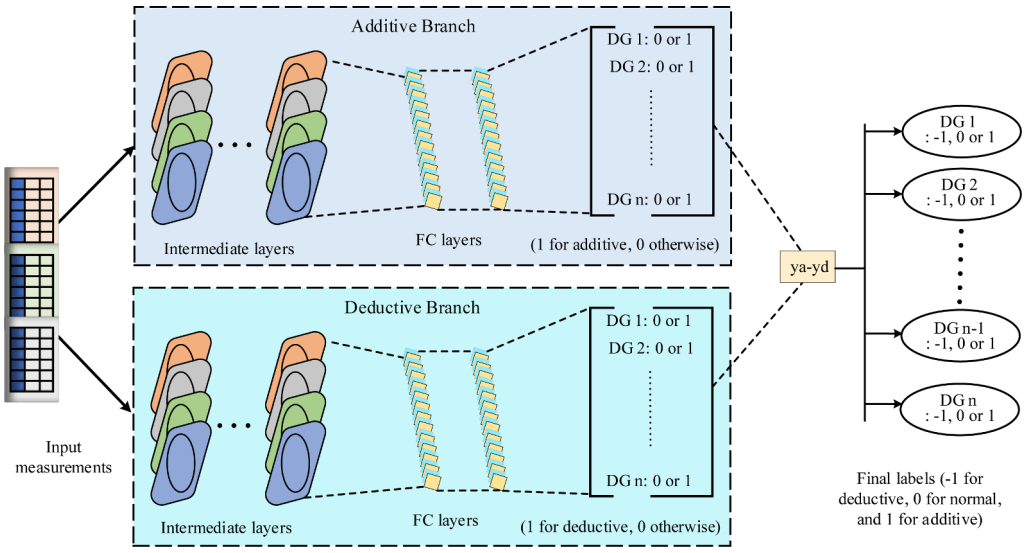

Fig. 1. General block diagram of proposed attack detection model. The measurements obtained from meteorological recording stations, smart meter data, and the SCADA measurements are the inputs. The output is the status of each DG, 0 for normal, −1 for deductive attack, and 1 for additive attack. FC layers stands for fully-connected layers. ya is the output classes label obtained from the additive branch and the yd is the output class label obtained from the deductive branch.

model in detecting a combination of attacks is low in terms of accuracy and precision. The reason is that deep learning methods detect accurate labels when readings of attacked DGs are either above or below the actual value. However, they cannot map the relationship between input measurements and output labels appropriately when detecting both above and below the actual values of DG readings.

During the training phase, a common measurement vector ($z_t = z_t^1, z_t^2, \ldots, z_t^m$) is provided as input and class labels ($y_{t,a}$ and $y_{t,d}$) are provided as output for additive and deductive branches, respectively. Here, the common measurement vector is used for both parallel classifiers because the operator/aggregator will not know whether the available measurements have positive generation bias, negative generation bias, or a combination of both. The training output class labels for additive and deductive branches are determined as follows.

$$y_{t,a}^k = \begin{cases} 1, & \text{if the DG } k \text{ is under additive attack at } t \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$y_{t,d}^k = \begin{cases} 1, & \text{if the DG } k \text{ is under deductive attack at } t \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

From (1) and (2), the additive branch gets compromised class labels for additive attacks whereas the deductive branch gets compromised class labels for deductive attacks.

The class labels obtained from the deductive branch are subtracted from the class labels obtained from the additive branch to get the final class labels for all the scenarios (additive, deductive, or combined). Mathematically, it can be expressed as follows.

$$\hat{y}_t = \hat{y}_{t,a} - \hat{y}_{t,d}, \tag{3}$$

where $\hat{y}_{t,a}$ and $\hat{y}_{y,d}$ are predicted output class labels from additive and deductive classifiers, respectively.

Final class labels for combined attacks are obtained using (3). The final classes are labeled as 0, −1, and 1 for uncompromised, deductive attacked, and additive attacked DGs, respectively. With different labels for additive and deductive attacks, the utility operator/aggregator can easily determine the type of attack on an individual DG. As additive and deductive attacks are mutually exclusive, if class-labels obtained from both additive and deductive branches for a specific DG are all 1's, Eq. (3) will automatically penalize it, which results in the reduction of the performance of the proposed model.

3.2. Architectures of machine learning models

Various machine learning models are investigated to determine their capability to detect the coordinated data falsification attacks in the DG domain. The architectures of the machine learning model investigated for the proposed work are described as follows. Note that our main purpose is not to advance machine learning techniques, rather utilize the advanced machine learning techniques to solve problem in power system domain.

3.2.1. Multi-layer Perceptron (MLP)

MLP has a layered architecture with input, hidden, and output layers. The normalized input is fed at the input layer. The cardinality of the input vector determines the number of neurons in the input layer. There can be multiple hidden layers in the MLP. The final prediction output is obtained from the output layer. The MLP structure presented for the proposed multi-label problem consists of an input layer taking measurements z_t, multiple hidden dense layers, and fully-connected output layers to classify whether the DGs are under attack or not. Using the input measurement vector z_t, the input layer generates the following feature maps.

$$mlp_1 = \sigma(w_1 \times z_t + b_1), \tag{4}$$

where σ denotes the activation function and w_1 and b_1 are weight and bias of the input layer. Non-linear rectified linear unit (ReLU) is used as an activation function for the input and the hidden layers.

Similarly, the feature map of the qth hidden layer can be expressed as follows.

$$mlp_q = \sigma(w_q \times mlp_{q-1} + b_q), \tag{5}$$

where w_q and b_q are weight and bias vectors of the qth layer and mlp_{q-1} is the feature map of the $(q-1)$th layer.

To determine the output classes of DGs, the sigmoid function is applied as an activation function. The output class vector can be expressed as follows.

$$\hat{y}_t = \sigma(w_d \times mlp_{d-1} + b_d), \tag{6}$$

where w_d and b_d are the weight and bias vector of the output layer, and mlp_{d-1} is feature map of the output layer; and m_{d-1} is the feature map of the layer just before the final layer.

The sigmoid activation function in the output layer produces continuous values between 0 and 1 for each DG. The continues values are converted into 0 or 1 using discriminator threshold (e.g., 0.5). For the proposed problem, two dense layer architectures of MLP are used; the number of neurons in the first layer equals two times the number of input measurements. Due to limited data, deeper networks tend to overfit training data and deteriorates the performance on test datasets. The number of neuron in the output layer is the total number (N) of DGs in the network. This structure is used for both the additive branch and the deductive branch of the proposed model (Fig. 1).

3.2.2. Convolutional Neural Networks (CNNs)

The CNNs generally have convolutional layers followed by pooling layers. The convolutional and pooling layers find the low-level feature of the input vector. Fully connected layers are added after the convolutional and pooling layers to predict the output. One of the benefits of CNNs is that they are easy to train, can automatically extract the features, and have a fewer parameters as compared to the fully connected neural network with the same number of hidden units. The feature maps of input, hidden, and output layers for CNN are described as follows.

The feature map generated by the input layer can be expressed as follows.

$$cnn_1 = \sigma(z_t * h_1 + b_1) \tag{7}$$

where h_1 is a convolutional kernel (1-D filter); b_1 is bias vector; and * is convolution operator. Similarly, feature map of the hidden layer can be presented as follows.

$$cnn_q = \sigma(cnn_{q-1} * h_q + b_q) \tag{8}$$

where cnn_{q-1} is the feature map of the $(q-1)$th layer and h_q and b_q are convolutional kernel and bias vector of the qth layer, respectively. After the convolution layers, some fully connected (FC) hidden layers are used. The feature map of the hidden FC or dense layer can be expressed as follows.

$$flat_{cnn} = \sigma(w_f \times cnn_{last} + b_f) \tag{9}$$

where cnn_{last} is the feature map of the last layer of the hidden convolutional layers and w_f and b_f are the weight and bias vector of the FC layer. The final output dense layer is same as the layer given in (6), which classifies attacks on DGs.

The architecture of CNNs for the proposed approach sequentially consists of two 1-D convolutional layers with 64 filters and a kernel size of 3, one flatten layer, and a dense layer with sigmoid as activation function to determine the classes of the DGs. Due to limited data, more deep networks tend to overfit the training data which lead to deterioration of the performance on test datasets. This structure is used for both the additive and the deductive branches of the proposed model given in Fig. 1.

3.2.3. Residual Neural Network (ResNet)

ResNet follows a structure similar to pyramidal cells in the cerebral cortex. ResNet is formed by skipping the connections or by jumping over some layers of the feed-forward neural network. Typical ResNet is formed by skipping two or three layers that contain batch normalization and a non-linear function (rectified linear unit (ReLU)) in between. Skipped connections are important in "vanishing" and "exploding" gradient issues by reusing activation function from a previous layer until the adjacent layer learns its weights [27,28]. Another advantage of skipping layers is that it simplifies the network and speeds up learning processes as fewer layers are used in the training.

To classify DGs, a convolutional layer based on ResNet is developed as shown in Fig. 2, where its derived architectures are provided in [29,28,30–32]. Hereinafter, the term "ResNetC" is used to denote this architecture. Each block consists of a number of hidden convolutional layers and a direct information flow from the input through a convolutional layer as shown in Fig. 2 with two blocks having two hidden layers in each of the blocks. The advantage of this architecture is that it improves the information flow and recovers the missing features. All the input and hidden layers of ResNetC consist of ReLU as an activation function and the output layer is a fully connected dense layer with sigmoid as an activation function. For this study, one flatten layer and one output dense layer are used for the ResNetC with three blocks and two hidden layers in each block. The first convolutional layer of each block of the ResNetC has 128 filters with kernel size of 3, and the rest of the layers have 64 filters (including the directly connecting convolutional layer) with kernel size of 3. This structure is used for both the additive and the deductive branches of the proposed model as shown in Fig. 1.

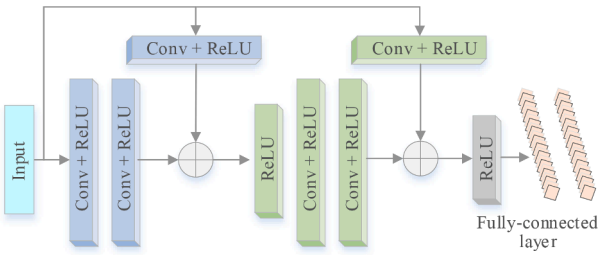

Fig. 2. Architecture of ResNetC with $K = 2$ and two blocks, where K denotes number of hidden layers in a block.

3.2.4. Classical machine learning models

Detection capability of various classical machine learning models including k-nearest neighbor (KNN), decision tree (DT), and logistic regression (LR) are also investigated to test their performance to detect the coordinated data falsification attacks. Detailed description of these models are not included for brevity.

In this work, we have taken the default parameter setting from the scikit-learn library of Python for KNN, DT, and LR. Since LR is a binary classifier, we have used the multi-output function of the scikit-learn library to address the multi-label classification problem through LR.

Several aspects including data processing, feature engineering, appropriate model selection, parameter tuning, parameter optimization, etc., are integral parts of machine learning-based approaches in the real applications and numerous works have been dedicated to it. However, these are out of scope of this work and we left them as future work. Readers are referred to [33,34] for further details on perspectives of machine learning models.

3.3. Training attributes

To use the proposed model in real-time, we need to train it to optimize the learning parameters such as weights, w bias, b, and convolutional kernel, h, in each layer. The optimized parameters can map the relationship between input measurements and output class labels.

Following the normal trend in the machine learning, the training and testing dataset are separated as 7/10 and 3/10, respectively. Different sizes of mini-batches are used for different case scenarios. Here, the mini-batch size is denoted by p for demonstration. We adopted commonly used cross-entropy as loss function which aims to minimize the deviation between the actual and the predicted class labels. The cross-entropy loss function over a mini-batch, $b = t_1, t_2, \ldots, t_p$, can be expressed as follows.

$$J = \sum_{t \in b} -\frac{1}{n} \sum_{k=1}^{n} (\hat{y}_t^k \ln(y_t^k) + (1 - \hat{y}_t^k \ln(1 - y_t^k))), \tag{10}$$

where $\hat{y}_t^k$ is the predicted classification of DGs. Adaptive moment estimation (Adam) is adopted as an optimizer to obtain the optimal parameters.

3.4. Evaluation metrics

The performance of the investigated deep learning models for the proposed work is compared using the following standard evaluation metrics.

1. Accuracy: It is the fraction of true predicted labels among all predicted labels, which can be expressed as follows.

$$A = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \tag{11}$$

2. Precision: It is the fraction of true predicted positive labels among all positive predicted labels, which can be expressed as follows.

$$P = \frac{T_p}{T_p + F_p} \tag{12}$$

3. Recall: It is the fraction of true predicted positive labels among the actual positive labels, which can be expressed as follows.

$$R = \frac{T_p}{T_p + F_n} \tag{13}$$

4. F_1-Score: It is the harmonic mean of precision and recall, which can be expressed as follows.

$$F_1 - \text{Score} = 2 \times \frac{P \times R}{P + R} \tag{14}$$

5. False Alarm: It is the fraction of false predicted positive labels among the actual negative labels, which can be expressed as follows.

$$FA = \frac{F_p}{F_p + T_n} \tag{15}$$

where T_p represents the true positive (compromised DG labeled as compromised); T_n denotes true negative (uncompromised DG labeled as uncompromised); F_p denotes the false positive (ncompromised DG labeled as compromised); and F_n is false negative (compromised DG is labeled as uncompromised).

Although the proposed model can have three labels ($0, -1$ or 1) in the final output, the results are demonstrated by taking labels -1 and 1 as compromised DGs and the label 0 as uncompromised. If separate metrics are required for additive and deductive classifier they can be easily determined.

4. Data generation preliminaries

Historical real power system datasets are not accessible for training and testing of the proposed data-driven cyber-attack detection framework. We have utilized a real power system (240-node distribution system) and power demand profiles obtained from smart meters based on U.S. to generate an uncompromised dataset. We have also utilized the standard IEEE 123-node test system with normalized power demand profile to generate a dataset to further validate the proposed model. The falsified dataset is generated using the threat models described in Section 2 and the uncompromised dataset. The detailed process of normal and falsified dataset generation is described as follows.

4.1. Uncompromised dataset generation

A real 240-node distribution feeder and the standard IEEE 123-node distribution test feeder are used to generate the normal (uncompromised) dataset.

The 240-node system is a real distribution system located in Midwest U.S.A. [35]. This system as shown in Fig. 3 consists of three feeders (17 nodes on feeder A, 60 nodes feeder B, and 162 nodes on feeder C) that are supplied by a 69 kV substation. This distribution system has a total of 23 miles of primary feeder conductor. This test system provides power supply to more than 1100 customers supplied through secondary transformers. Real power consumption (in kW) is obtained directly from smart meters installed at 1120 customer premises. The dataset range from January 1*st* to December 31, 2017 in the frequency of one hour. The reactive power profile for each node has been generated using a randomly picked power factor in the range of 0.9 to 0.95. In this paper, we have assumed that there are 77 (5 on feeder A, 20 on feeder B, and 52 on feeder C) intelligent electric devices (IEDs) installed as SCADA measurement units as shown in Fig. 3 by orange squares. The number and locations of SCADA measurement devices are arbitrarily chosen because the actual locations and number of IEDs of the system are not known. Determining the required number of power system monitoring devices to make a system observable is out of the scope of this work—several methods have been proposed in the literature for determining optimal number and locations of IEDs to ensure that power systems are observable [36–38].

The IEEE-123 node test feeder, as shown in Fig. 4, is characterized by having overhead and underground lines, four voltage regulators, four shunt capacitor banks, multiple sectionalizing and tie-switches, and unbalanced loading with constant current, power, and impedance models. The total real and reactive loads of this system are, respectively, 3490 kW and 1925 kVar. Network data of the IEEE 123-node test feeder are given in [39]. We have assumed that a total of 20 IEDs are installed by an electric utility as SCADA measurement units as shown in Fig. 4 by orange squares.

The historical solar irradiance, ambient temperature, and wind speed datasets are obtained from the system advisor model (SAM) provided by the U.S. National Renewable Energy Laboratory (NREL). For the 240-node system, the meteorological data of the city of Ames (a city in Iowa, USA) are used. For the IEEE 123-node system, meteorological data from the city of Reno (a city in Nevada, USA) are used.

Different levels of PV and wind turbine generator penetration are described in specific case studies in the experimental validation section. The power generated by PV depends upon solar irradiance and ambient temperature whereas that of a wind turbine depends upon the wind speed. Power generation preliminaries of wind turbine generators and the PV panels are described as follows.

4.1.1. Output power of wind turbine generators

The output power produced by a wind turbine considering wind turbine physical constraints can be expressed as follows [40].

$$P_w = \begin{cases} 0, & \text{if } v < v_{\text{cut-in}} \\ \frac{1}{2}\rho A C_p v^3, & \text{if } v_{\text{cut-in}} \leqslant v < v_r \\ P_{wr}, & \text{if } v_r \leqslant v < v_{\text{cut-out}} \\ 0, & \text{if } v_{\text{cut-out}} \leqslant v \end{cases}, \tag{16}$$

where P_{wr} represents rated output power of wind turbine (watts); ρ denotes the air density (kg/m^3); v is wind speed (m/s); C_p is the power coefficient; A denotes area swept by wind turbine (m^2); v_r is the rated speed; and $v_{\text{cut-in}}$ and $v_{\text{cut-out}}$ are the designed cut-in speed and cut-out speed, respectively. It can be seen from (16) that wind power generation is highly dependent on wind speed. To generate output profiles for wind turbines, the following typical parameters are assumed based on Vesta 2-MW type wind turbines: $P_{wr} = 2$ MW; $\rho = 1.225$ (kg/m^3); $C_p = 0.45$; $A = 6363$ m^2; $v_r = 15$ m/s; $v_{\text{cut-in}} = 4$ m/s; and $v_{\text{cut-out}} = 25$ m/s.

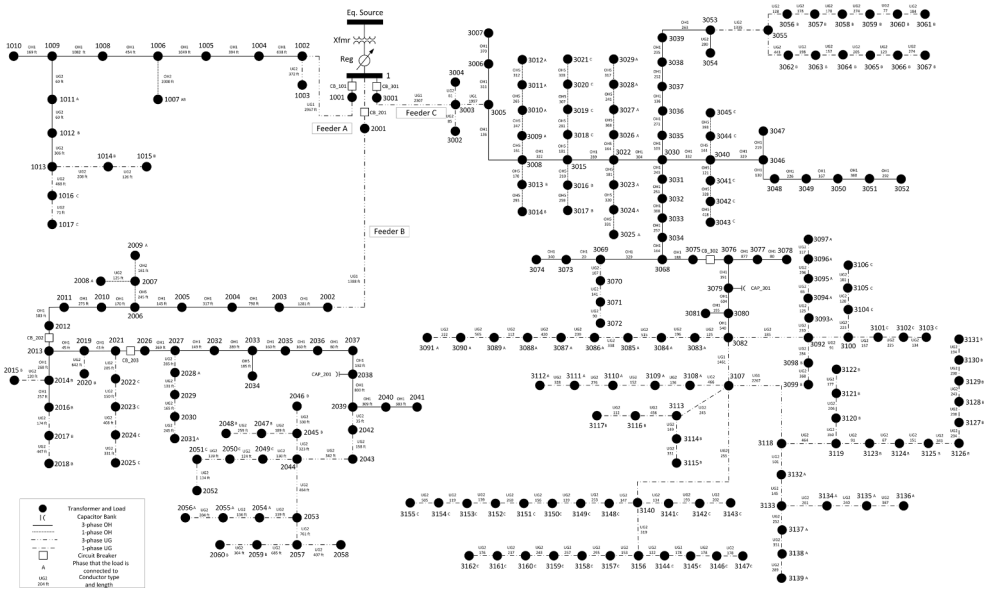

Fig. 3. One-line diagram of 240-node distribution test system. This is a Midwest U.S.A.-based real distribution grid. The real system owned by a municipal utility and is a fully observable network with smart meters installed at all customers. Notice that the secondary transformers have been used to connect the customers to these primary network nodes [35].

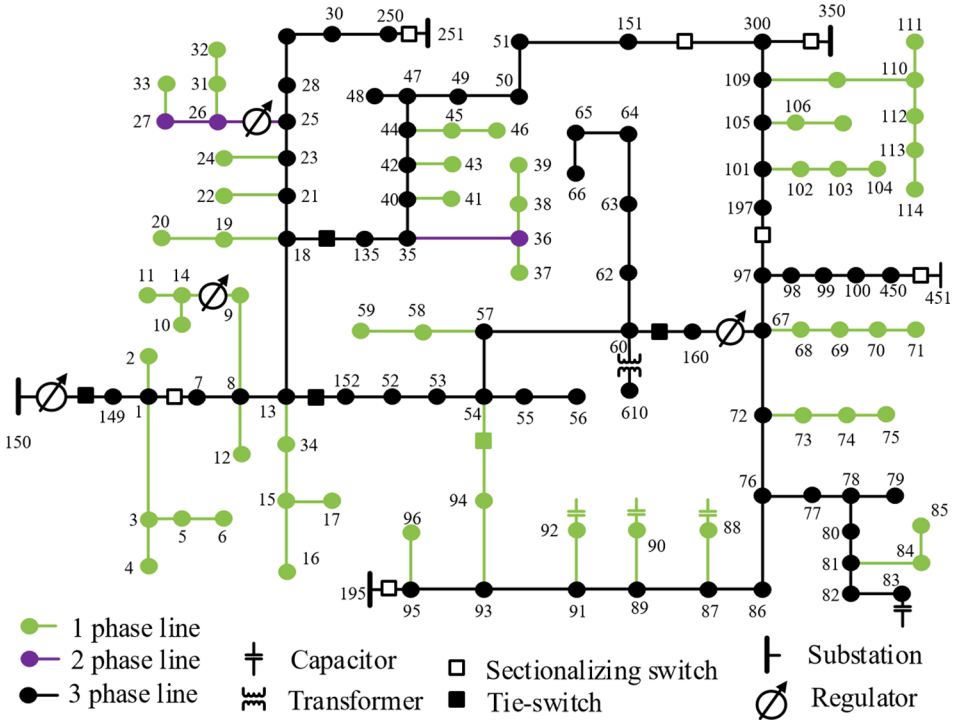

Fig. 4. IEEE 123 node distribution test system.

4.1.2. Output power of photovoltaic panels

The output power produced by a solar panel can be calculated as follows [41].

$$P_{PV} = F_f \times V_{OC} \times I_{SC} \tag{17}$$

where V_{OC} is the open circuit voltage which depends upon cell temperature; I_{SC} is the short-circuit current of a PV panel which depends-on solar irradiance and cell temperature; and F_f is the fill factor which can be expressed as follows.

$$F_f = \frac{V_{MPPT} \times I_{MPPT}}{V_{OC} \times I_{SC}} \tag{18}$$

where V_{MPPT} and I_{MPPT} are respectively the voltage and current at maximum power point tracking (MPPT). MPPT can be defined as a point at which the output power produced by a solar panel is maximum. The output power produced by PV panels directly depends on solar irradiance and cell temperature. As cell temperature directly depends upon the ambient temperature, the inclusion of ambient temperature along with the solar irradiance model can improve the performance of the proposed method to detect the coordinated cyber-attacks.

4.2. Generation of compromised datasets

In this work, we have assumed that an adversary can only manipulate smart meter data and cannot manipulate the meteorological recording station data and SCADA measurements (IEDs). This is an acceptable assumption as IEDs and meteorological recording station data have an extra level of security than AMI infrastructures. We have also assumed that attackers launch organized and powerful attacks from several DG meters. Even if an attacker tries to keep false data at low margin, the attack can cause long-term damage to a large number of nodes without being easily detected.

Malicious data required for our case are created by adding attack vectors as described in Section 2.2 to the uncompromised dataset generated based on the approach described in Section 4.1. An intelligent attacker can falsify normal datasets in several ways including: constant percentage increase or decrease attack (e.g., 20 percent positive or negative generation bias at all DG units under attacks); varying percentage increase or decrease attack (varying the percentage of false data for different DGs); zero generation attack (reporting zero generation during the time of minimum power generation in all of the generators); minimum generation attacks (reporting minimum generations from all customers instead of actual values); maximum generations attack (reporting maximum generations from all of the DG meters); and combination of constant or time-varying simultaneous increase and decrease attacks. After applying these attack strategies, if the power generation is greater than the actual value, then the attack is additive; it is deductive attack if the generation is less than the actual value. In the case of a combined attack, both additive and deductive attacks are launched on a different set of DG units.

In this paper, it is assumed that the attacker launches combinations of additive and deductive attacks to avoid easy detection. In this process, out of the total N DGs, N_c DGs are compromised. Out of total compromised DGs, half will face additive attack and the other half will face deductive attack. As the intelligent attacker can change the generation bias, time to time, the proposed model is trained and tested on a dataset with time-varying generation bias. At each time sequence, a random number between 5 and 20 is generated and that percentage of actual generation is used as a generation bias of the instant. The generation bias are limited within 5% to 20% (i.e., $\Delta P_t = 0.05{*}P_{DG,t}(\text{act})$ to $0.2{*}P_{DG,t}$ (act)) because generation bias higher than 20% can be easily detected and that generation bias lower than 5% may not justify attack budget of the attacker.

Algorithm 1 provides the procedure of attack function and class label generation.

Algorithm 1. Attack function and class labels generation for the

---

Input : Input uncompromised readings $P_{DG}(act)$
Initialize $P_{DG} = P_{DG}(act)$
Randomly generate n_a attack instants from the total of 8760 time sequences
for $k \leftarrow 1$ **to** n_a **do**
  randomly choose N_c (out of total N meters) meters that are attacked
  choose N_c^a and N_c^d meters ($N_c = N_c^a + N_c^d$) with additive and deductive attacks, respectively
  randomly generate the size of percentage generation bias lying between 5% and 20%
  obtain the generation bias for all the attacked meters
  apply the additive and deductive attacks in respective meters
  obtain the generation values after the attacks
  update P_{DG} including the attacks
Output1: Generation with attack function, P_{DG}
Determine the additive, deductive, and overall classes
Initialize deductive class $P_{\text{class, ded}} = \text{zeros}(8760, N)$
Initialize additive class $P_{\text{class, add}} = \text{zeros}(8760, N)$
Initialize overall class $P_{\text{class}} = \text{zeros}(8760, N)$
for $i \leftarrow 1$ **to** 8760 **do**
  for $j \leftarrow 1$ **to** N **do**
    if $P_{DG}(i,j) < P_{DG}(act)$ **then**
      $P_{\text{class, ded}}(i,j) = 1$
      $P_{\text{class}}(i,j) = -1$
    if $P_{DG}(i,j) > P_{DG}(act)$ **then**
      $P_{\text{class, add}}(i,j) = 1$
      $P_{\text{class}}(i,j) = 1$
Output : P_{DG}, $P_{\text{class, ded}}$, $P_{\text{class, add}}$, and P_{class}

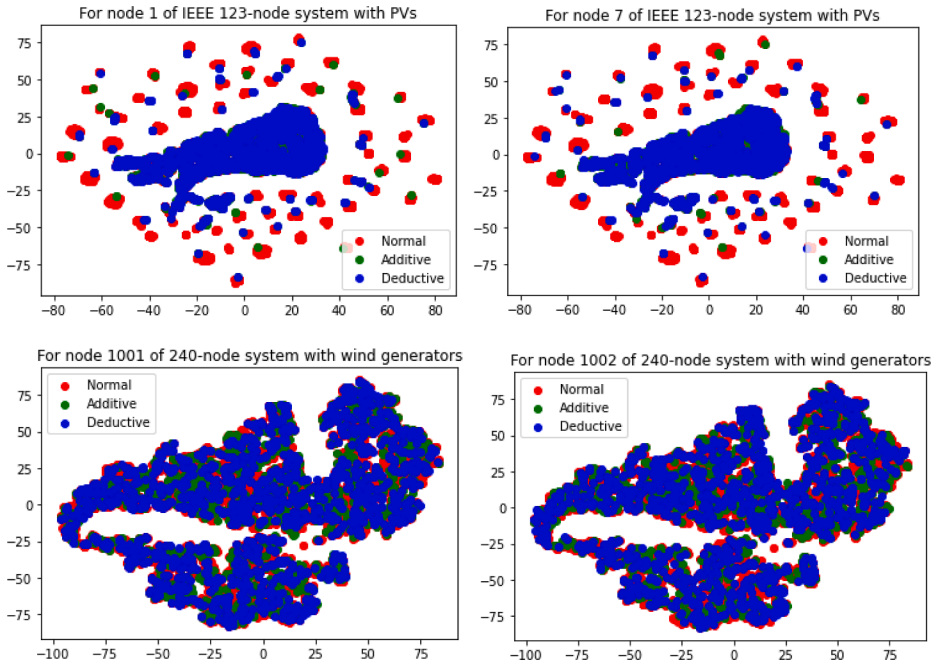

Fig. 5. t-SNE for the feature visualization of the 8760 data sample of the proposed work.

proposed work.

4.3. Visualization of data features

Table 1
Performance of MLP, CNN, and ResNetC for IEEE 123 bus system with PV only integrated scenario. KNN represents k-nearest neighbor, DT denotes decision tree, and LR represents logistic regression.

Model structures	A (%)	P (%)	R (%)	F^1 (%)	FA (%)
KNN	76.62	72.68	49.81	59.11	9.60
DT	75.94	72.135	47.36	57.18	9.392
LR	92.21	99.99	77.043	87.03	0.001
MLP	92.117	93.75	82.24	87.62	2.81
CNN	98.52	97.22	98.40	97.81	0.90
ResNetC	99.02	98.35	98.75	98.55	0.84

The insight of features of the data of the proposed work can be visualized using t-distributed stochastic neighbor embedding (t-SNE)

Table 2
Performance of MLP, CNN, and ResNetC for 240 node distribution test system with PV only integrated scenario. KNN represents k-nearest neighbor, DT denotes decision tree, and LR represents logistic regression.

Model structures	A (%)	P (%)	R (%)	F^1 (%)	FA (%)
KNN	74.46	73.46	42.88	54.15	8.40
DT	77.049	99.41	32.50	49.02	0.097
LR	76.81	82.77	43.048	56.64	4.86
MLP	90.48	92.63	79.25	85.42	3.41
CNN	92.50	89.98	88.53	89.25	5.35
ResNetC	97.36	96.55	95.92	96.23	1.8

[42,43]. The t-SNE is a non-linear dimension reduction technique for the representation of the high-dimension data in low-dimension space. Fig. 5 shows that some of the nodes of the IEEE 123-node system and a 240-node distribution system as representative cases of the dataset of the proposed work. As can be seen from the figure, there is a significant overlap between normal and malicious conditions when using t-SNE representation with reduced features. For malicious conditions too, there is an overlap between additive and deductive attacks. Due to this overlap, developing a classifier to separate normal conditions, additive attacks, and deductive attacks is a complex problem.

5. Experimental validations

Comprehensive case studies are performed on the IEEE 123-node test system and a real 240-node distribution system. The performance of deep learning models: MLP, CNN, and ResNetC are tested for three scenarios: (a) PV panels only integrated scenario (Section 5.1); (b) wind turbine only integrated scenario (Section 5.2); and (c) both PV and wind turbine generators integrated scenario (Section 5.3). These scenarios are chosen to show the effect of varying the power output of different renewable DGs on the performance of the proposed approach. Another reason for choosing these cases is to represent distribution systems in different geographical regions with different prospects of PV and wind powers. For example, the system with PV only scenario represents the geographical region with prospects of only solar power without any prospect of wind power. Similarly, some geographical areas may have prospects of more than one renewable source (e.g., solar, wind, and water resources) and can produce a significant amount of power from all of them. MLP, CNN, and ResNetC as described in Section 3.2 are used on

Table 3
Performance of MLP, CNN, and ResNetC for IEEE 123 bus system with wind generators only integrated scenario. KNN represents k-nearest neighbor, DT denotes decision tree, and LR represents logistic regression.

Model structures	A (%)	P (%)	R (%)	F^1 (%)	FA (%)
KNN	86.83	75.36	35.93	48.66	2.46
DT	87.23	69.82	46.69	55.94	4.23
LR	96.69	99.95	81.00	89.48	0.006
MLP	99.74	99.63	98.88	99.25	0.075
CNN	99.96	99.91	99.90	99.90	0.018
ResNetC	99.94	99.82	99.87	99.85	0.036

Table 4
Performance of MLP, CNN, and ResNetC for 240 node test system with wind generators only integrated scenario. KNN represents k-nearest neighbor, DT denotes decision tree, and LR represents logistic regression.

Model structures	A (%)	P (%)	R (%)	F^1 (%)	FA (%)
KNN	66.99	30.81	6.25	10.25	6.20
DT	73.37	59.05	42.50	49.43	13.00
LR	79.62	74.33	51.06	60.54	7.77
MLP	91.81	90.94	81.35	85.88	3.570
CNN	99.61	99.74	98.99	99.36	0.110
ResNetC	99.88	99.80	99.81	99.81	0.086

Table 6
Performance of MLP, CNN, and ResNetC for 240 node test system with both PV and wind generators integrated scenario. KNN represents k-nearest neighbor, DT denotes decition tree, and LR represents logistic regression.

Model structures	A (%)	P (%)	R (%)	F^1 (%)	FA (%)
KNN	80.88	39.45	7.17	12.144	2.43
DT	83.11	55.71	40.23	46.23	7.21
LR	84.90	79.35	24.31	37.21	1.42
MLP	86.08	80.89	31.96	45.82	1.70
CNN	94.53	88.76	80.48	84.42	2.29
ResNetC	96.72	92.10	89.87	90.97	1.73

both of the branch of proposed architecture of Fig. 1.

Since the tested systems are unbalanced systems, power flow is solved using an OpenDSS and MATLAB integrated environment to generate the dataset for the proposed method. OpenDSS is an open distribution system simulator developed by Electric Power Research Institute (EPRI) [44]. OpenDSS calculates unbalanced power flow using Newton's Method. While integrating MATLAB and OpenDSS, all distribution system data including real and reactive power profile of each node and meteorological data for renewable DG are provided in the MATLAB environment. For solar PV, temperature and solar irradiance are provided whereas for the wind turbine, power generation output is obtained using (16) with time-varying wind speed. MATLAB performs all control and loop structures and calls the OpenDSS engine in order to perform unbalanced power flow calculations. OpenDSS provides all monitored information back to the MATLAB to perform the rest of the calculation. The unbalanced power flow is run for 8760 snapshots and the power generation profile of each node with DG and readings of all SCADA measurements are recorded. Keras with the Tensorflow backend (in Python) is used for training and testing deep learning models.

Although the proposed multi-label classification approach can detect the three scenarios (additive only, deductive only, and combined attack), the results are shown only for the combined attacks (results for additive only and deductive only attack are not provided because of the simplicity of the proposition—the shared source code can be used to reproduce results for those scenarios).

5.1. PV panels only integrated scenarios

For this scenario, a total of 40% of peak load at all three-phase nodes of the distribution system is considered as a PV penetration level. For the IEEE 123-node system, there are 56 three-phase nodes. A total of 56 PV panels, each having size 40% of specific three-phase loads, are installed. Similarly, for 240-node distribution system, a total of 108 PV panels (12 on feeder A; 47 on feeder B; and 49 on feeder C) are installed on 108 three-phase nodes. The sizes of PV panels are assumed as 40% of the size of loads at corresponding nodes. If there are no loads connected at three-phase nodes, the size of PV panels is 40% of the load of the neighboring node. Although the sizes of PV panels are chosen arbitrary based on empirical data, actual sizes can be used for real power systems. While same profile of solar irradiance and temperature are used for all PV panels, different temperatures and solar irradiance profiles can be used if they are available. Out of a total of 8760 time sequence data, approximately 3200 (this number varies due to the random process of attack generation function) are compromised DGs and the rest are normal.

Table 5
Performance of MLP, CNN, and ResNetC for IEEE 123 bus system with both PV and wind generators integrated scenario. KNN represents k-nearest neighbor, DT denotes decision tree, and LR represents logistic regression.

Model structures	A (%)	P (%)	R (%)	F^1 (%)	FA (%)
KNN	80.80	88.66	32.36	47.41	1.51
DT	88.44	91.42	62.66	74.36	2.14
LR	92.39	100	71.57	83.43	0.00
MLP	91.29	86.12	62.85	72.67	2.280
CNN	99.45	99.96	97.98	98.96	0.012
ResNetC	99.64	100	98.67	99.33	0.000

For this scenario, a mini-batch size of 128 is used for MLP and is run for 1500 epochs. CNN and ResNetC are run for 500 epochs with a mini-batch size of 16.

We have chosen these parameters based on empirical evaluation of network configurations. Although several aspects including data processing, feature engineering, appropriate model selection, parameter tuning, parameter optimization, etc., are integral parts of machine learning-based approaches in the real applications, numerous methods have been dedicated to it and details on optimal parameter tuning, feature engineering, data cleaning, etc., are out of scope of this work.

Tables 1 and 2 show the performance of MLP, CNN, and ResNetC in terms of accuracy, precision, recall, F_1-score, and false alarm for the 123- and 240-node distribution test systems, respectively.

Note that the most time consuming part of the proposed machine learning driven work is to train the machine learning model which is done offline. Once the machine learning models are trained, they can be used in real time to detect cyber attacks.

For the IEEE 123-node distribution system with PVs, the trained machine learning model takes 3.087 s to predict the class labels of 2628 samples. Therefore, on average it takes 1.17 ms per sample. For the 240-node distribution system with PVs, the trained machine learning model takes 12.72 s to predict the class labels of 2628 samples. Therefore, on average it takes 4.84 ms per sample for the 240-node system.

5.2. Wind turbine only integrated scenarios

For this scenario, a total of 20 and 44 wind turbine generators have been installed for the IEEE 123-node test system and the 240-node distribution system. Wind turbine generators are randomly located on the three-phase nodes with sizes equals to 80% of the respective loads for both systems. Although these sizes and locations are chosen arbitrarily, actual sizes and locations of wind turbine generators can be used if available. Out of a total of 8760 time sequence data, approximately 2000 (this number varies due to the random process of attack generation function) are compromised DGs and the rest are normal.

In this case, a mini-batch size of 128 is used for MLP and is run for 1500 epochs. CNN and ResNetC are run for 500 epochs with mini-batch size of 16.

Tables 3 and 4 show the performance of MLP, CNN, and ResNetC for the proposed problem in terms of accuracy, precision, recall, F_1-score, and false alarm for IEEE 123- and 240- node distribution systems, respectively.

For the IEEE 123-node distribution system with wind generators, the trained machine learning model takes 3.12 s to predict the class labels of 2628 samples. Therefore, on average it takes 1.18 ms per sample. For the 240-node distribution system with wind generators, the trained machine

learning model takes 8.495764 s to predict the class labels of 2628 samples. Therefore, on average it takes 3.23 ms per sample.

5.3. Both PV and wind integrated scenarios

For this scenario, a total of 25 PV panels and 15 wind turbine generators are integrated into various three-phase nodes for the IEEE 123-node distribution system. For 240 node distribution system, a total of 50 PV panels and 30 wind turbine generators are integrated. The size of PV panels at a selected location is 40% of the peak load of that location. The size of a wind turbine generator at a selected location is 80% of the peak load of that location. Out of total 8760 time sequence data, approximately 2600 are compromised DGs and the rest are normal.

In this case, a mini-batch size of 128 is used for all tested models. MLP is run for 1500 epochs while CNN and ResNetC are run for 500 epochs.

Tables 5 and 6 show the performance of MLP, CNN, and ResNetC in terms of accuracy, precision, recall, f_1-score, and false alarm for IEEE 123 and 240 node distribution systems, respectively.

For the IEEE 123-node distribution system with both PVs and wind generators, the trained machine learning model takes 2.84 s to predict the class labels of 2628 samples. Therefore, on average it takes 1.082 ms per sample. For the 240-node distribution system with both PVs and Wind generators, the trained machine learning model takes 10.18 s to predict the class labels of 2628 samples. Therefore, on average it takes 3.78 ms per sample.

5.4. Discussion

The results obtained using CNN and ResNetC for all cases are much better than that of MLP. This is because the structure of CNN and ResNetC allow them accurately map the relationship between input measurements and output states. In other words, they can better capture complex and interrelated patterns within data sets than MLP.

It can be seen from Table 1 that the overall accuracy of class labels from ResNetC is 99.02%, which is better than the accuracy obtained using all of the compared approaches for PV only integrated scenario of the IEEE-123 node distribution system. The precision of class labels obtained from ResNetC is 98.35%, which is also better than the accuracy obtained using all other compared approaches. In other words, the accuracy of correct positive prediction or the accuracy of minority class is also high with ResNetC than any other compared models. Similarly, the sensitivity or recall of the class labels predicted from ResNetC is 98.02%, which is better than the accuracy obtained using other models. In other words, there are a very few number of missed true predictions from ResNetC. The high value of precision and recall together are very important factors for machine learning models. From Table 1, we can see that F_1 score of ResNetC is 98.55%, which indicates high precision and recall. Furthermore, low value of false alarm rate shows that there are very few number of false alarms. Similar inferences can be drawn from other tables (Tables 2–6) as well.

For both IEEE 123- and 240-node distribution systems, it is seen that the results obtained for wind only integrated system obtains better results than that for PV only integrated system. The reason for this is that the investigated deep learning models can better map the input–output relationship between wind speed and wind turbine output power. On the contrary, the complicated relationship between the input and output variables in case of PV system makes it more difficult for the investigated model to perfectly capture the relationship.

Time taken to predict the class labels for each of the provided cases show that once the measurements are available, the proposed work can detect attacks within a few milliseconds, which is suitable for real-time applications.

6. Conclusions

This paper has proposed a deep learning-based multi-label classification approach to detect data falsification attacks in the DG domain. In the proposed work, three types of attack scenarios for modifying the output power of DGs were described: additive, deductive, and a combination of both types of attacks. The proposed work dealt with coordinated attacks that can be launched simultaneously on a large number of DGs in a distribution system. This paper utilizes surrounding environment parameters along with power system measurements, such as DG meter readings and SCADA measurements, to detect coordinated attacks. A coordinated attack vector was launched on the normal DG meter readings to generate attacked data and the respective output class labels. A two branch-based classifier was developed to separately classify the additive and deductive attacks. Deep learning methods such as MLP, CNN, and ResNet were used as classifiers on both of the branches to test their performance for the proposed solution. Several case studies were conducted on the IEEE 123-node distribution system and a 240-node real distribution test system. The results showed that the proposed model can detect coordinated low margin attacks (as low as 5% of actual DG outputs) with a high percentage (99.9% for wind only integrated systems and 99% for PV only integrated systems).

The advantages of the proposed approach are as follows. This work proposes a multi-label classification based approach to detect and distinguish coordinated data falsification attacks in DG domain, specifically PV and wind generators. The proposed approach simplifies the complicated multi-label multi class problem into a simple multi-label problem. The proposed work also helps in accurately determining the nature of attacks: additive or deductive in a DG, which is important since it provides the power system operator with clear information to take further remedial actions.

The major challenge and/or limitation of proposed work is that its performance depends on the availability of a large number of datasets; adequate datasets are required for training and testing of machine learning models in the proposed work.

Source Codes

All of the developed source codes (including unbalanced quasi-static power flow in OpenDSS-MATLAB environment and deep learning in Python) of the proposed solution are publicly available at: https://github.com/nbhusal/Coordinated-cyber-attack-detection-in-Renewable-DG.

CRediT authorship contribution statement

Narayan Bhusal: Conceptualization, Methodology, Data curation, Writing - original draft, Investigation, Validation, Formal analysis. **Mukesh Gautam:** Conceptualization, Methodology, Writing - original draft, Data curation, Formal analysis. **Raj Mani Shukla:** Conceptualization, Methodology, Software, Writing - review & editing. **Mohammed Benidris:** Conceptualization, Methodology, Supervision, Validation, Writing - review & editing. **Shamik Sengupta:** Supervision, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the U.S. National Science Foundation (NSF) under Grant NSF 1847578.

References

[1] IEEE standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces, IEEE Std 1547–2018 (Revision of IEEE Std 1547–2003), 2018; 1–138.
[2] Singh SK, Bose R, Joshi A. Entropy-based electricity theft detection in AMI network. IET Cyber-Phys Syst: Theory Appl 2018;3(2):99–105.
[3] Jindal A, Dua A, Kaur K, Singh M, Kumar N, Mishra S. Decision tree and SVM-based data analytics for theft detection in smart grid. IEEE Trans Industr Inf 2016;12(3):1005–16.
[4] Xiang Y, Wang L, Liu N. Coordinated attacks on electric power systems in a cyber-physical environment. Electric Power Syst Res 2017;149:156–68.
[5] Gunturi SK, Sarkar D. Ensemble machine learning models for the detection of energy theft. Electric Power Syst Res 2021;192:106904.
[6] Pereira J, Saraiva F. Convolutional neural network applied to detect electricity theft: A comparative study on unbalanced data handling techniques. Int J Electrical Power Energy Syst 2021;131:107085.
[7] Zheng Z, Yang Y, Niu X, Dai H, Zhou Y. Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids. IEEE Trans Industr Inf 2018;14(4):1606–15.
[8] Zheng K, Chen Q, Wang Y, Kang C, Xia Q. A novel combined data-driven approach for electricity theft detection. IEEE Trans Industr Inf 2019;15(3):1809–19.
[9] Nabil M, Mahmoud M, Ismail M, Serpedin E. Deep recurrent electricity theft detection in AMI networks with evolutionary hyper-parameter tuning. In: 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData); 2019. p. 1002–8.
[10] Bhattacharjee S, Das SK. Detection and forensics against stealthy data falsification in smart metering infrastructure. IEEE Trans Dependable Secure Comput. 2021.
[11] Lore KG, Shila DM, Ren L. Detecting data integrity attacks on correlated solar farms using multi-layer data driven algorithm. In: 2018 IEEE Conference on Communications and Network Security (CNS); 2018. p. 1–9.
[12] Yuan X, Shi M, Sun Z. Research of electricity stealing identification method for distributed PV based on the least squares approach. In: 2015 5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT), Changsha, China; 2015. p. 2471–474.
[13] Krishna VB, Gunter CA, Sanders WH. Evaluating detectors on optimal attack vectors that enable electricity theft and DER fraud. IEEE J Sel Top Signal Process 2018;12(4):790–805.
[14] Lai C, Jacobs N, Hossain-McKenzie S, Carter C, Cordeiro P, Onunkwo I, et al. Cyber security primer for DER vendors, aggregators, and grid operators. Tech Rep 2017. SAND2017-13113, [Accessed December 2017].
[15] Johnson J. Roadmap for photovoltaic cyber security. Tech. Rep. SAND2017-13262, December 2017.
[16] Ismail M, Shaaban MF, Naidu M, Serpedin E. Deep learning detection of electricity theft cyber-attacks in renewable distributed generation. IEEE Trans Smart Grid 2020;11(4):3428–37.
[17] Funding opportunity announcement (FOA) number: DE-FOA-0002243.
[18] McLaughlin S, Podkuiko D, McDaniel P. Energy theft in the advanced metering infrastructure. In: Rome E, Bloomfield R, editors. Critical Information Infrastructures Security. Berlin Heidelberg, Berlin, Heidelberg: Springer; 2010. p. 176–87.
[19] Jiang R, Lu R, Wang Y, Luo J, Shen C, Shen X. Energy-theft detection issues for advanced metering infrastructure in smart grid. Tsinghua Sci Technol 2014;19(2):105–20.
[20] Koppel T. Lights Out: A Cyberattack, A Nation Unprepared, Surviving the Aftermath. New York: Crown Publishers; 2015.
[21] Qi J, Hahn A, Lu X, Wang J, Liu C. Cybersecurity for distributed energy resources and smart inverters. IET Cyber-Phys Syst: Theory Appl 2016;1(1):28–39.
[22] Mohsenian-Rad A, Leon-Garcia A. Distributed internet-based load altering attacks against smart power grids. IEEE Trans Smart Grid 2011;2(4):667–74.
[23] Bhattacharjee S, Thakur A, Silvestri S, Das SK. Statistical security incident forensics against data falsification in smart grid advanced metering infrastructure. In: Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, CODASPY '17. New York, NY, USA: ACM; 2017. p. 35–45.
[24] (March 2017). [link]. https://www.telegraph.co.uk/news/2017/03/06/smart-energy-meters-giving-readings-seven-times-high-study-finds/.
[25] Herrera F, Charte F, Rivera AJ, del Jesus MJ. Multilabel Classification. Cham: Springer International Publishing; 2016. p. 17–31.
[26] Wang S, Bi S, Zhang YA. Locational detection of false data injection attack in smart grid: a multi-label classification approach. IEEE Internet Things J 2020.
[27] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA; 2016. p. 770–78.
[28] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 2261–9.
[29] Wang Z, Yan W, Oates T. Time series classification from scratch with deep neural networks: A strong baseline. In: 2017 International Joint Conference on Neural Networks (IJCNN); 2017. p. 1578–85.
[30] Chen K, Chen K, Wang Q, He Z, Hu J, He J. Short-term load forecasting with deep residual networks. IEEE Transactions on Smart Grid 2019;10(4):3943–52.
[31] Zhang L, Wang G, Giannakis GB. Real-time power system state estimation and forecasting via deep unrolled neural networks. IEEE Trans Signal Process 2019;67(15):4069–77.
[32] Zhao L, Li M, Meng D, Li X, Zhang Z, Zhuang Y, Tu Z, Wang J. Deep convolutional neural networks with merge-and-run mappings. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. AAAI Press; 2018. p. 3170–6.
[33] Vinayakumar R, Alazab M, Srinivasan S, Pham Q-V, Padannayil SK, Simran K. A visualized botnet detection system based deep learning for the internet of things networks of smart cities. IEEE Trans Ind Appl 2020;56(4):4436–56.
[34] Ravi V, Alazab M, Srinivasan S, Arunachalam A, Soman K. Adversarial defense: DGA-based botnets and DNS homographs detection through integrated deep learning. IEEE Trans Eng Manage 2021.
[35] Bu F, Yuan Y, Wang Z, Dehghanpour K, Kimber A. A time-series distribution test system based on real utility data. In: 2019 North American Power Symposium (NAPS), Wichita, KS, USA, USA; 2019., p. 1–6.
[36] Shaaban MF, Osman AH, Aseeri FM. A multi-objective allocation approach for power quality monitoring devices. IEEE Access 2019;7:40866–77.
[37] Eldery MA, El-Saadany EF, Salama MMA, Vannelli A. A novel power quality monitoring allocation algorithm. IEEE Trans Power Deliv 2006;21(2):768–77.
[38] Asgari A, Firouzjah KG. Optimal PMU placement for power system observability considering network expansion and N-1 contingencies. IET Generat Transmiss Distrib 2018;12(18):4216–24.
[39] Distribution System Analysis Subcommittee, 1992 test feeder cases. Tech. rep., IEEE, PES; 1992. URL http://sites.ieee.org/pestestfeeders/resources/.
[40] Manwell JF, McGowan JG, Rogers AL. Wind Energy Explained:Theory, Design and Application. 2nd ed. Hoboken, NJ, USA: Wiley; 2010.
[41] Masters GM. Renewable and Efficient Electric Power Systems. 2nd ed. Hoboken, NJ, USA: Wiley; 2013.
[42] Vinayakumar R, Alazab M, Soman K, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. IEEE Access 2019;7:41525–50.
[43] Javaid A, Niyaz Q, Sun W, Alam M. A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS); 2016. p. 21–6.
[44] EPRI, Open distribution system simulator. https://smartgrid.epri.com/SimulationTool.aspx.