# Detection of False Data Injection Attacks in Smart Grids using Recurrent Neural Networks

Abdelrahman Ayad[1], Hany E.Z. Farag[2], Amr Youssef[3] and Ehab F. El-Saadany[1,4]

[1]ECE Department, University of Waterloo, ON, Canada
[2]Department of Electrical Engineering and Computer Science, York University, ON, Canada
[3]Concordia Institute for Information Systems Engineering, Concordia University, QC, Canada
[4]On leave with Khalifa University, Petroleum Institute, Abu Dhabi, UAE

***Abstract*— False Data Injection (FDI) attacks create serious security challenges to the operation of power systems, especially when they are carefully constructed to bypass conventional state estimation bad data detection techniques implemented in the power system control room. This paper investigates the utilization of Recurrent Neural Networks (RNN) as a machine learning technique to detect these FDI attacks. The proposed detection algorithm is validated throughout simulations of FDI in power flow data over the span of five years using IEEE-30 Bus system. The simulation results confirm that the proposed RNN-based algorithm achieves high accuracy in detecting anomalies in the data, by observing the temporal variation in the successive data sequence.**

***Index Terms*— Smart grid, Cyber-security, False Data Injection, Recurrent Neural Networks, State estimation, Bad data detection**

## I. INTRODUCTION

Smart grids aim to increase the hosting capacity of renewable generation, improve asset utilization, and increase grid ability to respond to and resolve problems on the network faster. This new, continually evolving, format of smartening the grid has added many advantages to power systems, but also has posed different challenges to the power system operators. The complex structure and functionalities of smart grids require sophisticated and decentralized control schemes to ensure a continuous, stable and coherent operation of the grid. Toward that end, the addition of a cyber layer to provide the needed communication between different entities and/or devices is a key component in smart grids. This communication, however, creates vulnerability that the power networks can be attacked. The crucial operation requirements of power systems have been addressed by several authors, and also implications of cyber-attacks have been pointed at extensively, e.g., [1]. During the transition towards smart grid, the capability of conventionally used techniques to defend against any attempt to manipulate the system needs to be carefully assessed [2], [3]. As smart grids are considered as cyber-physical systems, attacks on the grid can target the cyber layer, physical layer, or both. Many findings such as the work in [4] show high vulnerabilities of power systems to be attacked. A common conclusion is that cyber-security approaches do not specifically address the physical aspect of the grid. As such, both cyber-security and system based theories are required to be combined together in order to secure the smart grid. Attacks on the smart grid can take various forms, such as Denial of Service Attacks (DOS) [5], and time synchronization attacks [6]. One important type of attacks, which is investigated in this paper, is the False Data Injection (FDI) attacks. During these attacks, the attacker injects incorrect data or measurements in order to alter the state measurements of the system [7]. The literature classifies methods of protection against FDI attacks into protection-based approaches and detection-based approaches. Protection-based approaches depend on protecting measurements of certain sensors from being attacked [8]. The realization of these approaches depends on the determination of minimal set of measurements needed for protection. Drawbacks of these approaches include the drop of measurements redundancy and the unguaranteed effectiveness under all operating conditions [9]. Detection-based approaches, however, use Bayesian framework to detect the attack [10]. The objective of these methods is to detect any anomaly, or abnormal data points in the system state or measurements. One general drawback of these methods is the inability of detecting anomalies that fit the historical distribution of the data [9]. Much attention has been drawn to the problem of detecting FDI attacks, especially when attackers are able to construct undetectable attacks that bypass the bad data detection within the power system state estimators [11]. Several previous works have pointed to the consequences and implications of FDI on the grid [3] . The authors in [12] reviewed the theoretical basis of FDI attacks and their defense strategies in modern systems. In [13], graphical methods are used to detect attacks by using tree-pruning based approximation algorithm. The authors in [9] proposed a statistical method based on tracking the dynamics of measurements and using probability distributions derived from measurements variation to detect the attacks. Another approach for anomaly detection is based on complementing bad data detection methods with independent data such as historical and forecast data to detect data anomalies [14]. Machine learning techniques have been used to model complex systems with high accuracy in different fields, including power systems. Deep neural networks have been

978-1-5386-2453-1/18/$31.00 ©2018 IEEE

used in energy disaggregation to estimate each appliance consumption from the overall home's electrical consumption [15]. Also, adaptive neural networks has been used to detect FDI attacks on the sensors of an unmanned aerial vehicles [16]. In [17], Support Vector Machines (SVM) have been implemented to detect stealthy attacks in the smart grid.

This paper investigates the application of Recurrent Neural Networks (RNN) approach to detect well-constructed FDI attacks that are not detectable by conventional power system state estimators. RNN is a special architecture of neural networks that is able to create a memory and use previous inputs and outputs to predict the next state. RNN combines the features of FDI attack detection approaches in [9] and [14], by observing the dynamics of measurement variations and thus implicitly considering the historical data, represented in the successive data sequence. This feature inherently exists in RNN as it utilizes previous output states in the subsequent prediction, and is extremely useful for detecting a measurement that has been manipulated by an attack constructed to bypass bad data detection.

The remainder of this paper is organized as follows: Section II provides a background overview on flow measurements in power systems, state estimation used in control centers and a brief overview of the construction of attacks that bypass the state estimation. Also, related work in the area of cyber-security in power systems is presented. Section III presents the methodology of attack detection using RNN with an overview on the background and theory of RNN. Section IV presents the case study and implementation of the algorithm on collected data. Section V shows the simulation results of the case study with a brief discussion of the efficiency of the used method. Finally conclusions and suggested future work is presented in Section VI.

## II. Background and Overview

### A. DC power flow

Power system operators continuously monitor the states of the system in the control center to ensure the system stability. Power flow measurements are at the heart of the of control system, indicating estimates of the system sate variables according to meter measurements. State variables include bus voltage angles and magnitudes. The DC model is often used by power systems operators as an alternative for the nonlinear AC model, which is computationally expensive and its solution might not converge for large power systems. The DC power flow model is constructed as a linearized model as follows [18]

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{e} \tag{1}$$

where $\mathbf{z} \in \mathbb{R}^m$ is the power flow measurements vector; $\mathbf{x} \in \mathbb{R}^n$ is the system states variables vector; $\mathbf{H} \in \mathbb{R}^{m \times n}$ is the measurement Jacobian matrix that maps the states to the measurements; and $\mathbf{e} \in \mathbb{R}^m$ is the measurements errors vector.

### B. Power systems state estimation

State estimation infers the system state variables from available measurements, which include bus voltages, active and reactive power injection at each bus, and power flow in branches. Based on the DC Power flow model in (1), and the weighted least squares (WLS) method , the state estimation problem aims to find an estimate $\hat{\mathbf{x}}$ that minimizes the WLS error defined as follows:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} (\mathbf{z} - \mathbf{H}\mathbf{x})^{\intercal} \mathbf{W} (\mathbf{z} - \mathbf{H}\mathbf{x}) \tag{2}$$

where $\mathbf{W}$ is a diagonal matrix whose elements are the reciprocals of the variances of meter errors.

$$\hat{\mathbf{x}} = (\mathbf{H}^{\intercal}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\intercal}\mathbf{W}\mathbf{z} \tag{3}$$

Power system operators often use the $L_2$-norm (i.e., $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|$) to detect the presence of bad measurements. This is achieved by comparing the residue $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|$ with a threshold $\tau$, and bad data is detected if $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| > \tau$. Measurements errors are to be assumed as independent, following a normal distribution with zero mean [19].

### C. Construction of FDI attacks to bypass bad data state estimation

Liu *et al.* [11] showed that attackers, equipped with a prior knowledge of the network topology, are able to manipulate the measurements by inserting attack vectors that are undetected by the residue test of the state estimation techniques. Let $\mathbf{z}$ be the original measurement vector than can pass the bad measurement detection. For an attack vector $\mathbf{a} \in \mathbb{R}^m$, the compromised measurement data is presented as

$$\mathbf{z_a} = \mathbf{z} + \mathbf{a} \tag{4}$$

If $\mathbf{a}$ is constructed as a linear combination of the column vectors of $\mathbf{H}$ (i.e., $\mathbf{a} = \mathbf{H}\mathbf{c}$), then $\mathbf{z_a}$ can pass the bad data detection. The obtained state variables estimated from compromised data $\mathbf{z_a}$ are referred to as $\hat{\mathbf{x}}_{\mathbf{comp}}$, and can be represented as $\hat{\mathbf{x}}_{\mathbf{comp}} = \hat{\mathbf{x}} + \mathbf{c}$, where $\mathbf{c}$ is a non-zero vector $\in \mathbb{R}^n$. Since original measurements $\mathbf{z}$ can pass the bad data detection, then

$$\begin{aligned} \|\mathbf{z_a} - \mathbf{H}\hat{\mathbf{x}}_{\mathbf{comp}}\| &= \|\mathbf{z} + \mathbf{a} - \mathbf{H}(\hat{\mathbf{x}} + \mathbf{c})\| \\ &= \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}} + (\mathbf{a} - \mathbf{H}\mathbf{c})\| \\ &= \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| \leq \tau \end{aligned} \tag{5}$$

Thus, the $L_2$-norm of $\mathbf{z_a}$, which is less than threshold $\tau$ can bypass the bad data detector. The results from [11] have triggered an important alarm to the necessity of revisiting the techniques used to defend against possible cyber-physical attacks in the power networks, as conventional state estimation approaches might fail to detect FDI that are constructed with prior knowledge of the network topology.

## III. Proposed RNN for detection of FDI attacks

### A. RNN overview

Unlike regular neural networks which assume inputs (outputs) are independent from each other, Recurrent Neural Networks (RNN) are a special type of neural networks, that

make use of sequential information to predict the output [20]. Sequential data might contain temporal correlation between inputs at time $t$ and inputs at time $t-1, t-2, \cdots$. By using information of previously calculated outputs, RNN is capable of constructing a memory, which is to be used in computing the output.

In this paper, RNN is utilized as a sequence classifier. For an observation sequence $\in \mathbb{R}^l, \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_l}\}$ and corresponding labels $\in \mathbb{R}^l, \{y_1, y_2, ..., y_l\}$, the objective is develop a learning function that maps (labels) the feature to its corresponding label, i.e., $f : \mathbf{x} \rightarrow y$. RNN models dynamic systems, by sending feedback signals, so that subsequent outputs depend on computed output. This can be mathematically modelled as [21]:

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t) \tag{6}$$

where $\mathbf{h}_t$ is the hidden state, $\mathbf{h}_{t-1}$ is the previous hidden state, and $\mathbf{x}_t$ is the current feature observed, and $f$ is a nonlinear mapping function. Equation (6) captures the essence of RNN and what differentiates it from regular neural networks. The hidden state $\mathbf{h}_t$ is used as a memory to capture sequence information. Fig. 1 shows the unfolding of RNN in time of the computation. $\mathbf{x}_t$ denotes the input at time $t$, while the hidden state $\mathbf{h}_t$ represents the memory of the network, and it depends on the previous hidden state and current input. The output of the network at time $t$ is $z_t$. The memory concept of the RNN gives it great advantages in storing information about the time sequence.

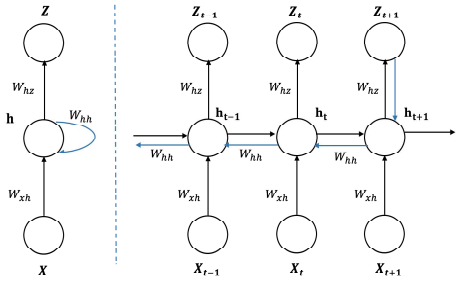


Fig. 1. Left: Recursive Description of RNN. Right: Corresponding Extended RNN model for time sequence [21]

### *B. Backpropagation Through Time (BPTT)*

Conventional Neural Networks use the Backpropagation Learning Algorithm (BPL) to train the network by adjusting the weights of the network. The BPL is based on the gradient descent technique, used to minimize the network cumulative error. The Backpropagation Through Time (BPTT) Algorithm is an extension of the BPL over a time sequence where the gradient at each output depends on calculations of current as well as previous steps. BPTT has been developed by many authors independently as in [22] and [23]. Derivation of the BPTT is summarized as follows [21]:

Starting with the RNN model described in Fig. 1, parameters are assumed to be the same across the whole sequence in each time step. This assumption is used to simplify the gradient calculations. At time $t$ we have,

$$\mathbf{h_t} = \tanh(W_{hh}\mathbf{h_{t-1}} + W_{xh}\mathbf{x_t} + \mathbf{b_h}) \tag{7}$$

$$z_t = softmax(W_{hz}\mathbf{h_t} + \mathbf{b_z}) \tag{8}$$

where $b_\mathbf{h}$ and $b_z$ are the bias terms for the hidden state and prediction at time step $t$. The maximum likelihood is used to estimate the model parameters. The minimization of objective function of negative log likelihood is

$$\mathcal{L}(\mathbf{x}, y) = -\sum_t y_t \log z_t \tag{9}$$

where $z_t$ is the prediction at time step $t$. The notation $\mathcal{L}$ will be used as objective function for simplicity. The notation $\mathcal{L}(t)$ indicates the output at time $t$ while $\mathcal{L}(t+1)$ indicates the output at time $t+1$. The derivative of equation (9) with respect to $z_t$ is

$$\frac{\partial \mathcal{L}}{\partial z_t} = -\sum_t y_t \frac{\partial log z_t}{\partial \mathbf{h}_t} = -\sum_t y_t \frac{1}{z_t}\frac{\partial z_t}{\partial \mathbf{h}_t} \tag{10}$$

Using the chain rule and by deriving the gradient of the $softmax$ function from (8), we get

$$\frac{\partial \mathcal{L}}{\partial z_t} = -(y_t - z_t) \tag{11}$$

The weight $W_{hz}$ between the hidden state $\mathbf{h}$ and output $z$ is the same across all time sequence. Therefore it can be differentiated at each time step and summed as follows:

$$\frac{\partial \mathcal{L}}{\partial W_{hz}} = \sum_t \frac{\partial \mathcal{L}}{\partial z_t}\frac{\partial z_t}{\partial W_{hz}} \tag{12}$$

The gradient with respect to a bias unit $b_z$ is obtained similarly as

$$\frac{\partial \mathcal{L}}{\partial b_z} = \sum_t \frac{\partial \mathcal{L}}{\partial z_t}\frac{\partial z_t}{\partial b_z} \tag{13}$$

Considering the time step $t \rightarrow t+1$, the gradient is derived with respect to the weight $W_{hh}$ as

$$\frac{\partial \mathcal{L}(t+1)}{\partial W_{hh}} = \frac{\partial \mathcal{L}(t+1)}{\partial z_{t+1}}\frac{\partial z_{t+1}}{\partial \mathbf{h}_{t+1}}\frac{\partial \mathbf{h}_{t+1}}{\partial W_{hh}} \tag{14}$$

The above equation only considers the time step $t \rightarrow t+1$. As the RNN model uses previous state for subsequent state calculation, the hidden state $\mathbf{h}_{t+1}$ depends partially on hidden state $\mathbf{h}_t$. Similar to $W_{hz}$, the weight $W_{hh}$ is shared across the whole time sequence. Therefore we get

$$\frac{\partial \mathcal{L}(t+1)}{\partial W_{hh}} = \frac{\partial \mathcal{L}(t+1)}{\partial z_{t+1}}\frac{\partial z_{t+1}}{\partial \mathbf{h}_{t+1}}\frac{\partial \mathbf{h}_{t+1}}{\partial \mathbf{h}_t}\frac{\partial \mathbf{h}_t}{\partial W_{hh}} \tag{15}$$

Aggregating gradients with respect to $W_{hh}$ over the whole sequence and using the BPTT from time $t$ to 0

$$\frac{\partial \mathcal{L}(t+1)}{\partial W_{hh}} = \sum_t \sum_{k=1}^{t+1} \frac{\partial \mathcal{L}(t+1)}{\partial z_{t+1}}\frac{\partial z_{t+1}}{\partial \mathbf{h}_{t+1}}\frac{\partial \mathbf{h}_{t+1}}{\partial \mathbf{h}_k}\frac{\partial \mathbf{h}_k}{\partial W_{hh}} \tag{16}$$

The same process applied in (12)-(16) is also applied on the weights $W_{xh}$, by taking the gradient with respect to $W_{xh}$ over the whole sequence to obtain

$$\frac{\partial \mathcal{L}(t+1)}{\partial W_{xh}} = \sum_{t} \sum_{k=1}^{t+1} \frac{\partial \mathcal{L}(t+1)}{\partial z_{t+1}} \frac{\partial z_{t+1}}{\partial \mathbf{h}_{t+1}} \frac{\partial \mathbf{h}_{t+1}}{\partial \mathbf{h}_k} \frac{\partial \mathbf{h}_k}{\partial W_{xh}} \quad (17)$$

### C. RNN parameters tuning

Based on RNN theory, a simple algorithm is developed to detect the FDI attacks. After generating the flow measurements, adding the attack vectors, and taking a subset of the data, the first step is to find the optimal parameters for the RNN. There are three main parameters which add recurrent time delayed connections to the RNN [24]:

1) Input delays $dIn \in [0, 1, 2, ...]$: This allows the output to not only depend on current input, but also on previous inputs. For regular neural networks, $dIn = [0]$. The benefit of the Input delay is to use the temporal correlation between successive inputs to predict the next output.
2) Internal delays $dInternal \in [0, 1, 2, ...]$: This allows the current internal states to depend on previous $dInternal$ internal states, and specifies how many previous internal states to be used. For regular neural networks, $dInternal = [0]$.
3) Output delays $dOut \in [0, 1, 2, ...]$: This determines how many previous output states are used to predict current output. For regular neural networks, $dOut = [0]$. This parameter controls the recurrent behavior of using current output for subsequent outputs. Previous outputs are particularly important in applications where the predicted output heavily depends on previous outputs.

The optimal set of parameters that achieves least error has been determined by training the network several times using the BPTT algorithm. After finding the optimal parameters, the network is used to predict the attacks in the test data. As the output of the network is not constrained to a binary output, a threshold is used to determine the classification of output either 1 (indicating the presence of attack vector), and 0 (indicating normal flow vector).

## IV. Case Study

In order to validate the FDI attacks detection, experiments are conducted using the IEEE 30-bus system shown in Figure 2, which has 29 state variables and 112 measurements. First, the DC power flow model is used to generate the power flow measurements. The load data is obtained over a period of five years form the Independent Electricity System Operator (IESO) [25] website in Ontario, Canada. The configuration of the test system, including the matrix $\mathbf{H}$ is extracted from MATPOWER [26]. For the DC power flow model, voltage angles of all buses are used as state variables $\mathbf{x}$, while meter measurements $\mathbf{z}$ represent the real power injections of all buses and real power flows of all branches. The attack vectors are constructed according to the procedure prescribed in [11], which ensure that it will bypass the traditional bad data detection within the state estimation process. Next the attack vectors are added to the flow measurements as in (4). The data is divided into training data and testing data, representing roughly 60% (three years of data) and 40% (two years of data), respectively. Attacks are added to the measurements in the form of instances. For each attack instance, random attack vectors between 6-12 vectors are added. Attack instances are randomly distributed along all the training data as well as the testing data. All simulations were implemented in MATLAB environment.

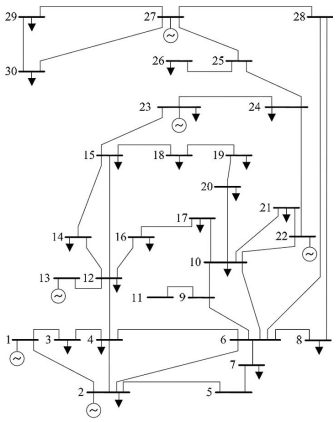


Fig. 2. IEEE 30-Bus System

To evaluate the performance of the RNN approach, statistical performance measures are used. For a given binary classification system, i.e., an algorithm that predicts either a positive or negative outcome of an experiment, a set of characteristics is used to evaluate its performance. This evaluation is especially critical for anomaly detection algorithms, where a single indication of the algorithm diagnostic ability is not enough. Let the presence of an attack in a power flow measurement be labeled as positive, while a normal measurement is labeled negative. Therefore, the mapping of instances used is as follows: True Positive (TP) is number of instances predicted positive while actual values are positive, False Positive (FP) is the number of instances predicted positive while actual values are negative, True Negative (TN) is the number of instances predicted negative while actual values are negative, and False Negative (FN) is the number of instances predicted negative while actual values are positive.

For the dataset and system under consideration, after adjusting the RNN parameters, it has been found that the optimal number of delay input is $dIn = 1$, $dInternal = 1$, and $dOut = 5$. This is due to the characteristics of data presented to the network. In the case of power flow measurements, where the load does not witness a sudden decrease or increase, there should be a correlation between the current output and previous inputs (power flow measurements). For the data set in use, increasing $dOut > 5$ causes gradients explosion and does not yield better results. The RNN has been able to detect every attack instance, where an attack instance consisted of a random number of consecutive attack

TABLE I

PERFORMANCE TABLE FOR RNN APPROACH ON IEEE 30-BUS

| Criteria | Score |
|---|---|
| # Attack Instances | 100 |
| # Detected Instances | 100 |
| # Attack Vectors | 915 |
| # Detected Vectors | 850 |
| True Positive (TP) | 778 |
| False Positive (FP) | 72 |
| True Negative (TN) | 209541 |
| False Negative (FN) | 137 |
| Accuracy | 99.901% |
| Sensitivity | 85.027% |
| Specificity | 99.969% |
| Precision | 91.529% |

vectors. In addition, the RNN has very good indications of all statistical performances for attack vectors as shown in table I. The BPTT algorithm was applied on MATLAB using the pyrenn toolbox [24] after making necessary modifications to suit the data.

Throughout the simulations, the FDI attacks were represented by 100 attack instances, each of which consisted of 6-12 attack vectors. A successful identification of an attack (true positive) is counted if the RNN was able to detect one or more attack vector during an interval when there was an attack instance. In most cases, all attack vectors of an attack are detected. False positives occurred exclusively immediately after an attack, i.e., the RNN predicted one attack vector after the attack instance was finished, but never identified a false positive in isolation of an attack instance. In some attack instances, the RNN experienced a delay in identifying an attack vector (false negative), in the range of 1-3 attack vectors during an attack instance, but succeeded to detect the rest of attack vectors, thus achieving the objective of detecting FDI attack instances in all the tested cases.

## V. CONCLUSION

In this paper, Recurrent Neural Networks (RNN) have been investigated as a new tool to detect False Data Injection (FDI) attacks that target the smart grid. The FDI attacks are constructed such that they bypass the state estimation bad error detection used in control systems, thus imposing a constraint that makes the attacks undetected by conventional methods. RNN proved to be superior in detecting such attacks, as it recognizes that the data is in a time sequence and captures the temporal correlation between the measurements at successive time instants as well as the spatial correlation within the measurements at the same time instant.

## VI. ACKNOWLEDGMENT

This publication was made possible by NPRP grant no. NPRP 9-055-2-022 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

## REFERENCES

[1] Y. Yang, T. Littler, S. Sezer, K. McLaughlin, and H. Wang, "Impact of cyber-security issues on smart grid," in *Innovative Smart Grid Technologies (ISGT Europe), 2011 2nd IEEE PES International Conference and Exhibition on*. IEEE, 2011, pp. 1–7.

[2] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Communications Surveys & Tutorials*, 2012.

[3] L. Kotut and L. A. Wahsheh, "Survey of cyber security challenges and solutions in smart grids," in *Cybersecurity Symposium (CYBERSEC), 2016*. IEEE, 2016, pp. 32–37.

[4] Y. Mo, T. H.-J. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber–physical security of a smart grid infrastructure," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195–209, 2012.

[5] L. R. Phillips, B. Tejani, J. Margulies, J. L. Hills, B. T. Richardson, M. J. Baca, and L. Weiland, "Analysis of operations and cyber security policies for a system of cooperating flexible alternating current transmission system (facts) devices." Sandia National Laboratories, Tech. Rep., 2005.

[6] Z. Zhang, S. Gong, A. D. Dimitrovski, and H. Li, "Time synchronization attack in smart grid: Impact and analysis," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 87–98, 2013.

[7] G. Liang, J. Zhao, F. Luo, S. Weller, and Z. Y. Dong, "A review of false data injection attacks against modern power systems," *IEEE Transactions on Smart Grid*, 2017.

[8] 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, 2014.

[9] G. Chaojun, P. Jirutitijaroen, and M. Motani, "Detecting false data injection attacks in ac state estimation," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2476–2483, 2015.

[10] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "On malicious data attacks on power system state estimation," in *Universities Power Engineering Conference (UPEC), 2010 45th International*. IEEE, 2010, pp. 1–6.

[11] 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 (TISSEC)*, vol. 14, no. 1, p. 13, 2011.

[12] G. Liang, J. Zhao, F. Luo, S. Weller, and Z. Y. Dong, "A review of false data injection attacks against modern power systems," *IEEE Transactions on Smart Grid*, 2017.

[13] 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.

[14] A. Ashok, M. Govindarasu, and J. Wang, "Cyber-physical attack-resilient wide-area monitoring, protection, and control for the power grid," *Proceedings of the IEEE*, 2017.

[15] J. Kelly and W. Knottenbelt, "Neural NILM: Deep neural networks applied to energy disaggregation," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. ACM, 2015, pp. 55–64.

[16] A. Abbaspour, K. K. Yen, S. Noei, and A. Sargolzaei, "Detection of fault data injection attack on uav using adaptive neural network," *Procedia computer science*, vol. 95, pp. 193–200, 2016.

[17] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Systems Journal*, 2014.

[18] A. Abur and A. G. Exposito, *Power system state estimation: theory and implementation*. CRC press, 2004.

[19] A. J. Wood and B. F. Wollenberg, *Power generation, operation, and control*. John Wiley & Sons, 2012.

[20] R. J. Williams and D. Zipser, "Gradient-based learning algorithms for recurrent networks and their computational complexity," *Backpropagation: Theory, architectures, and applications*, vol. 1, pp. 433–486, 1995.

[21] G. Chen, "A gentle tutorial of recurrent neural network with error backpropagation," *arXiv preprint arXiv:1610.02583*, 2016.

[22] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[23] J. A. Hertz, A. S. Krogh, and R. G. Palmer, *Introduction to the theory of neural computation*. Basic Books, 1991, vol. 1.

[24] D. Atabay, "pyrenn: First release. zenodo." [Online]. Available: http://doi.org/10.5281/zenodo.45022

[25] IESO, "http://www.ieso.ca/power-data/data-directory." [Online]. Available: http://www.ieso.ca/power-data/data-directory

[26] R. D. Zimmerman, "Matpower 4.0 b4 users manual," *Power Syst Eng Res Cent*, pp. 1–105, 2010.