Learning-Based Defense of False Data Injection Attacks in Power System State Estimation

Arnav Kundu
Electrical Engineering
Texas A&M University
arnav1993k@tamu.edu

Abhijeet Sahu
Electrical Engineering
Texas A&M University
abhijeet_ntpc@tamu.edu

Katherine Davis
Electrical Engineering
Texas A&M University
katedavis@tamu.edu

Erchin Serpedin
Electrical Engineering
Texas A&M University
eserpedin@tamu.edu

***Abstract*—The electric power grid has evolved immensely with time, and the modern power grid is dependent on communication networks for efficient transmission and distribution. Since communication networks are vulnerable to various kinds of cyberattacks, it is essential to detect them and prevent the power grid from getting affected by it. False data injection attacks (FDIA) are one of the most common attack strategies where an attacker tries to trick the underlying control system of the grid, by injecting false data in sensor measurements to cause disruptions. We propose a data augmented deep learning-based solution to detect such attacks in real-time. We aim at generating realistic attack simulations on standard IEEE 14-bus architecture and train neural networks to detect such attacks.**

***Index Terms*—Anomaly Detection, False Data Injection Attacks, Local anomaly detection, Long Short Term Memory (LSTM)**

I. Introduction

The power system is a dynamic and complex system connecting diverse electrical components such as generators, transmission lines, and distribution systems. To ensure reliable operation of such a complex system, we require secure system monitoring of systems via synchrophasors, Current Transformers (CTs), Potential Transformers (PTs), etc. State variables, like voltage and phase angles at each bus, are estimated from these measurements, and the system operator controls the estimated state to operate the grid. With the use of a state estimator and its associated contingency analysis, a system operator can review each critical contingency to determine whether each possible future state is within reliability limits, and make decisions regarding its operation. However, with the fusion of advanced cyber infrastructure to the physical domain, measurements are prone to alteration by the cyber invaders, which can affect the process of state estimation and mislead the power grid control system, resulting in catastrophic consequences [1]–[3].

False Data Injection Attacks (FDIAs) can be introduced to a transmission system to trick the state estimator into predicting wrong states without getting detected [4]. Detection methods try to find anomalies in the data received through the communication channel. Such methods depend on the real-time correlation between data points or the temporal structure of the data to classify a new set of measurements as anomalous. A significant drawback of this approach is that it does not adapt well to changing patterns in transmission behavior over time [5].

FDIAs are challenging to detect using conventional residual-based methods since the attacks specifically bypass these mechanisms, and spatial arrangements of the devices are not taken into consideration [5]. These methods were traditionally built to avoid bad data or severe measurement errors for DC state estimators, where it is assumed that bad data will necessarily lead to high residual error. However, with more sophisticated FDIAs, we can ensure that bad data can be injected with negligible impact on residual [4]. This is a classic contextual anomaly detection problem. Deep learning has shown significant promises in solving complex tasks and has been used in pattern recognition problems like object detection, speech recognition, and anomaly detection [6]–[8].

Deep learning uses a data-driven approach where a function approximator is trained using gradient descent over a given set of data points. The success of deep learning can be attributed to the ability of neural networks to learn complex functions and the availability of massive data-sets. Motivated by its application and success in the field of speech recognition [7] and anomaly detection [8], we explore how deep neural networks can be applied to detect false data injection attacks in the electric power grid.

Artificial Neural Networks (ANNs) have shown significant performance in representing complex functions [9]. Especially in anomaly detection, deep neural networks have been applied in many applications like fraud detection, sensor network anomaly detection, video surveillance, log anomaly detection, and Internet of Things (IoT) [8]. Deep neural networks have been used in supervised [10], semi-supervised [11] and unsupervised setting [12] in the past for anomaly detection. Specifically, for anomaly detection in spatially and temporally correlated data, direct supervision using classification networks and unsupervised methods using auto-encoders have shown impressive results in the past [13].

The organization of the rest of the papers sections are described as follows. Section II introduces the background on state estimation and FDI attacks. Section III explains attack design of random attacks and the measurement data generation process for the IEEE 14 bus system. Section IV presents the proposed models for detecting the FDI attacks. Section V presents the attack and defense case results and analysis on the IEEE 14 bus system. Finally, Section VI concludes our paper with the scope of future work.

II. Background

The electric power grid uses a set of measuring devices spread across various branches to determine the state of the system. These states are then used to take necessary control actions. However, the true state of the system cannot be directly determined from the measuring devices because of

978-1-7281-0407-2/19/$31.00 ©2019 IEEE

induced noise and measurement inaccuracies. A power system state estimator is used to determine the correct state of the grid using those measurements. The estimation mechanism is described by the Eq. 1,

$$z = Hx + \epsilon \tag{1}$$

where z denotes the measurement vector, x represents the state vector, H stands for the system characteristic matrix, and ϵ is the error in estimation. The objective is to find a state vector x that minimizes the energy (variance) of residual ϵ defined as

$$\min_x 1^T (z - Hx)^2 \tag{2}$$

In conventional state estimators, for a new state vector to be considered as a correct state, the residual should be below a defined threshold. In a false data injection attack, an adversary aims to hack the readings of multiple measurements, to mislead the state estimator to predict incorrect states without affecting the residue. These attacks can be random without any particular motive [4] or targeted to certain state variables with specific intentions [14].

Some researchers have established that such attacks can be prevented totally by securing a subset of all measuring devices on an encrypted network [15]. However, as the size of the network increases the number of devices that needs to be secured increases; hence, it is not scalable. The basic residual-based detection can also be improved by using L_∞-norm instead of L_2-norm [16].

In most of the prior work, it has been assumed that the attacker has complete knowledge of the system. However, in a practical scenario, it can be assumed that an intruder will not be aware of the entire power system. It has also been proven that FDIAs can be possible with partial system information as well [17]; hence, a hacker with incomplete information can cause an FDIA.

The residual-based detection systems fail to consider the spatial distribution of the measuring devices and temporal distribution of the measurements. This inherent information can be used to derive a spatio-temporal correlation between measurements and therefore, detect attacks. In a superficial sense, the problem can be reduced to detecting anomalies in a dense graph. Inspired by application in cyber intrusion and sensor networks, researchers have tried to apply nearest neighbor classifiers and other statistical classification techniques [18]. However, these methods are slow for large systems and have a nonlinear run time complexity. Besides, these models do not scale well and cannot be applied effectively to power grids [18]. With the current advancements in deep learning and sequential pattern recognition [8], we propose a deep learning-based anomaly detection system to detect and identify various kinds of intrusions.

III. Attack Designs

The basic concept behind FDIAs is straightforward, i.e., to generate an attack vector a such that:

$$z + a = H(x + c) + \epsilon \tag{3}$$

where, c is the change in states induced due to the attack vector.

A. Random Attacks

One of the simplest attacks is a least-effort random attack where an attacker with access to a fixed set of compromised measuring devices tries to bias random state variables. The following derivation, Eq. 4 and Eq. 5, shows why such an attack is possible [4].

$$\begin{aligned} \hat{x}_a &= \left(H^T \Sigma H\right)^{-1} H^T \Sigma z_a \\ &= \left(H^T \Sigma H\right)^{-1} H^T \Sigma (z + a) \\ &= \hat{x} + \left(H^T \Sigma H\right)^{-1} H^T \Sigma a \end{aligned} \tag{4}$$

$$\begin{aligned} \|z_a - H\hat{x}_a\| &= \left\| z + a - H\left(\hat{x} + \left(H^T \Sigma H\right)^{-1} H^T \Sigma a\right)\right\| \\ &= \left\| z - H\hat{x} + a - H\left(H^T \Sigma H\right)^{-1} H^T \Sigma a\right\| \\ &= \left\| z - H\hat{x} + a - H\left(H^T \Sigma H\right)^{-1} H^T \Sigma a\right\| \\ &= \|z - H\hat{x} + Hc - Hc\| \\ &= \|z - H\hat{x}\| \end{aligned} \tag{5}$$

This proves that if we can come up with a vector a as in Eq. 6

$$a = Hc \tag{6}$$

then we can introduce an attack without getting detected.

This attack can be possible only if the attacker has access to all the meters. However, in a real scenario, it is not feasible for the attacker to get hold of all the measuring devices in a network. As a result, we cannot choose any random attack vector. This is the reason behind modeling sparsity of attack vectors. The sparsity-preserving attacks are generated following the methods described in [4]. Let $I_{\text{meter}} = \{i_1, \ldots, i_k\}$ be the set of indices of the k meters the attacker has access to. Therefore, $a = (a_1, \ldots, a_m)^T$ with $a_i = 0$ for $i \notin I_{\text{meter}}$. In order to find one such attack vector we define a projection matrix in Eq. 7.

$$P = H\left(H^T \Sigma H\right)^{-1} H^T \Sigma \tag{7}$$

From the previous equation we can derive as follows:

$$\begin{aligned} a - H\left(H^T \Sigma H\right)^{-1} H^T \Sigma a &= 0 \\ Pa &= Ia \\ (P - I)a &= 0 \\ Ba &= 0 \end{aligned} \tag{8}$$

Therefore, an attacker needs to find a non-zero attack vector a such that $Ba = 0$ and $a_i = 0$ for $i \notin I_{\text{meter}}$. Let us represent
$a = (0, 0, \ldots, a_1, 0, 0 \ldots, a_2, 0, \ldots, a_3, \ldots a_k \ldots)^T$
$B = (\ldots, b_{i1} \ldots b_{i2}, \ldots b_{ik} \ldots)$,
where a_i is the attack corresponding to the i-th meter for $i \in I_{\text{meter}}$ and b_i is the column vector in B corresponding to the index of a_i in a. Therefore, we define $B' = (b_{i1}, b_{i2}, \ldots, b_{ik})$ and $a' = (a_1, a_2, \ldots, a_k)$ such that $Ba = 0$. If the rank of B is less than k then there can be infinite solutions to $Ba = 0$. According to Meyer [19], a' can be determined as 9:

$$a' = \left(I - B'^{-1} B'\right) d \tag{9}$$

where, d is some random non-zero vector. If rank of $B' \geq k$, then there is only one unique solution to $Ba = 0$ ie. $a = 0$. It

can also be logically inferred that the probability of generating a random attack increases if we have access to more meters.

In an ideal power system, attacks are rare. Besides, it is highly unlikely that in all scenarios where attacks are possible, the attacker has access to all measurement units. Once an attack is introduced, it can stay for a variable amount of time. Therefore, while generating simulations, we consider these factors in choosing the frequency, duration, and location of these attacks. We have created cases to select a fraction of random devices parameterized by k from n measuring devices ($k \in 0.1, 0.2, 0.3, 0.4, 0.5$). Similarly, we have assigned a probability (p) of the grid being under attack where $p \in [0, 0.2)$ and we have kept each attack live for a random number of samples ($t \in$[5,10]). This provides a huge range of possible combinations to store in our database of attacks. The algorithm used to generate these attack data-sets is described in Algorithm 1.

Algorithm 1 Generation of Attacks

function GENERATEATTACK($attackType$, $measurements, devices$)
  for $i \in$ sizeOf($measurements$) **do**
    $options \leftarrow [0.1, 0.2, 0.3, 0.4, 0.5]$
    $k \leftarrow$ `choice`$(options, 1)$
    $p \leftarrow$ `random`$(0, 1)$
    $t \leftarrow$ `randomInt`$(5, 11)$
    $hacked \leftarrow$ `choice`$(devices,$ `int`$(k \times$ `sizeOf`$(devices))$
    if $p < 0.2$ **then**
      for j in range(t) **do**
        $z \leftarrow measurements[i]$
        $a \leftarrow$ `getRandomAttack`$(hacked, t)$
        $zNew \leftarrow z + a$
        `saveRecords`$(zNew, hacked)$
        $j+ = 1, i+ = 1$
      end for
    else
      $z \leftarrow measurements[i]$
      `saveRecords`(z)
    end if
  end for
end function

IV. Defense Mechanism

As mentioned earlier, the state estimator relies on simple Euclidean distance-based anomaly detection mechanisms to recognize incorrect measurements. It is shown in Eq. 6 that such a system is easy to trick. Therefore, the inherent spatio-temporal correlation of these measurements missed by residual-based detection needs to be considered in our new FDIA detection system. Moreover, it is difficult to estimate the hyper-parameter k in some Euclidean distance-based techniques such as k-nearest neighbor for anomaly detection [20].

In [18], a correlation-based FDIA detection mechanism has been proposed, where a semi-supervised structure is employed. An operator needs to define a correlation sphere for various meters on the network. A single meter might lie in multiple correlation spheres. This approach ensures that the spatio-temporal correlation between the measurements is preserved. At every iteration, correlations within a correlation sphere are calculated, and if a considerable divergence is found, then an anomaly is flagged. This method is highly efficient in terms of run-time complexity but would need humongous effort in designing the correlation spheres manually. Besides, this method will not allow changes to network topologies.

In [21], an approach based on sparse optimization, low-rank matrix factorization, and nuclear norm minimization has been explained. The assumption here is that the gradually changing state variables will typically lead to a low-rank measurement matrix Z_0 and the attack matrix (attack vectors over time) is sparse. Therefore, the problem translates to a matrix separation problem as

$$\min_{\mathbf{Z}_0, \mathbf{A}} \text{Rank}\,(\mathbf{Z}_0) + \|\mathbf{A}\|_0 \tag{10}$$

s.t. $\mathbf{Z_a} = \mathbf{Z}_0 + \mathbf{A}$ which can be formulated as a convex optimization problem as follows.

$$\min_{\mathbf{Z}_0, \mathbf{A}} \|\mathbf{Z_o}\|_* + \lambda \|\mathbf{A}\|_1 \tag{11}$$

s.t. $\mathbf{Z_a} = \mathbf{Z}_0 + \mathbf{A}$

$\|\mathbf{Z_0}\|_*$ is the nuclear norm of Z_o, i.e., the sum of singular values of Z_o. This kind of optimization problem has been studied across the domains of compressive sensing and matrix completion and can be solved using off the shelf optimization algorithms. The problem with this approach is the computational complexity because of its iterative nature [21]. This paper also proposes a faster way using low-rank matrix factorization, where low-rank matrix Z_0 is represented as a product of two matrices U and V. Even though this approximation helps to remove the expensive Singular Value Decomposition (SVD) step, it is iterative, which is non-linear in time. When analyzed as a classification problem techniques like Support Vector Machines (SVM) [22] has also been used.

We propose a deep learning-based data-driven FDIA detection method, which is robust, has an almost linear run-time complexity. Recurrent neural networks are heavily used to capture temporal correlation in data [23], [24]. In addition to addressing variable-length sequences, they also help to keep the predictor small and are computationally light because of shared parameters.

A. Approach 1

In our first approach, we define a multi-layer Recurrent Neural Network (RNN) for the entire grid. In our model, we use an advanced version of RNN called Long Short Term Memory (LSTM) to prevent vanishing and exploding gradients [25]. The model architecture is shown in Figure 1. The first layer encodes the inputs at every time-step $z_i \in \mathbb{R}^{m \times 1}$ to an output state $o_i^1 \in \mathbb{R}^{h^1 \times 1}$, where m is the number of measurement devices in the grid, h is the dimension of the output of the first LSTM layer, and $i \in [t-k, t]$. Similarly, the second and third layer uses the output of the previous layer to produce their respective outputs for each time-step. The key factor here is that the weights for a given layer for every time-step remain the same. The fourth LSTM layer uses the outputs of the third layer and projects it to a single-dimensional value. This output is passed through a 'sigmoid' function to indicate the probability of an attack being present at a given time-step. The output of this network can be represented as:

$$Y = \sigma(f(W^4 f(W^3 f(W^2 f(W^1 \times Z)))))) \tag{12}$$

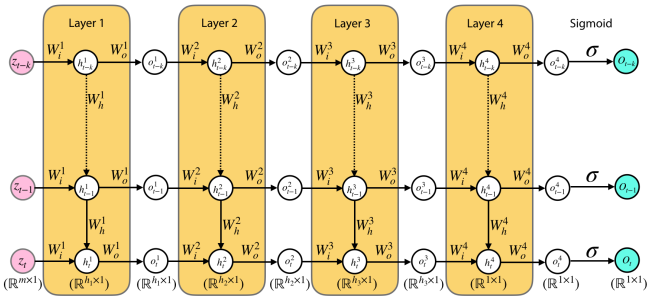

Fig. 1. Architecture of the network.

where, $Y \in \mathbb{R}^{k\times 1}$ is the vector of probabilities indicating whether the system is under attack in each of the k input time-steps in $Z \in \mathbb{R}^{k\times 1}$. W^i represents the combination of weights at i^{th} layer. The function f is the LSTM function as described in [25].

The model should be able to capture spatio-temporal correlations keeping a linear run-time by sharing model parameters. However, this model is a global intrusion detector which relies on training data for all possible attack scenarios that can occur. Such an extensive training dataset is difficult to generate for large power grids. Also, this approach will not be able to adapt easily to the addition or removal of buses from the grid, and the model would need retraining once any such changes are done. Therefore, we further propose a more localized and decentralized approach.

B. Approach 2

In the second approach, we define a similar network model, as discussed earlier, but we do not take all measurements as inputs. This is a more localized approach where we select a set of measuring devices which are interconnected by a particular bus. In this way, we are enforcing the spatial arrangement of the devices on the network. Therefore, the major learning happens in the temporal domain. The algorithm to define connections for a given bus is described in Algorithm 2. We have used a set to indicate the buses each device is connected to, which is returned by $device.buses$.

Algorithm 2 Mapping busses to measurements

```
function GETCONNECTIONS(busNum, gridMap)
    devices = []
    for device in gridMap do
        if busNum in device.buses then
            devices.append(device)
        end if
    end for
    return devices
end function
```

This approach focuses on having individual models for each bus, which makes this approach robust to changes. Suppose a new bus is added we do not need to train the entire network for all the buses like in Section IV-A. Similarly, if the network topology is changed, then we would need to retrain the models local to the affected region only. Besides, this approach can also help to locate compromised devices at a macro level.

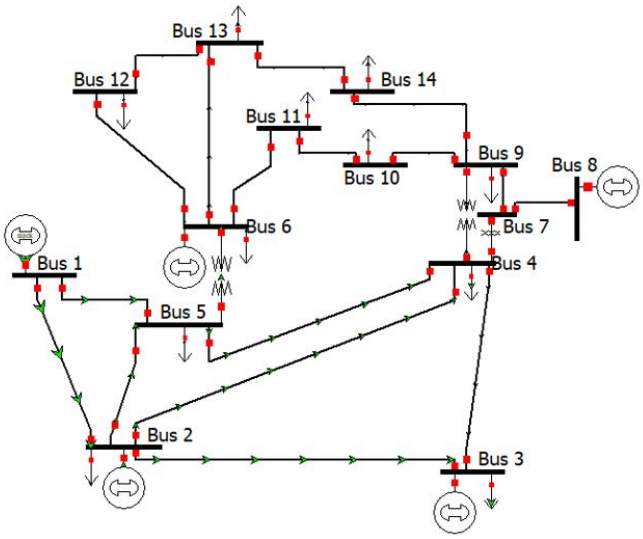

Fig. 2. Topology for the IEEE 14 Bus Case

V. Results and Analysis

A. IEEE 14 Bus Case

The simulation uses real-world power consumption data to generate 39 measurements and the intrusion state of these devices at each time step. The methodology for the bus level modeling is based on the synthetic load model proposed in [26]. This model has an hourly basis specification, including residential, commercial, and industrial sectors loads, and its results were validated utilizing actual utility measurement data. The topology for the model developed contains 14 buses, 2 generators, 3 synchronous condensers, and 11 loads as presented in Figure 2. It also contains a three winding transformer equivalent.

The data generated has 5-minute resolution data for an entire year, which gives around 105400 time-steps of SCADA data for the test case. For each time-step, a state estimation model is solved utilizing the power flow equations. In order to obtain a flawless resolution of 5-minute in our test case, a piece-wise polynomial algorithm based on the cubic spline extrapolation methodology is utilized to correct the missing measurements in the grid.

1) Attack generation: We construct random attacks on these measurements to affect the state variables. The attack data is stored along with the state variables under attack and the devices compromised (Line 13 and 18 Algorithm 1), which is treated as the training data for our deep learning models.

For the random FDI attacks, the attacker is trying to observe its influence on the probability of finding an attack vector by modifying the number of meters. The IEEE 14 bus system with 39 measurements and 26 state variables to estimate is considered for the case study. The experiment is repeated for 105400 timestamps.

It can be observed from Figure 3 that the probability of finding an attack vector is directly proportional to the number of meters that can be accessed. At n = 2, it can be seen that it has a probability of attack at around 0.27, while at n = 11, it is possible to have a higher probability of almost 1. From the equation, $k \geq m$-n+1, it is observed that if k = 11,

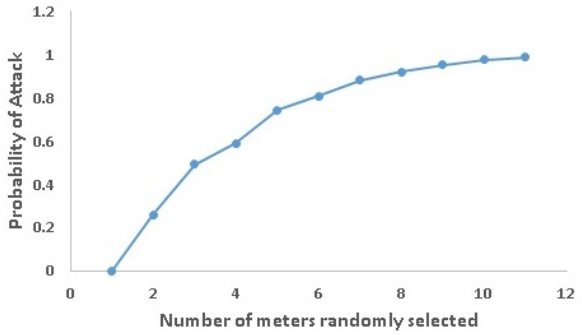

Fig. 3. Probability of finding an attack vector with varying number of meters compromised

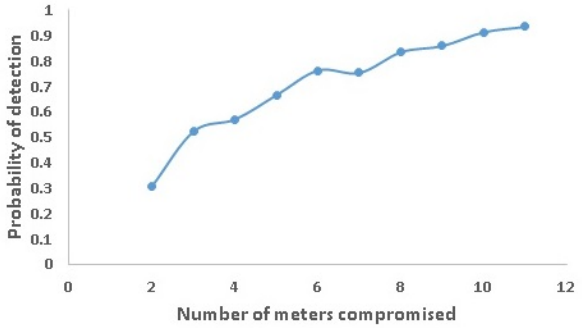

Fig. 4. Probability of detection using conventional Chi-square Test with level of significance α = 0.005

then to perform a successful attack with 26 state variables, 36 measurements are needed at least.

A Chi-square test is implemented to find the efficiency of an attack. As per the Chi-square test, the minimum threshold to prevent the attack detection is 29.8, for a degree of freedom 13, which is the difference between the number of meters and the state variables, with the level of significance α set to 0.005. While keeping this threshold, the probability of detection is evaluated, for a varying number of meters compromised by the attacker, as shown in Figure 4.

2) Preprocessing the data for training: Since random attacks are a super-set of all other kinds of attacks, we use random attacks for training and testing our models. We have split this data into 6 subgroups depending on the level of intrusion. This is decided by the number of devices compromised at the time of the attack. The attack data also needs to be formatted for the training process. This step splits the data into sequences over a rolling window for training. The data is already on a per-unit scale; therefore, normalization is not necessary. This split in sequences is crucial because we are using a recurrent framework which needs features at every time step over a fixed sequence length for training.

3) Training: We trained our model using each of the approaches described in Section IV on the attack data generated. Training is done using binary cross-entropy loss on an ADAM optimizer.

In $Approach$ 1, we trained our model on all 39 measurements to predict the state of the system. We use training data with attacks at three levels of intrusion comprising of 4, 12, and 20 devices compromised at each level. We call this method the Global RNN Detector. For testing, we used intrusions of different levels, i.e., when 8, 16, and some random number of devices are compromised. To regularize the model, we have used a dropout [27] of 5% for each LSTM layer. We have compared the performance of our model with an SVM model trained using the same data with Radial Basis Function (RBF) as kernel and a fully connected Artificial Neural Network (ANN) with the equal number of layers as the RNN model. We use F1-scores [28] to indicate the performance of these models. Our model returns the probability score of an attack being present. However, we need to decide a classification threshold of an attack. This is done by splitting the training data into training and validation data in a 9:1 ratio. The validation data is evaluated by our model to obtain probability scores of attacks. We use precision-recall curve [29] on these scores to decide the correct classification threshold. The results are shown in Table I and II.

TABLE I
COMPARISON OF FDIA DETECTION USING GLOBAL RNN DETECTOR

Number of devices compromised	8	16	Random
KNN	0.9845	1	0.5063
SVM	0.9158	0.9997	0.3749
ANN	0.9879	1.0000	0.7039
RNN	**0.9705**	**1.0000**	**0.7178**

TABLE II
COMPARISON BASED ON RECALL

Number of devices compromised	8	16	Random
KNN	0.9746	1	0.3759
SVM	0.8447	0.9994	0.2307
ANN	0.9760	1.0000	0.5431
RNN	**0.9861**	**1.0000**	**0.5836**

The Global RNN detector performed better in comparison to the k-Nearest Neighbor (KNN), SVM and ANN detector. Higher F1-Score for the detector indicates the detector has a high precision and recall (as shown in Table II), which means the model is able to capture intrusions correctly and ignore non intruded states with good accuracy as well. On the other hand, the KNN and SVM models are proving to be very good estimators for fixed scenarios like when 8 or 16 devices are compromised, but they fail to efficiently predict intrusions when a random number($\leq$ 20) of devices are compromised.

Similarly, for $Approach$ 2, we trained a model for every bus using data from the measuring devices connected to the respective buses according to Algorithm 2. As done in the previous case, we use data from 4, 12, and 20 devices compromised cases for training the network and data from 8, 16, and random devices compromised cases as test data. A critical point to ensure here is that the distribution of attacks in the training data should be similar to that in the testing conditions. We call this approach the distributed local approach. Just like the global detector, we use 5% dropout for every LSTM layer for regularization. In the above case, we have already observed that the ANN performs better than KNN and SVM. So, in this case, we have compared the

performance(F1-Score) of ANN and RNN Detector. Also, we have used the mean scores of all buses that are under attack in our scenarios. The results are shown in Table III and IV.

TABLE III
COMPARISON DETECTION OF FDIAS USING LOCAL RNN DETECTOR BASED ON F1-SCORE

Number of devices compromised	8	16	Random
ANN	0.8378	0.9501	0.9170
RNN	**0.9515**	**0.9569**	**0.9538**

TABLE IV
COMPARISON DETECTION OF FDIAS USING LOCAL RNN DETECTOR BASED ON RECALL

Number of devices compromised	8	16	Random
ANN	0.9596	0.9669	0.9603
RNN	**0.9563**	**0.9628**	**0.9577**

VI. CONCLUSION AND FUTURE WORKS

We have explored how RNNs can help detect FDIAs using a global and a local approach. RNNs are found to be a better detector than KNN, SVM and ANN, especially when put under random levels of intrusion. It is also observed that the performance of the global detector degrades when subjected to random levels of intrusion, but that of the local detector stays consistent. This might be occurring because the global detector has to learn a higher dimensional space i.e., it has to classify intrusions for 0-39 devices. On the other hand, the local detector focuses on a limited number of devices and is not affected by the intrusions that are happening on other buses other than the bus it is trained for. Therefore, the search space for identifying attacks is restricted for the local detector; therefore, random levels of intrusion does not affect its performance.

This architecture will be further tested on larger power system cases and specific types of attacks in our future work. Moreover, as the power system size increases the generation of attack data for training neural networks might be difficult. Therefore, deep learning-based unsupervised and generative neural network models will also be designed.

ACKNOWLEDGMENT

The material presented in this paper is based upon work supported by the NSF division of Electrical, Communication and Cyber System under Award Number 1808064.

REFERENCES

[1] D. Alert, "Analysis of the cyber attack on the ukrainian power grid," 2016.
[2] L. Streltsov, "The system of cybersecurity in ukraine: principles, actors, challenges, accomplishments," *European Journal for Security Research*, vol. 2, no. 2, pp. 147–184, 2017.
[3] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.
[4] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security*, vol. 14, no. 1, pp. 1–33, May 2011. [Online]. Available: https://doi.org/10.1145/1952982.1952995
[5] S. Bi and Y. J. Zhang, "Graphical methods for defense against false-data injection attacks on power system state estimation," *IEEE Transactions on Smart Grid*, vol. 5, no. 3, pp. 1216–1227, 2014.
[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017. [Online]. Available: http://doi.acm.org/10.1145/3065386
[7] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde, "Jasper: An end-to-end convolutional neural acoustic model," 2019.
[8] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019.
[9] K. Mehrotra, C. K. Mohan, and S. Ranka., *Elements of artificial neural networks.*, 1997.
[10] R. Chalapathy, E. Z. Borzeshi, and M. Piccardi, "An investigation of recurrent neural architectures for drug name recognition," 2016.
[11] D. Wulsin, J. Blanco, R. Mani, and B. Litt, "Semi-supervised anomaly detection for eeg waveforms using deep belief nets," in *2010 Ninth International Conference on Machine Learning and Applications*, Dec 2010, pp. 436–441.
[12] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," 2017.
[13] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," *arXiv preprint arXiv:1510.01553*, 2015.
[14] Q. Yang, J. Yang, W. Yu, D. An, N. Zhang, and W. Zhao, "On false data-injection attacks against power system state estimation: Modeling and countermeasures," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 717–729, March 2014.
[15] S. Bi and Y. J. Zhang, "Defending mechanisms against false-data injection attacks in the power system state estimation," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*, Dec 2011, pp. 1162–1167.
[16] O. Kosut, Liyan Jia, R. J. Thomas, and Lang Tong, "Limiting false data attacks on power system state estimation," in *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, March 2010.
[17] M. A. Rahman and H. Mohsenian-Rad, "False data injection attacks with incomplete information against smart power grids," in *2012 IEEE Global Communications Conference (GLOBECOM)*, Dec 2012, pp. 3153–3158.
[18] P. Chen, S. Yang, J. A. McCann, J. Lin, and X. Yang, "Detection of false data injection attacks in smart-grid systems," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 206–213, Feb 2015.
[19] C. D. Meyer, *Matrix analysis and applied linear algebra*, 2000, vol. 71.
[20] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, no. 12, pp. 3448 – 3470, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S138912860700062X
[21] L. Liu, M. Esmalifalak, Q. Ding, V. A. Emesih, and Z. Han, "Detecting false data injection attacks on power grid by sparse optimization," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 612–621, March 2014.
[22] M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor, "Smarter security in the smart grid," in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2012, pp. 312–317.
[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams., *Learning representations by back-propagating errors*, pp. 696–699.
[24] H.-j. Kim and K.-s. Shin, "A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets," *Appl. Soft Comput.*, vol. 7, no. 2, pp. 569–576, Mar. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2006.03.004
[25] S. Hochreiter and J. Schmidhuber., *Long Short- Term Memory*, November 1997, pp. 1735–1780.
[26] H. Li, A. L. Bornsheuer, T. Xu, A. B. Birchfield, and T. J. Overbye, "Load modeling in synthetic electric grids," in *2018 IEEE Texas Power and Energy Conference (TPEC)*, Feb 2018, pp. 1–6.
[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[28] Wikipedia, "F1 score — Wikipedia, the free encyclopedia," http://en.wikipedia.org/w/index.php?title=F1%20scoreoldid=911716685, 2019, [Online; accessed 22-August-2019].
[29] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, p. e0118432, 2015.