ISSN: 2834-4928

Research Article

Journal of Electrical Electronics Engineering

Detection of False Data Injection Attacks in Smart-Grid Systems: Benchmarking Deep Learning Techniques

Lukumba Phiri[1]*, Simon Tembo [2]

[1,2]*Department of Electrical and Electronic Engineering, School of Engineering, University of Zambia, Lusaka, Zambia*

***Corresponding Author**
Lukumba Phiri, Department of Electrical and Electronic Engineering, School of Engineering, University of Zambia, Lusaka, Zambia

Submitted: 21 Feb 2023; **Accepted:** 28 Feb 2023; **Published:** 03 Mar 2023

Citation: Phiri L, Tembo S. (2023). Detection of False Data Injection Attacks in Smart-Grid Systems: Benchmarking Deep Learning Techniques. *J Electrical Electron Eng*, 2(1), 41-49.

Abstract

In essence, smart grids are electrical networks that transmit and distribute electricity in a reliable, effective manner using information and communication technology (ICT). Trust and security are of the utmost importance. False data injection (FDI) attacks are one of the most serious new security problems, and they can drastically raise the price of the energy distribution process. However, rather than smart grid infrastructures, the majority of current research focuses on FDI defenses for conventional electricity networks. By utilizing spatial-temporal correlations between grid components, we create an effective and real-time technique to identify FDI attacks in smart grids called a deep learning framework. We show that the suggested method offers an accurate and dependable solution using realistic simulations based on the smart grid compared to the benchmarked techniques.

Keywords: Smart Grids, Bad Data Detection, State Vector Estimation, Deep Learning, Ieee Test Bus Systems, Matpower, Keras With Tensorflow

Introduction

The needs of the present cannot be satisfied by the conventional power grid infrastructure from the 20th century. New technologies are constantly being implemented, such as electric vehicles (EVs), intelligent inverters, and generators powered by renewable energy [1]. They add bi-lateral power flow, alter power system operation paradigms, and produce a dynamic operational structure that was not initially intended [2]. Power systems now have additional measurement, communication, and control capabilities to address these problems. This allows for the acquisition of more precise information regarding the grid's current state and the making of decisions in quasi-real time [3]. The Smart Grid (SG) is the aggregate name for this contemporary power system configuration. The term "SG idea" is used to describe a variety of things, including "A network where all consumers may reach efficient, affordable, accessible, and reliable energy by using control and communication technology" [4]. As an alternative, SG is a system that is adaptable, dependable, and interactive, and enables the integration and optimization of renewable energy sources [5, 6].

The National Institute of Standards and Technology (NIST) provides a broad perspective and categorizes SGs in addition to these categories. Additionally, the application characteristics and SG infrastructure requirements are separated into many levels [7].

1. Application
2. Security
3. Communication
4. Control of Power
5. Power system

It is expected that some cybersecurity flaws will become more common than others given the large geographic range of SGs and the volume of devices they host. All businesses with energy-providing authorities concur that security breaches in such essential infrastructure will have serious repercussions [8]. SG can be viewed as an electrical system that employs cyber secure information and communication technologies, according to [9]. The system aims to combine a computational intelligence system with energy transmission, generation, and distribution substations that is safe, dependable, and effective. The requirements of SG communication systems are tough for current cybersecurity solutions to satisfy. It is clear from recent research that traditional cybersecurity techniques and algorithms have typically been explored, and there have been independent studies on power and communication concerning cyber dangers. Traditional risks are now taken into account in risk assessments if there are cybersecurity risks in vital systems like the communication infrastructure for the power system. However, the security of SG communication systems is a topic that is still in its infancy; there aren't many academic or experimental investigations available [10].

Since Liu et al. suggested that an attacker can use FDIAs against state estimation to avoid being discovered by the estimation residual-based bad data detection (BDD) methods, a lot of studies are being done to investigate the building and defensive mechanism of FDIAs [11]. While current work in AC transmission systems has emerged as a result of their reactively correct analytical models, some research on FDIA construction has been reported in various application situations in DC power systems. A thorough study of FDIA construction techniques was carried out by [12, 14]. On the other hand, numerous results utilizing various statistical and probabilistic techniques, such as sparse optimization claimed to fight against FDIA in DC system state estimation. However, because these techniques depend on knowledge of measurement data distributions and system operation states, they may become obsolete and ineffectual if these prerequisites change [15, 16].

The rapid development of advanced metering infrastructure, which produces a great amount of data, has led to an increase in the application of machine learning and data-driven methodologies to improve the operation of power systems [17, 18]. This is a result of their strong capacity for information extraction and adaptable extension. To discover FDIAs in transmission systems, several learning-based techniques, including deep belief networks (DBN) support vector machines (SVM) and deep neural networks (DNN) have been developed [19-22]. However, AC power system models are extensively used by real-world utilities, and algorithms that are performed on DC power systems, including overlook the sophistication of unobservable attacks or the complexity of power system complexity. When dealing with unobservable attacks in AC transmission systems uses wavelet transform and DNN approaches to address this shortcoming by examining the state dynamics to capture the discrepancy between abnormal and normal observations. The method in nevertheless imposes a heavy computing cost and calls for measurements with labels from continuous samplings that may not be accessible in actual operation. It is crucial to remember that the majority of supervised machine learning techniques for detecting FDIAs, such as those in evaluate anomalous data that differs in some manner from the labeled data used for training.

The datasets obtained from actual cyber-physical systems are only partially labeled due to high labeling expenses [23]. In practice, unlabeled data are typically much larger in scale than labeled data, and the supervised learning approach hardly ever employs this enormous unlabeled data. This absence results in severe data loss, which ultimately leads to the failure of the process.

The main contributions of the proposed method are listed:

• This paper presents a novel learning-based FDIA detection algorithm for unobservable attacks or outliers that bypass the conventional BDD mechanism. This method enables the detection of these attacks within milliseconds and thus can be implemented online.

• In contrast to supervised learning, the proposed semisupervised detection method only requires a limited number of labeled data to detect the attacked measurement data. Specifically, with as few as 1,000 labeled training data, this method self-learns with an accurate detection ability.

• The proposed algorithm is fully data-driven and thus extensible and does not depend on the information of network topology and parameters in distribution systems.

• We benched the proposed deep learning framework called DLLD with other FDIA detection techniques.

In other words, the proposed method can use the few to detect the many, thanks to the generative models.

The remainder of this article is organized as follows. We briefly introduce the conventional state estimation method and its vulnerability against FDIA in Section II. In Section III, we illustrate the architecture and implementation issues of the proposed FDIA locational detection mechanism. The simulation results with parameter sensitivity are presented in Section IV. Finally, this article is concluded in Section V.

State Estimation and FDIA

The master's program known as the energy management system (EMS) is at the core of the power system. EMS is a high-performance critical application that oversees all of the electric grid monitoring control and optimization activities it receives redundant readings from numerous phasor measurement units (PMUs) and SCADA devices field instrument transformers are being sampled for current, voltage, and power flow. When compared to standard SCADA devices the PMUs sample at a rate of 30/60/120/240 messages per second with a substantially higher degree of accuracy and are thus widely used by utilities to improve real-time monitoring [24, 25]. A local phasor data concentrator (PDC) at the substation level receives data packets from PMUs and synchronizes and aligns them. Before sending a report to a data concentrator at the main control center, regional PDCs collect and assemble data from station-level PDCs. Figure 1 displays the architecture of the PMU-PDC.

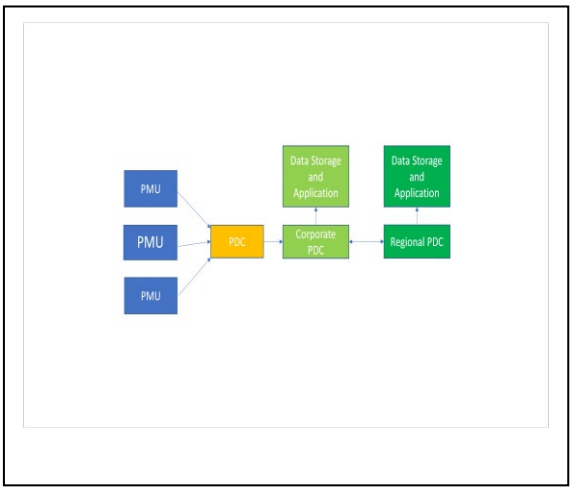

Figure 1: The architecture of the PMU-PDC

Power System State Estimation

DC State Estimation

The linear DC state estimate using only traditional SCADA meters is built on the following linear measurement function [26]:

$$\boldsymbol{z} = \boldsymbol{Hx} + \boldsymbol{e} \quad (1)$$

where e is a m x 1 vector of random Gaussian errors, z is a m x 1 vector of measurements, H is the m x n Jacobian matrix, x is then a n x 1 vector of state variables, and m,n is the total number of measurements and states, respectively. The following presumptions are true for DC state estimation: (1) the voltage magnitudes at all buses in the network are assumed to be constant and equal to 1 per unit (p.u.); (2) the shunt susceptances and series resistances of transmission lines are neglected; (3) the bus angle differences between two buses are thought to be very small; (4) reactive power is entirely neglected, and (5) state variables only consist of bus voltage angles.

The measurement residual arising from the difference between measured and estimated states is defined as,

$$\boldsymbol{r = z - Hx} \tag{2}$$

The state variables can be estimated by minimizing the objective function J,

$$\boldsymbol{J(x) = (z - Hx)^T R^{-1}(z - Hx)} \tag{3}$$

Straightforwardly for DC state estimation, the states are estimated as,

$$\boldsymbol{\hat{x} = (H^T R^{-1} H)^{-1} H^T R^{-1} z} \tag{4}$$

AC State Estimation

The oversimplified DC state estimation model might not be suitable for real-time power system state estimation since measurements in power systems are related to their states by a non-linear function. The link between the state variables and the states can be stated as (5) for AC state estimation[26].

$$\boldsymbol{z = h(x) + e} \tag{5}$$

where z is an m x 1 vector of measurements from SCADA meters and PMUs, h is a set of non-linear power flow functions relating measurements to state variables, x is an n x 1 vector of state variables, e is an m x 1 vector of random Gaussian errors, and m, n is a total number of measurements and states respectively [26].

The non-linear functions h(x) which relate the measurement to the state variables comprise active and reactive power injections at the bus, active and reactive power flow in transmission lines, and branch real and imaginary currents. The real and reactive power injection at bus m is,

$$\mathbf{P}_m = V_m \sum_n V_n \left(\mathbf{g}_{mn} \cos \delta_{mn} + \mathbf{B}_{mn} \sin \delta_{mn} \right. \tag{6}$$

$$\mathbf{Q}_m = V_m \sum_n V_n \left(\mathbf{g}_{mn} \sin \delta_{mn} - \mathbf{B}_{mn} \cos \delta_{mn} \right. \tag{7}$$

The real and reactive power flow from bus m to bus n is,

$$\mathbf{P}_{mn} = V^2{}_m \, \mathbf{g}_{mn} - V_m V_n \left[g_{mn} \cos(\delta_m - \delta_n) + \mathbf{b}_{mn} \sin(\delta_m - \delta_n) \right. \tag{8}$$

$$\mathbf{Q}_{mn} = -V^2{}_m \, \mathbf{b}_{mn} - V_m V_n \left[g_{mn} \sin(\delta_m - \delta_n) - \mathbf{b}_{mn} \cos(\delta_m - \delta_n) \right] \tag{9}$$

The real and imaginary branch current between bus m and bus n is,

$$\mathbf{I}_{mn},\text{real} = V_m \left[\mathbf{g}_{mn} \cos \delta_m - \mathbf{b}_{ij} \sin \delta_m \right] - V_n \left[\mathbf{g}_{mn} \cos \delta_n - \mathbf{b}_{ij} \sin \delta_n \right] \tag{10}$$

$$\mathbf{I}_{mn},\ \text{imag} = V_m \left[\mathbf{b}_{mn} \cos \delta_m + \mathbf{g}_{ij} \sin \delta_m \right] - V_n \left[\mathbf{b}_{mn} \cos \delta_n - \mathbf{g}_{ij} \sin \delta_n \right] \tag{11}$$

The weighted least squares method is used to minimize the measurement residuals to accurately estimate the states with the objective function defined as [27].

$$\boldsymbol{J(x) = (z - h(x))^T R^{-1}(z - h(x))} \tag{12}$$

where R is the measurement error covariance matrix. The estimates of the state are found by an iterative process like the Newton-Raphson method,

$$\boldsymbol{\Delta \hat{x} = (H^T R^{-1} H)^{-1} H^T R^{-1}(z - h(x))} \tag{13}$$

$$\boldsymbol{\hat{x}_i + 1 = \hat{x}_i + \Delta \ \hat{x}_i} \tag{14}$$

where H is the measurement Jacobian matrix and is defined as $H = (\partial h(x))/\partial x$,

In matrix H, the first and the sixth columns are related to bus voltage magnitude and angle-system states which are directly measured by the PMUs, and hence have an identity relation with the estimated states.

Bad Data Detection

Bad PMU and SCADA data can naturally occur as the result of instrumentation errors, thermal degradation of equipment, or random electrical noise. One of the most popular techniques for detecting erroneous measurements is comparing the L2 (norm) of the measurement residuals to a detection threshold τ. For DC state estimation, no bad data is detected when[27].

$$\boldsymbol{\|z - H\hat{x}\| < \tau} \tag{16}$$

Similarly, for AC state estimation, no bad data is detected when,

$$\boldsymbol{\|z - h(\hat{x})\| < \tau} \tag{17}$$

In general, the threshold τ is determined and obtained from the cumulative chi-square distribution for m - n degrees of freedom [26]. Residuals that satisfy (16) and (17) are assumed to be free of bad data while those that fail to satisfy this condition are excluded from the data set for subsequent calculations. The discarded bad data is often substituted by pseudo-measurements obtained from historical values to ensure that SE converges [27].

False Data Injection Attack

The objective of FDIA is to mislead the system operator into considering a compromised state estimate $\hat{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{c}$ as a valid estimation, where c ≠ 0 is the deviation of the power system state. To achieve this, an attacker changes the received measurements at the control center to $\hat{\boldsymbol{z}} = \boldsymbol{z} + \boldsymbol{a}$ where a = $(\mathbf{a}_1,\mathbf{a}_2,\ldots,\mathbf{a}_n)^T$ is the compromised attack vector. Then, the observation model can be described as

$$\hat{\boldsymbol{z}} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{e} + \boldsymbol{a} \tag{18}$$

In general, an unstructured a is likely to be identified by the traditional BDD (16). To circumvent the BDD mechanism, the attack vector should be structured, such as a = Hc. In such cases, the $\boldsymbol{l}_2$ -norm of the residual is unchanged

$$\left\|\hat{\boldsymbol{Z}} - \boldsymbol{H}\hat{\boldsymbol{x}}\right\| = \|\boldsymbol{z} + \boldsymbol{a} - \boldsymbol{H}(\boldsymbol{x} + \boldsymbol{c})\| = \|\boldsymbol{z} - \boldsymbol{H}\boldsymbol{x}\| \tag{19}$$

and thus the attacker can bypass the BDD. Accordingly, the power system operator would mistake x+c for a valid estimate, and thus an error vector c is introduced.

In this article, we develop a new data-driven mechanism that can detect the location of FDIA in a SCADA system. It is formulated as a multilabel classification problem that determines whether each meter measurement is compromised. The problem is formulated and solved in Section III.

Proposed Detection Mechanism

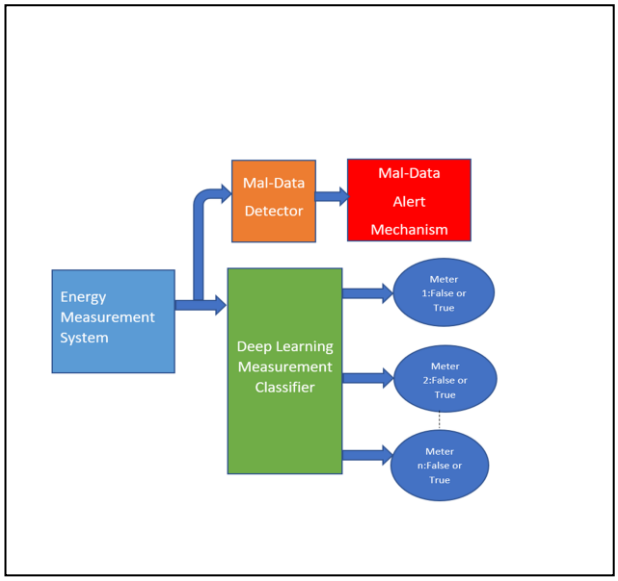

Figure 2: The Proposed Detection Mechanism

To identify unobservable FDIAs in three-phase distribution systems, this section suggests a deep learning detection method; Fig.2 gives a summary of the suggested detection mechanism. We formulate the detection problem as a binary classification problem with the detection indicator **ω** for unobservable FDIAs or outliers represented by the attack vector a.

$$\boldsymbol{\omega} = \begin{cases} \mathbf{0}\ \boldsymbol{if}\ \boldsymbol{a} = \mathbf{0} \\ \mathbf{1}\ \boldsymbol{if}\ \boldsymbol{a} \neq \mathbf{0} \end{cases} \tag{20}$$

The measurement vector z, which includes three-phase voltages, currents, and powers, is gathered as the input of the deep learning measurement classifier-based detection algorithm. Only a small portion of them are labeled with **ω** = 0 or 1.

Locational Detection

Mathematically, classifying the entire measurement vector, x, into the categories of 1) exists or 2) not is equivalent to detecting the presence of FDIA. From the standpoint of machine learning, this is a single-label classification issue. To pinpoint the attack's location, we must divide each component of the measurement vector, xi, into two groups. The intricacy and broad applicability of multilabel classification continue to be of significant interest to researchers, despite the decade-long success of deep learning algorithms in single-label classification.

To solve the issue of BDD circumvention, we painstakingly created the CNN structure in Fig. 3 to extract and characterize the relevant data information and generate successful multi-label classification results. In addition, we will assess how well our numerical trials compare to the other techniques for label approaches.

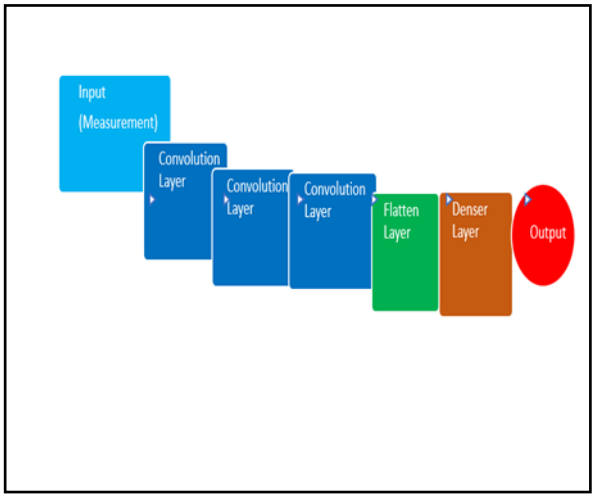

Figure 3: The Convolution Neural Net-work Achitecture

Proposed Mechanism

Fig. 2 shows the suggested FDIA locational detection mechanism. The suggested system collects data from a series of discrete sampling time instances, i.e. the times when the traditional state estimation occurs. This demonstrates that the proposed mechanism does not rely on any previous statistical assumptions, together with the fact that the CNN classifier's training procedure simply needs measurements and ground-truth labels (e.g., H). The real-time measurement input data first passes via

the BDD detector at sampling time instance t. As stated in (17), BDD determines the measurement residuals $l_2$norm and compares it to a predefined threshold τ to determine the quality of the measurement data. If $R \geq \tau^2$, BDD classifies the present meter as compromised or noisy. Because of their high residual values, sampling and communication mistakes as well as potential unstructured FDIA can be efficiently detected in this way [28]. A CNN-based multilabel classifier will identify the presence and position of structured FDIAs by examining the data's inconsistency and co-occurrence dependency if the measurement data pass the BDD.

The CNN theory proposes using CNN to separate and examine the high-dimensional contextual highlights of the FDIA.

1). Data: We refer to the data (also known as the estimations), ground truth labels (also known as meter classes), and yields (also known as CNN period t classifications) as $m^t=(m_1{}^t,\ldots,m_n{}^t)$, $\omega^t=(\omega_1{}^t,\ldots,\omega_n{}^t)$, and, $\omega^t=(\omega_1{}^t,\ldots,\omega_n{}^t)$ respectively. For instance, because there are 180 measurements inside the IEEE 118-bus system, thus in our simulation settings in section IV, the input and output data dimensions for the IEEE 118-bus system adhere to the actual parameters. According to the following rule, the ground-truth label of meter i at time t is determined:

$$\omega_i^t = \begin{cases} 0 \; if \; a = 0 \\ 1 \; if \; a \neq 0 \end{cases} \qquad (21)$$

CNN's output is a continuous range of values between 0 and 1. In line with this, the classifier establishes a discrimination threshold to categorize the outputs as either 0 or 1. One can change the discrimination threshold to make it higher or lower the responsiveness to application variables. The discriminating threshold in this article is set to 0.5, as is customary unless otherwise stated.

2. Architecture: Figure 3 depicts the deep CNN's architectural design for FDIA locational identification. There are numerous convolutional layers, a flattening layer, a fully linked hidden layer, and an output layer in addition to the input layer,n integers reflecting the n measurements at each time occurrence are fed into the input layer. Windows in the input layer are subjected to each filter in the first convolutional layer, which produces features by convolution, batch normalization, and nonlinear transformation with the rectified linear unit (ReLU) activation function [29]. The first convolutional layer's feature mappings c_1, j were created from the input data z and are expressed as:

$$c_1, j = ReLU(\omega * h_1, j + b_1, j) \qquad (22)$$

Here, h1,j is the jth convolution kernel, which is essentially a 1-D filter, and b1,j is the proper scalar bias. The convolution output is all given a deep learning representation known as a scalar bias, or b1,j, in equation (22). The equation states that the convolution operation is denoted by * and the output at point i is defined as

$$\sum_{k=1}^{l_{1,j}} (h_{1,j})[i] x(\omega)[i - k + \frac{i_{1,j}}{2}] \qquad (23)$$

Here, $l1,j$, and × denote the length of the filter $\boldsymbol{h}1,j$, and the inner product operation, respectively. The hidden features generated by filters in the $(q - 1)$ convolutional layer are then used as the input to the qth convolutional layer and processed similarly. The output can be written as

$$c_q, j = ReLU(c_{q-1} * h_q, j + b_q, j) \qquad (24)$$

where cq,j is the jth feature map at the qth convolutional layer. The number of filters in each layer and the depth of convolutional layers are hyperparameters, which will be further discussed in the simulation section. The extracted features learned by the last convolutional layer, i.e., the q^{max} th convolutional layer, are merged into one single vector in the flattened layer and fed into a fully connected hidden layer (also known as the dense layer) with the activation function ReLU. That is

$$c_f, j = ReLU(w_f * c_q^{max} \; x \; b_f) \qquad (25)$$

where cFj, $\boldsymbol{w}F$, and $\boldsymbol{b}F$ denote the feature maps, weights, and biases of the flattened layer, respectively. The nodes in the dense layer are also fully connected to n nodes in the output layer. The sigmoid function is applied to the nodes in the output layer to classify the type of each measurement. For meter j at time t, the final multilabel classification result ω_j^t is

$$\hat{\omega}_1^t = signoid(w_D x \; c_F + b_D) \qquad (26)$$

where w_D and b_D denote the weights and biases of the dense layer, respectively.

First point: In addition to the convolutional layers, pooling and dropout layers are significant elements of standard CNN architectures. However, for the reasons listed below, they are absent from our design. First, downsampling high-dimensional computations, such as 2-D and 3-D convolution computations, typically uses pooling layers. All of the convolutional layers in our challenge are 1-D convolutional layers, whose processing on a GPU is highly effective. Second, one of the primary methods used to achieve nonlinear mapping in deep CNN has historically been layer pooling.

But the well-liked ReLU activation function also adds nonlinearity to deep models. Because pooling layers may reject essential details, rendering up pooling layers occasionally results in even higher performance [30]. Third, a common method for preventing overfitting is a dropout. The suggested DLLD has already used the mini-batch overfitting control mechanism, which purposefully adds enough noise to each gradient update.

Indeed, we investigated the effectiveness of pooling and dropout and discovered that there is no performance benefit.

Training

Before classifying the measurements with the proposed FDIA locational detection system, we must first tune the learning pa-

rameters, specifically the filters h, weights w, and biases b, in each layer. Training is the process of tuning parameters to identify the best parameters that match the input and output in the training data.

- Mini-Batch and Cross-Validation: We train the network using a mini-batch gradient descent technique to speed convergence rates and avoid overfitting. Each mini-batch in our simulations has 200 data instances. The gradient is computed for each iteration using a fixed number of training samples, or a mini-batch, that are randomly chosen from the training set.
- We split the data from each batch into two sets, as is typical in machine learning: a training set that contains 7/10 of the data and a validation set that contains 3/10 of the data. The Adam optimizer is then used to do fitting with a learning rate of 0.001 and patience of 5.
- Loss Function: To determine the best set of learning parameters, we incorporate a loss function that calculates the difference between each mini-actual batch's output and ground truth output. The proposed CNN's loss function is selected as the cross-entropy function to incorporate multi-label classification into our system. More specifically, the cross entropy loss function over a mini-batch $\theta = \{t_1,\ldots,t_{200}\}$ is defined as follows:

Cross-entropy (θ)

$$\sum_{t\in\theta} -\frac{i}{n}\sum_{i=j}^{n}\left(\hat{\omega}_i^t \, log\left(\hat{\omega}_t^t\right) + \left(1-\hat{\omega}_1^t \, log\left(1-\hat{\omega}_t^t\right)\right)\right)$$

We can use the Adam [29] optimizer to determine the best settings given a mini-batch if the loss function is explicitly stated.

Experiments

In this section, we briefly describe the design of the software, and the architecture of the deep learning model employed to build a classification model to identify secure data and attacked data.

Datasets

In 118-bus power grids, the suggested FDIA locational detector is evaluated in this section. You may get the grid topologies from MATPOWER [31]. These are the power topologies, summed up as follows:
IEEE118-bus system:

• The number of transmission lines and buses is 186 lines and 118 buses, respectively.
• The number of total meters measurements is 180, of which 110 are flow measurements and 70 are injected measurements.
• The training and testing datasets are adopted from [32] and can be summarized as follows:
• The network topology is used to index meter readings. First, starting with k = 1, the line flow meters are indexed as follows:
• The policy goes back to the first stage if k > 118, at which point the indexing procedure is ended, and the unindexed meters connecting bus k are indexed and set to k = k + 1. The injection meters are then labeled using the bus index in ascending order after continuing the index from line meters [32].
• By artificially boosting the loads on each bus, 110,000 sets of pristine data are generated by extending the real-world data. The generated loads are normally distributed, with a mean equal to the base load and a standard deviation equal to one-sixth of the base load's size [33, 34].
• Ten thousand sets of loads are randomly chosen to implement the FDIA:
• For each attack, a random selection of target state variables to compromise is made. Target state variables in the 118-bus power system have a discrete uniform (2, 10) distribution, in the power system.
• The transmission line impedance is set following and the injected data's L2-norm (the anticipated value of the attack vector's Euclidean norm) ranges from 1 to 5. Both compromised and uncompromised data were added with a noise standard deviation of 0.2 [35].
• The min-cut algorithm in is used to construct a stealthy FDIA for each collection of loads and their unique target state variables.
• Finally, to take into consideration the noise in measurement, a random Gaussian noise with a standard deviation of 0.2 was added in both compromised and uncompromised data.
• Following the creation of the training data, the aforementioned procedure is carried out again ten times to produce ten separate sets of testing data, which naturally introduces differences in validation.

Training and Testing Datasets

Under each level of attack, the dataset is prepared as follows [32]

• For training, input measurements and training labels are generated with a dimension of 110,000 x B. The training data are composed of 100, 000 samples with no attack vector and 10,000 instances under attack.
• For testing, a testing set is generated with a dimension of 10,000 x B for measurements and labels. Input measurements are composed of 5000 uncompromised samples and 5000 compromised samples [33]. Here, B represents the number of meter measurements of the IEEE test case, i.e., 180 for the IEEE 118-bus System. Over all of the test datasets, the results of all trials have been averaged.

Implementation Details

The Proposed Approach LSTM is trained using the Keras package with Tensorflow as the backend[37] using two filters, each with a kernel size of 5 x 1 and 3 x 1, causal padding, and a RELU activation function, then it is classified using multiple labels using a layer that is added after the multi-label classification layer [36]. Additionally, with a 100-step period, validation happens every 100 steps. There will be 100 batches in total. With an initial learning rate of 0.001 and patience of 5, the Adam optimizer is used to fit the data. The loss function for prediction is the custom cross-entropy. We contrast the suggested method with cutting-edge techniques, such as support vector machines (SVM), light gradient boosting machines (LightGBM), and identification methods based on deep learning-based identification (DLBI) [19].

• DLBI: To extract high-dimensional temporal information, He et al.[19] presented a conditional deep belief network (CDBN) architecture. By examining the temporal attack patterns exhibited by the real-time measurement data from the geographically dispersed meters, the CDBN can identify the FDIA.
• SVM: The maximum margin classifier SVM creates one or more hyperplanes in a high-dimensional space. Since achieving top performance in various categorization issues (such as text spam and photos) in the 1990s, it has found widespread use.
• Microsoft has made available LightGBM a gradient-boosting framework that employs methods of tree-based learning. Many of the top machine learning competition solutions used LightGBM [39].
• To fine-tune the hyperparameters, we select the model that provides the greatest F1-Score to the validation data (such as the number of convolutional layers and filters).

Locational Detection Performance

Performance Evaluation Metrics

For the accuracy and recall of the projected outcomes in our simulation, we use labels and the F1-score as performance measures. The following is a description of the precision and recall:

$$\text{Precision} = \frac{True\ Positive\ (TP)}{True\ Positive + False\ Positive\ (TP+FP)} \quad (28)$$

$$\text{Recall} = \frac{True\ Positive\ (TP)}{True\ Positive+False\ Negative\ (TP+FN)} \quad (29)$$

where TP, FP, and FN are, respectively, the likelihood that the detector will classify a site with compromised meters as compromised, a location with uncompromised meters as compromised, and a location with uncompromised meters as uncompromised [32].

$$\text{F1-Score} = \frac{2\ x\ Precision\ x\ Recall}{Precision+Recall} \quad (30)$$

We first assess the proposed technique with the injection data l_2 norm at 2 and the measurement noise's standard deviation at 0.2. We contrast the suggested mechanism not just with cutting-edge approaches like SVM and LightGBM, but also with a variant in which multi-labels (multilayer perceptron-MLPs) are used in place of CNNs in our location and detection process. This leads to the names DLLD and MLP-DLLD for the suggested method and MLP alternative, respectively. In particular, the MLP's hidden layer count ranges from 2 to 6, and the number of units is chosen using the F1-Score with the highest value. We employ identical data sets for the training and testing phases of all four algorithms to ensure a fair comparison.

IEEE 118 – Bus System

This section depicts the performance metrics defined above.

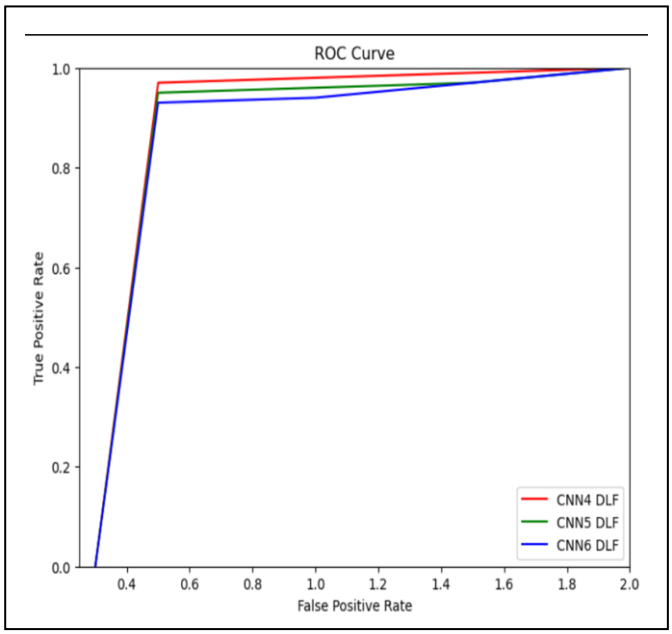

Figure 4: ROC Curve For The Proposed Mechanism

The outputs of the CNN are continuous between [0,1] and are quantized to 0 or 1 by a discriminating threshold. We set the discrimination threshold in figure 4 above at 0.4. The tradeoff between TPR and FPR is often determined by the value of the threshold. A lower threshold specifically causes a greater TPR and a lower FPR. In Fig. 4, where the FPR against TPR is plotted when the threshold ranges from 0 to 1, we examine the tradeoff. The area under the ROC (AUC) is frequently used as a performance assessment of the discriminatory capability to show relative tradeoffs between TPR and FPR [38]. The area between the FPR, TPR, x-axis, and y-axis is what is meant by AUC in this context. AUC close to 1 indicates a good model, which has a high level of separability. The model assumes that a 1 is a 1 and a 0 is a 0. A model predicates 0s as 1s and 1s as 0s when its AUC is close to 0. The proposed mechanism's AUC is close to 1, which illustrates its outstanding discriminatory capacity, as seen in the figure.

Robustness

In Fig. 5, we assess the suggested mechanism's resistance to the attacker's aggression and the noise present in the data-collecting environment. We specifically assess the suggested mechanism in the manner described below.

- Aggression: We altered the injection's L2 norm from 1 to 5, while fixing the standard deviation to be 0.2.
- Noise: The injection's L2 norm is fixed at 2, and the standard deviation ranges from 0.1 to 0.5.

Presence-Detection Performance

We take a step back and look into how effective the suggested approach is at spotting the presence of attacks. Specifically, if $\omega_i^t = 0$, for all i = 1,..., n, we consider the power system to be secure or that no attacks have occurred. Otherwise, it is assumed that the power system has been infiltrated or that there are ongoing attacks. We examine the suggested mechanism's FDIA presence-detection performance in Fig. 6. We compare the detection

accuracy in particular to the following two benchmarks:

1. SVM and
2. DLBI. We only plot the accuracy in the IEEE 118-bus system for simplicity.

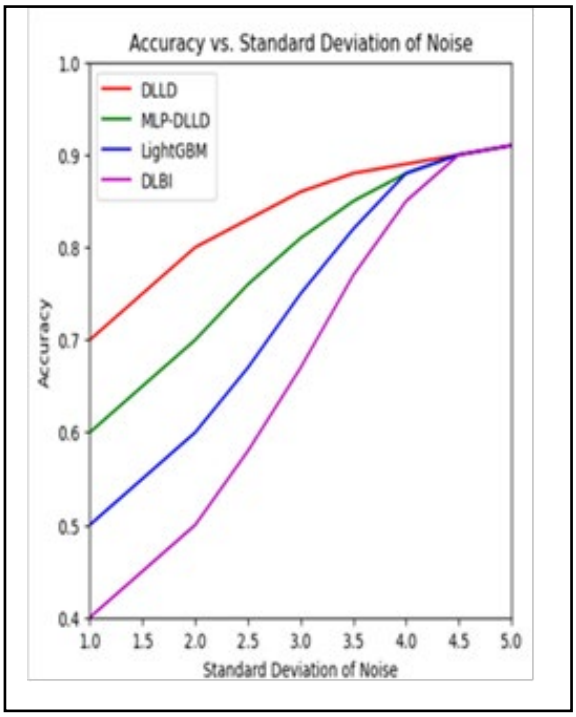

Figure 5: Accuracy versus standard deviation of noise

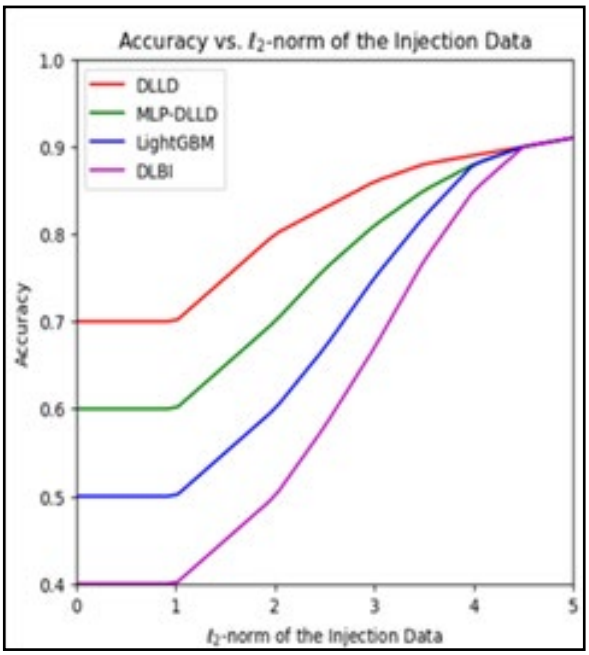

Figure 6: Accuracy versus l2-norm injection data

We compare the detection accuracy attained by DLBI, SVM, MLP-DLLD, and DLLD in Fig. 5 and Fig.6. Overall, the suggested detection approach obtains the maximum detection accuracy when compared to DLBI and SVM algorithms. Additionally, we can see that the detection accuracy of the DLBI and SVM approaches declines as the noise level rises, which is consistent with the finding in Fig. 6.

The suggested approach accomplishes the best detection accuracy, as expected. The accuracy of detection for all four approaches rises as the noise standard deviation does. Before we conclude this section, we want to underline that the suggested multilabel classification method increases the accuracy of presence detection even if it is intended to find FDIA locations. This is so that multilabel classification can account for the meter measurements' co-occurrence dependency and inconsistency.

Conclusion

In this article, we have developed a BDD-CNN architecture as a multilabel classifier and structured the locational detection problem of FDIA as a multilabel classification problem. Real-time measurement data quality is estimated using the standard BDD detector, which is also used to filter out low-quality data. The CNN will record the co-occurrence dependency and inconsistent behavior that FDIA has established. The mechanism is cost-friendly in that it is built on the existing BDD, requires no modification of the current BDD system, and is model-free in that the architecture is independent of any assumed attack model. Additionally, the detection process runs in just a few hundred microseconds on a standard home computer. To show the viability, we have also conducted in-depth simulations in the IEEE 118-bus power systems.

In particular, we have demonstrated that, in a variety of noise and attack settings, DLLD can carry out locational detection for the entire bus system. We have also shown that the presence-detection accuracy may be further enhanced using multilabel classification formulation, and as a result, the resulting presence-detection accuracy is superior to that of the leading-edge benchmarks.

References

1. Aleem, S. A., Hussain, S. S., & Ustun, T. S. (2020). A review of strategies to increase PV pene-tration level in smart grids. Energies, 13(3), 636.
2. Ustun, T. S., & Ayyubi, S. (2019). Automated network topology extraction based on graph theory for distributed microgrid protection in dynamic power systems. Electronics, 8(6), 655.
3. Ustun, T. S., Farooq, S. M., & Hussain, S. S. (2020). Implementing secure routable GOOSE and SV messages based on IEC 61850-90-5. IEEE Access, 8, 26162-26171.
4. Fan, Z., Kulkarni, P., Gormus, S., Efthymiou, C., Kalogridis, G., Sooriyabandara, M., & Chin, W. H. (2011). Smart grid communications: Overview of research challenges, solutions, and stand-ardization activities. IEEE Communications Surveys & Tutorials, 15(1), 21-38.
5. Abdullah, A. A., & Hassan, T. M. (2022). Smart grid (SG) properties and challenges: an overview. Discover Energy, 2(1), 8.
6. Atmaja, T. D., Andriani, D., & Darussalam, R. (2019). Smart Grid communication applications: measurement equipment and networks architecture for data and energy flow. Journal of Mechatronics, Electrical Power, and Vehicular Technology, 10(2), 73-84.
7. Faquir, D., Chouliaras, N., Sofia, V., Olga, K., & Maglaras, L. (2021). Cybersecurity in smart grids, challenges and solutions. AIMS Electronics and Electrical Engineering,

5(1), 24-37. L. P. Costantini, A. Bochman, A. Gopstein, D. Saebeler, A. Raffety, and D. S. Byrnett. (2022). "Understanding Cybersecurity for the Smart Grid: Questions for Utilities," 2020, Accessed: Dec. 30, 2022.
8. Fang, X., Misra, S., Xue, G., & Yang, D. (2011). Smart grid—The new and improved power grid: A survey. IEEE communications surveys & tutorials, 14(4), 944-980.
9. Shafiullah, M., Refat, A. M., Haque, M. E., Chowdhury, D. M. H., Hossain, M. S., Alharbi, A. G., ... & Hossain, S. (2022). Review of Recent Developments in Microgrid Energy Management Strategies. Sustainability, 14(22), 14794.
10. Liu, Y., Ning, P., & Reiter, M. K. (2011). False data injection attacks against state estimation in electric power grids. ACM Transactions on Information and System Security (TISSEC), 14(1), 1-33.
11. Chaojun, G., Jirutitijaroen, P., & Motani, M. (2015). Detecting false data injection attacks in AC state estimation. IEEE Transactions on Smart Grid, 6(5), 2476-2483.
12. Liu, X., & Li, Z. (2017). False data attacks against AC state estimation with incomplete network information. IEEE Transactions on smart grid, 8(5), 2239-2248.
13. Liang, G., Zhao, J., Luo, F., Weller, S. R., & Dong, Z. Y. (2017). A review of false data injection attacks against modern power systems. IEEE Transactions on Smart Grid, 8(4), 1630-1638.
14. Liu, L., Esmalifalak, M., Ding, Q., Emesih, V. A., & Han, Z. (2014). Detecting false data injection attacks on power grid by sparse optimization. IEEE Transactions on Smart Grid, 5(2), 612-621.
15. Manandhar, K., Cao, X., Hu, F., & Liu, Y. (2014). Detection of faults and attacks including false data injection attack in smart grid using Kalman filter. IEEE transactions on control of network systems, 1(4), 370-379.
16. Cui, M., Khodayar, M., Chen, C., Wang, X., Zhang, Y., & Khodayar, M. E. (2019). Deep learn-ing-based time-varying parameter identification for system-wide load modeling. IEEE Transac-tions on Smart Grid, 10(6), 6102-6114.
17. Lin, Y., & Wang, J. (2020). Probabilistic deep autoencoder for power system measurement outlier detection and reconstruction. IEEE Transactions on Smart Grid, 11(2), 1796-1798.
18. He, Y., Mendis, G. J., & Wei, J. (2017). Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism. IEEE Transactions on Smart Grid, 8(5), 2505-2516.
19. Foroutan, S. A., & Salmasi, F. R. (2017). Detection of false data injection attacks against state estimation in smart grids based on a mixture Gaussian distribution learning method. IET Cyber-Physical Systems: Theory & Applications, 2(4), 161-171.
20. M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni, and H. V. Poor. (2016). "Machine Learning Methods for Attack Detection in the Smart Grid," IEEE Trans Neural Netw Learn Syst, vol. 27, no. 8, pp. 1773–1786, Aug.
21. James, J. Q., Hou, Y., & Li, V. O. (2018). Online false data injection attack detection with wavelet transform and deep neural networks. IEEE Transactions on Industrial Informatics, 14(7), 3271-3280.
22. Li, Y., Yang, C., & Sun, Y. (2022). Sintering quality prediction model based on semi-supervised dynamic time feature extraction framework. Sensors, 22(15), 5861.
23. De La Ree, J., Centeno, V., Thorp, J. S., & Phadke, A. G. (2010). Synchronized phasor meas-urement applications in power systems. IEEE Transactions on smart grid, 1(1), 20-27.
24. Monticelli, A. (1999). State estimation in electric power systems: a generalized approach. Springer Science & Business Media.
25. A. Monticelli, "Electric power system state estimation," Proceedings of the IEEE, vol. 88, no. 2, pp. 262–282, 2000.
26. Karvelis, G. I., Korres, G. N., & Darmis, O. A. (2022). State estimation using scada and pmu measurements for networks containing classic hvdc links. Electric Power Systems Research, 212, 108544.
27. Deng, R., Xiao, G., Lu, R., Liang, H., & Vasilakos, A. V. (2017). False data injection on state estimation in power systems—Attacks, impacts, and defense: A survey. IEEE Transactions on Industrial Informatics, 13(2), 411-423.
28. Deep Learning. (2022).
29. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
30. Zimmerman, R. D., Murillo-Sánchez, C. E., & Thomas, R. J. (2010). MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Transactions on power systems, 26(1), 12-19."
31. Wang, S., Bi, S., & Zhang, Y. J. A. (2020). Locational detection of the false data injection attack in a smart grid: A multilabel classification approach. IEEE Internet of Things Journal, 7(9), 8218-8227.
32. Moslemi, R., Mesbahi, A., & Velni, J. M. (2018). A fast, decentralized covariance selection-based approach to detect cyber attacks in smart grids. IEEE Transactions on Smart Grid, 9(5), 4930-4941.
33. Sedghi, H., & Jonckheere, E. (2015). Statistical structure learning to ensure data integrity in smart grid. IEEE Transactions on Smart Grid, 6(4), 1924-1933.
34. Bi, S., & Zhang, Y. J. (2014). Using covert topological information for defense against malicious attacks on DC state estimation. IEEE Journal on Selected Areas in Communications, 32(7), 1471-1485.
35. Keras the Python deep learning API. (2023).
36. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., & Zheng, X. (2016). Ten-sorflow: Large-scale machine learning on heterogeneous distributed systems.
37. Cortes, C., & Vapnik, V. (1995). Support-vector networks Machine learning Vol. 20.
38. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30.
39. T. Fawcett. (2005).