# GRAPHICAL CONVOLUTION NETWORK BASED SEMI-SUPERVISED METHODS FOR DETECTING PMU DATA MANIPULATION ATTACKS

by

Wenyu Wang

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Engineering

at

The University of Wisconsin-Milwaukee

August 2020

ABSTRACT

GRAPHICAL CONVOLUTION NETWORK BASED SEMI-SUPERVISED METHODS FOR DETECTING PMU DATA MANIPULATION ATTACKS

by

Wenyu Wang

The University of Wisconsin-Milwaukee, 2020
Under the Supervision of Professor Lingfeng Wang

With the integration of information and communications technologies (ICTs) into the power grid, electricity infrastructures are gradually transformed towards smart grid and power systems become more open to and accessible from outside networks. With ubiquitous sensors, computers and communication networks, modern power systems have become complicated cyber-physical systems. The cyber security issues and the impact of potential attacks on the smart grid have become an important issue. Among these attacks, false data injection attack (FDIA) becomes a growing concern because of its varied types and impacts. Several detection algorithms have been developed in the last few years, which were model-based, trajectory prediction-based or learning-based methods.

Phasor measurement units (PMUs) and supervisory control and data acquisition (SCADA) system work together to monitor the power system operation. The unsecured

devices could offer opportunities to adversaries to compromise the system. In the literature review part of this thesis, the main methods are compared considering computing accuracy and complexity. Most work about PMUs ignored the reality that the number of PMUs installed in a power system is limited to realize observability because of high installing cost. Therefore, based on observable truth of PMU and the topology structure of power system, the graph convolution network (GCN) is proposed in this thesis. The main idea is using selected features to define violated PMU, and GCN is used to classify susceptible violated nodes and normal nodes. The basic detection method is introduced at first. And then the calculation process of neural network and Fourier transform are described with more details about graph convolution network. Later, the proposed detection mechanism and algorithm are introduced. Finally, the simulation results are given and analyzed.

To
my parents and my brother

TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor Dr. Lingfeng Wang who has the attitude and substance of genius and gave me this precious opportunity to accomplish the research thesis, especially with limited time as a dual M.S. degree graduate student of both Chong Qing University (CQU) and UW-Milwaukee (UWM). I am grateful for his consistent guidance, responsible supervision, ample encouragement and support throughout this work. Without his help and support, I could not have finished this dual master's degree program.

My appreciation is also extended to Dr. Liu who gave me many useful feedbacks on my research progress updates. Besides, my thanks go to my roommate Sherly, and my good friends Annie, Candice, Michelle and Shayne. Their encouragement and optimistic attitudes toward life inspired me a lot to overcome the difficulty of finding new ideas and coding. Their accompany made my life more interesting and fancier.

Lots of gratitude goes to the CQU's Director of the Dual Master's Degree Program — Zhouyang Ren, who helped me go through the application process to the program with his tolerance and understanding which made the application submission much easier.

I am indebted to my family, whose value to me grows with age. Thanks are due to their everlasting love and consistent support so that I could have the chance to grow following my own heart.

# Chapter 1 Introduction

## 1.1. Background on Power System Security

Power system plays an important role on daily life. Reliable electricity supply is supported by basic cyber physical systems. Modern power system encompassed a large-scale use of Cyber-Physical Systems (CPS) known as Cyber-Physical Production Systems (CPPS) [1]. Cyber physical systems (CPSs) are an integration of sub-systems with multi layers and physical domains interconnected through communication networks. The integration of the information and communication technologies (ICT) into the power grid which could be regarded as CPS realizes efficient and reliable bidirectional power flow [2]. However, the involvement of varied technologies, the interconnection between each layers and algorithms implemented in the power grid result in the vulnerability of the system. For example, a distributed system relies on the data collected from different entities. In such system, the security of shared data plays an important role on effective decision making and control. Therefore, more attention should be paid to protect the security of the transmitted data in these networks [3].

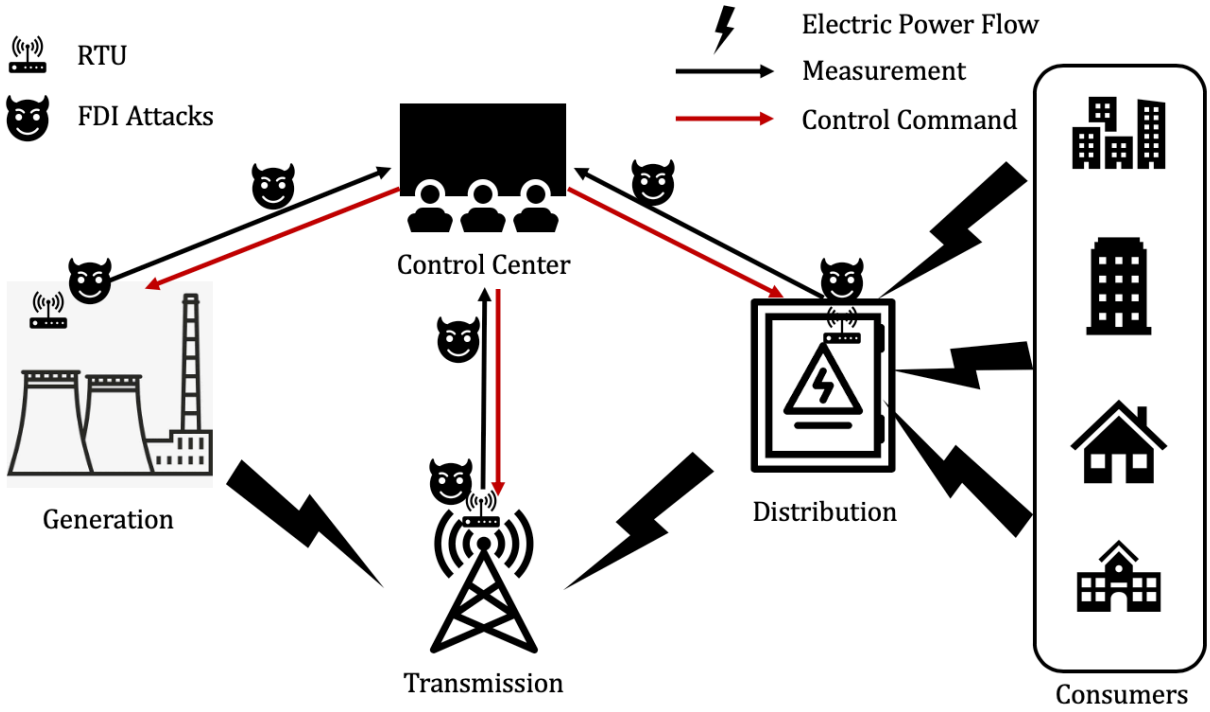



Figure 1 Vulnerabilities of Power System Towards False Data Injection Attacks

Power system security contains two aspects: physical security and cyber security. Physical security is about the ability of a power system to work well with the existence of severe disturbances. Cyber security illustrates the security of the communication networks and computer systems which support the power system operation [4]. The vulnerabilities of power system towards false data injection attacks is shown in Figure 1 [41]. The whole power system is divided into five parts including generation, transmission, distribution, consumers and control center. Each part installs remote terminal units (RTU) to send and receive information from each other. Cyber-attacks could be involved into any connections. Cyber-attacks have the capability of undermining or totally disrupting the control systems in the

power system. Cyber-attacks have resulted in my security problems in recent years. For example, in 2003, the well-known Slammer worm penetrated the control system of the David-Besse nuclear plant in Ohio, USA [5]. A wide breakout took place in Kiev, Ukraine for several hours affecting three major distribution companies and more than 225,000 customers in 2015[6].

### 1.2. False Data Injection Attacks in Power System

A major part of cyber security and the cross-domain vulnerabilities of power system is false data injection [7]. FDIA could be used in different systems and layers in the smart grid as shown in Figure 2. The way the FDIA could be used in any processor-based devices is presented in [8]. R. Macwan and C. Drew illustrated how a FDIA works on the IEC61850 standard Ethernet-based communication protocol [9]. G. Liang and J. Zhao demonstrates several possible cyber-based FDIA and the associated impacts in the power grid [10]. A successful FDIA cause the state estimator to generate erroneous values which may lead the system operators make wrong decisions, the system response unpredictably and unstably, and then make either economic impacts or stability impacts to the power system.

PMU data manipulation attack is another type of FDIA aiming at wide area measurement system (WAMS), which attempts to blind the control centers in accurate awareness of real-

time operating conditions of power systems [12]. The Phasor measurement units (PMUs) are measurement devices equipped with the global positioning system (GPS) technology for precise timing. By synchronizing to GPS time, PMUs have the ability to provide accurate synchronous phasor measurements for geographically dispersed nodes in power grids [11]. PMU measurements are important because control centers may directly use the data or results given by PMUs to make a decision. In most situation, the compromised data could be detected by state estimators of bad data detection model, however, adversaries still could manipulate the data by maliciously injecting a set of measurements. It may lead the control center to make improper actions and cause unwanted consequences of the power system. Worse further, some automatic processes, such as automatic generation control, automatic voltage regulation and transient stability assessment, heavily rely on correct measurements to work. Once these inputs are no longer accurate, the resulted erroneous control actions may threaten the stability of power system. Moreover, if dispatchers see on the screen, e.g., a “fault” is happening and isn’t automatically removed, they will probably think there is something wrong with the relay protection system and hence scramble to cut off the “fault” line manually, which can also cause severe consequences [12].

S. Pal, B. Sikdar and J. H. Chow [13] proposed a method to detecting FDIAs using transmission line parameters, i.e., the equivalent impedance of transmission lines. The detection is realized by continuously monitoring the equivalent impedances of transmission lines and classifying observed anomalies for detecting the presence and location of attacks. J. Wang, D. Shi, Y. Li, J. Chen, H. Ding and X. Duan[12] use machine learning tools which is called deep autoencoder to detect distributed PMU data manipulation attacks. However, both of the two methods ignored the reality that PMUs are not installed at all the buses of a system because of their high capital cost. It is important to find the best locations to place PMUs so that the number of PMUs can be minimized.

In general, only limited PMUs are installed in the power system to realize the observability of the whole system. The magnitude and phase angle of buses which do not install PMUs are get by Kirchhoff's Current Law and Kirchhoff's Voltage Law. Therefore, a graph convolution network (GCN) based FDIA detection method is proposed in the thesis using specific feature selection given to PMUs to help verify stealthy attacks which evade bad data detector. At first, four features are selected to form the feature space including voltage magnitude and phase angle, active power and inactive power. After the feature space is verified, compromised PMUs are detected using GCN. GCN is trained by predefined data

sets with normal and abnormal information to learn the weights and bias. And test data sets are used to check the ability of the model. That's to say, GCN works as a semi-supervised deep learning method to detect false data attacks by learning weights and bias of power system topology. Besides, instead of sequential methods mentioned in forehand researches, GCN could learn the topology of power systems which means the spatial features of a system is used into FDIAs.

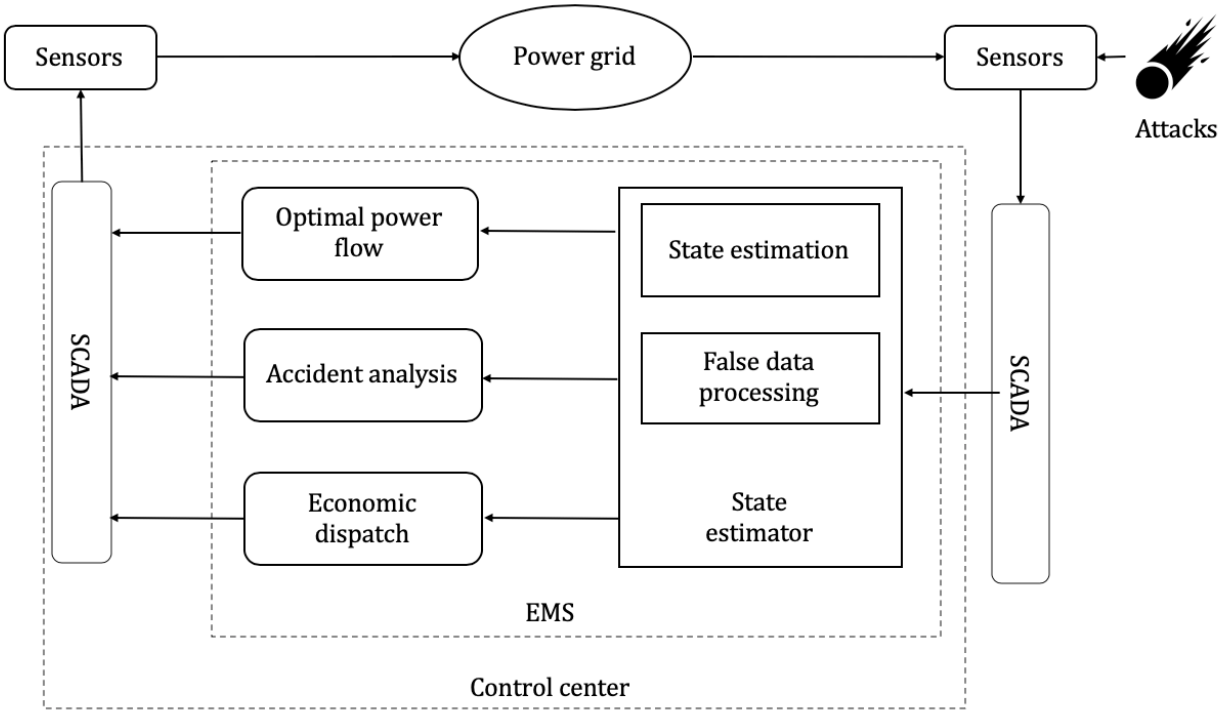



Figure 2 False Data Injection Attacks in Smart Grid

### 1.3. Thesis Structure

The rest of the thesis is structed as follows. Section II is related to literature review, existing methods of detecting FDIAs are reviewed. Section III gives basic theoretical

backgrounds on state estimation and FDIA; Section IV introduces knowledge about Fourier Transform and GCN to support the detection method proposed in the thesis. Besides, the feature selection procedure and mythology are also described in section V. In Section VI, the performance of the proposed method is evaluated, with potential improvements given as future works. Section VII makes a conclusion of the thesis.

# Chapter 2 Literature Review

The detection methods could be mainly categorized into three types: the first one is state estimation-based methods, the second one is trajectory prediction-based methods and machine learning based methods.

## 2.1. State Estimation-Based Methods

In recent years, a number of false data detection (FDD) methods based on state estimation which are designed to alleviate FDIAs in smart grid CPS have been proposed. The research given by Liu [14] is one of the first to look at the vulnerability of state estimation to cyber-attacks, where the attacker is assumed to have knowledge about measurement configuration to create undetectable attacks. Merrill and Schweppe presented a bad data suppression estimator based on a non-quadratic cost function to improve the performance of static SE [24]. Cutsem et al. proposed an identification method attempting to mitigate some existing difficulties, such as multiple and interacting bad data [25]. Two security indexes to quantify the threat of FDIA on power grid are proposed in [15]. Multiple least trimmed squares state estimations method is proposed in [22]. The most common method is weighted least squares (WLS). A recursive WLS method was proposed in [35] to improve the convergence speed. The author of [52] also used WLS to detect FDIA in voltage controller and

transmission system. [36] used median filtering by combing the direct measurements and calculated ones. Kriging Estimator (KE) used estimated states predicted by measurements from adjacent nodes [37]. And the author of [38] used maximum likelihood (ML) estimator to detect FDIA. Gabriela Hug [16] extended the work to AC model. The authors in [23] propose an adaptive sliding mode observer with online parameter estimation to detect and respond to attacks on agents' states and sensor systems.

### 2.2. Trajectory Prediction-Based Methods

A review of definitions and proposed methods for dynamic state estimation with PMUs is available in [20]. The improvements of dynamic state estimation in monitoring the power grid are also discussed in the research [21] develops a risk mitigation approach for dynamic state estimation related to the cyber-attack impacts. Both extended Kalman Filter (EKF) [17] and unscented Kalman Filter (UKF) [18],[19] were proposed to track and forecast the power states because the equations in the AC model about state estimations are nonlinear. Authors of [22],[23] and [24] use a robust generalized maximum-likelihood-estimator on the successive batch-mode regression representations of the classical Kalman filter, extended Kalman filter and unscented Kalman filter.

### 2.3. Machine Learning-Based Methods

Traditional statistical approaches have been proposed to detect FDIAs on the state estimation in power system as mentioned before. Attempts to explore machine learning techniques has been blossoming. Esmalifalak et al. [26] proposed two machine learning based models for FDI attack detection in smart grid systems. Both models utilize principle component analysis to reduce the dimensionality of complex simulations. Authors in [27] use density ration estimation (DRE) to detect FDIAs. He et al. [28] proposed a state vector estimation (SVE) and a deep learning-based identification (DLBI) algorithm to prevent electricity theft. Wei and Mendis [29] use Conditional Deep Belief Network (CDBN) [32] to identify alteration in data that may affect the wide area monitoring systems (WAMS) in the power grid. Recurrent neural network (RNN) is used in [35] to detect FDIA. The dynamic or real-time states of a power system could be considered by the backward loop in the RNN layers. Convolution neural network (CNN) [39] performs well in extracting different features of samples. Support vector machine bases on linear non-probabilistic binary strategy which relies on two parallel hyperplanes boundaries [40]. Autoencoder (AE) is a deep neural network that provides a nonlinear compression (encoding) and expansion (decoding) of the measurement samples. The detection scheme in this algorithm is based on the error between the decoded sample and the input to the network where an alarm is flagged when

the error exceeds a certain level [54]. Hidden Markov Model (HMM) and Generative Adversarial Network is used in [56] and [59].

Table 1 Summary of False Data Injection Attacks Algorithm [13]

| Category | Algorithm | Computation Complexity | | Detection Rate |
|---|---|---|---|---|
| | | Estimation | Detection | |
| State Estimation-Based Methods | Weighted Least Squares | $\mathcal{O}(n^3t)$ | $\mathcal{O}(n^2)$ | 0.90-0.95 |
| | Median Filter | $\mathcal{O}(n)$ | $\mathcal{O}(n^2)$ | 0.99 |
| | Kriging Estimator | $\mathcal{O}(n^3)$ | $\mathcal{O}(n^2)$ | 0.96 |
| | Maximum Likelihood | $\mathcal{O}(n^3lgn)$ | $\mathcal{O}(n^2)$ | 0.997 |
| Trajectory Prediction-Based Methods | Kalman Filter | $\mathcal{O}(n^3)$ | $\mathcal{O}(n^2)$ | Detected |
| | Unscented Kalman Filter | $\mathcal{O}(n^2)$ | $\mathcal{O}(n^2)$ | Detected |
| | Extended Kalman Filter | $\mathcal{O}(n^3)$ | $\mathcal{O}(n^2)$ | Detected |
| Machine-Learning Based Methods | Support Vector Machine | $\mathcal{O}(s^2n+s^3)$ | $\mathcal{O}(nn_{sv})$ | 0.58-0.99 |
| | Convolution Neural Network | $\mathcal{O}(snn_n^2t)$ | $\mathcal{O}(nn_n^2)$ | 0.93 |
| | Recursive Neural Network | $\mathcal{O}(snn_n^2t)$ | $\mathcal{O}(nn_n^2)$ | 0.75-0.99 |
| | Deep Belief Network | $\mathcal{O}(snn_n^2t)$ | $\mathcal{O}(nn_n^2)$ | 0.93-0.98 |
| | Principal Component Analysis | $\mathcal{O}(sn^2t)$ | $\mathcal{O}(n)$ | 0.95-0.99 |

Compared to machine learning methods, deep learning could have better comprehension of high-level features hidden in a set of data such as convolution neural network (CNN). However, CNN only works well for Euclidean structure because CNN used a specific convolution kernel to abstract features of an image. For non-Euclidean structure such as topology, the number of adjacency nodes around a node is different. Therefore, graph convolution network (GCN) is proposed. The thesis uses the method classify normal and abnormal data to help state estimator detect false data injection attacks.

The attack assumption made in this thesis is that attackers can intercept the measurement data packets, modify the contents, and then transmit them to the original destinations. Besides, the number of PMUs installed in a power system is limited. The assumptions are reasonable because PMU measurements are quite susceptible to manipulation in practice. On one hand, the IEEE C37.118 Standard lacks a predefined security mechanism [10]. On the other hand, manufacturers tend to use some simple but fragile algorithms to encrypt the data to ensure the timeliness because of the high sampling rates of PMUs [11]. And the cost of a PMU is extremely high. To control the cost of PMU installation, PMUs need to optimize allocation.

# Chapter 3 Preliminary

## 3.1. Problem Formulation

### *3.1.1. State Estimation Model*

Quasi-static model represents the scenario in which the system's operating points change in a smooth and slow nature with the assumption of instantaneous response of the controller in the system. This yields negligible to transient response. Under the assumption, the various systems in smart grids could be modeled using the general measurement model realized as [30]:

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{e} \tag{1}$$

where the vector $z = (z_1, z_2, \ldots\ldots, z_m)^T$ denotes the measurements data; $x = (x_1, x_2, \ldots\ldots, x_m)^T$ denotes the system states; $e = (e_1, e_2, \ldots\ldots, e_m)^T$ denotes measurement noise, which is assumed to be Gaussian distributed with zero mean and a variance of $\sigma^2 \in R^n$. $h(x)$ denotes the functional dependency between measurements and state variables. The precise form of $h(x)$ is determined by the grid structure and line parameters.

Model (1) is commonly solved by the weight least squares (WLS) method. To find the estimated state variables $\hat{x}$, the following formula must be solved [31]:

$$\min J(x) = \sum_{i=1}^{n} w_i \left(z_i - h_i(\hat{x})\right)^2 = [\boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}})]^T \boldsymbol{W} [\boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}})] \quad (2)$$

where $w_i = {\sigma_i}^{-2}$ represents the weight for the measurement $z_i$, $\boldsymbol{W} \in \boldsymbol{R^{n \times n}}$ is a diagonal matrix composed of the weights $w_i$, and $n$ is the total number of measurements. And the solution can be computed in closed form:

$$\hat{x} = (H^T W H)^{-1} H^T W z \quad (3)$$

### 3.2. False Data Injection Attack

The goal of adversaries is to inject a false data vector $\boldsymbol{a} \in R^n$ into the measurements without being detected by the operator. The resulting observation model is [32]

$$\tilde{\boldsymbol{z}} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{a} + \boldsymbol{e} \quad (4)$$

The false data injection vector, $\boldsymbol{a}$, is a nonzero vector, such that $\boldsymbol{a_i} \neq \boldsymbol{0}, \forall \boldsymbol{i} \in \boldsymbol{\mathcal{A}}$, where $\boldsymbol{\mathcal{A}}$ is the set of indices of the measurement variables that will be attacked. The secure variable satisfies the constraint $\boldsymbol{a_i} = \boldsymbol{0}, \forall \boldsymbol{i} \in \bar{\boldsymbol{\mathcal{A}}}$, where $\bar{\boldsymbol{\mathcal{A}}}$ is the set complement of $\boldsymbol{\mathcal{A}}$. [33]

[14].

$$\rho = \|\tilde{\boldsymbol{z}} - \boldsymbol{h}(\hat{\boldsymbol{x}})\|_2^2 \quad (5)$$

where $\hat{\boldsymbol{x}} \in \boldsymbol{R^D}$ is the value calculated by formula (3). If $\rho > \tau$, where $\tau \in R$ is an arbitrary threshold, which determines the tradeoff between the detection and the false alarm probabilities, then the network operator declares that the measurements are attacked [32].

### 3.3. Undetectable Attacks and Protection Model

Potentially, FDI attacks can bypass these detection methods, resulting in erroneous estimation of system states and making the power system unstable. An attacker can inject an attack vector $\vec{a}$ with $m \times 1$ dimensions to the measured data as $\overrightarrow{z_a} = \vec{z} + \vec{a}$, where $\overrightarrow{z_a}$ denotes compromised measurements[34].$\overrightarrow{x_a}$ and $\overrightarrow{r_a}$ is used to denote state vectors and residual vectors.

$$\overrightarrow{r_a} = \overrightarrow{z_a} - h(\overrightarrow{x_a})$$

$$= \vec{z} + \vec{a} - h(\vec{x} + \vec{c})$$

$$= \vec{z} - h(\vec{x}) + \left(\vec{a} - h(\vec{c})\right)$$

$$= \vec{r} + \left(\vec{a} - h(\vec{c})\right) \quad (6)$$

where $\vec{c} = [\overrightarrow{c_1}, \overrightarrow{c_2}, \overrightarrow{c_3}, \ldots, \overrightarrow{c_n}]^T$ is a random non-zero sparse vector with $n \times 1$ dimension. To circumvent BDD, the attack vector $\vec{a}$ can be constructed as $\vec{a} = h(\vec{x} + \vec{c}) - h(\vec{x})$. Therefore, $\overrightarrow{r_a}$ is approximately equal to $\vec{r}$. If the attack vector $\vec{a}$ satisfies the

condition that the L2-norm of $\vec{a} - h(\vec{c})$ is approximately equal to zero. That's to say, the residual $\rho$ is unchanged. FDIA can bypass detection system.

In fact, the construction of FDI attack only needs partial information of a grid to find the topology information of power system. According to equations (6), adversaries could exploit small measurement errors tolerated by state estimation algorithm and only need to manipulate minority of measured data, which could hide the detection of BDD.

# Chapter 4

# Graph Convolution Neural Network

## 4.1. Introduction of Neural Network

The most common network topology is feedforward network which includes multiple layers with connections only between nodes in neighboring layers. A three-layer back propagation neural network including input layer, hidden layer and output layer [35]. There are not any connections between nodes which belong to the same layer. The input layer has m nodes that correspond to the m inputs of the network; The output layer consists of n nodes that correspond to the output of the related physical system. The number of nodes of hidden layers could be varied to fit the system target. Information is passed in one direction through the network. It begins from the input layer and ends at output layer after passing successive hidden layers [42]. The structure of neural net is given in Figure 3.

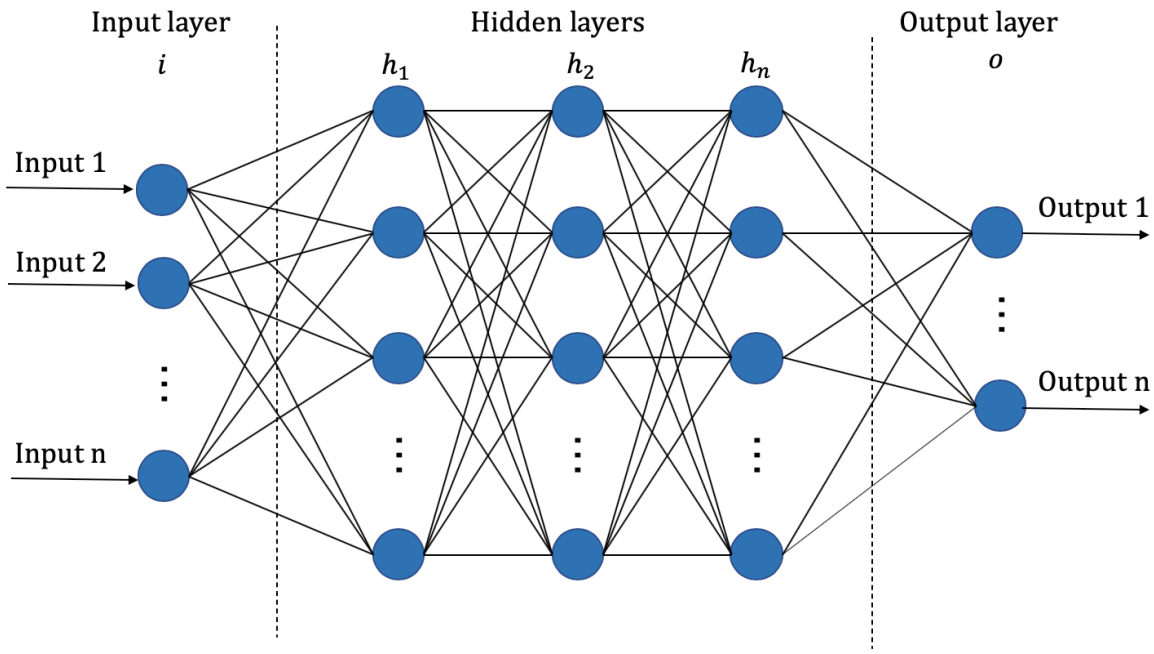



Figure 3 The Structure of Neural Network [14]

The output function of the hidden layer node should be

$$b_r = f\left(\sum_j W_{ir} a_i - T_r\right) (r = 1, \dots, u) \tag{7}$$

The output function of the output layer node should be

$$c_j = f\left(\sum_j V_{rj} b_r - \theta_j\right) (j = 1, \dots, n) \tag{8}$$

where variable $W_{ir}$ represents the weight between nodes of the input layer and hidden layers, and $V_{rj}$ represents the weight of nodes between hidden layers and the output layer. $T_r$ and $\theta_j$ represents the bias of hidden layers' nodes and output layer's nodes separately. And $f(\dots)$ is an activation function.

## 4.2. Preliminaries of Graphs Convolution

*4.2.1. Preliminaries of Graph*

Graph theory is the theory of studying graph structure data. A graph is a mathematical model that describes the relationships of a physical model and has three essential elements: nodes, edges and the weight of edges. An undirected graph is represented by $G = (V, \mathrm{E}, \mathrm{W})$, where $V$ is a finite set of vertices with $V = \{v_1, v_2, v_3, \ldots v_n\}$ and $E$ is the edge set with $E = \{e_1, e_2, e_3, \ldots e_n\}$, and W is a weighted adjacency matrix with $W = \{w_1, w_2, w_3, \ldots w_n\}$. The vertices of the graph can be any actual or abstract buses, while the edges describe the relationship between two vertices.

Suppose $\{(x)_i\}_{i=1}^{n}$ represents the vertices. Let $M$ denotes a set of vertices with the same class and $N$ denotes vertices of different classes, i.e.

$$M = \{(x_i, x_j) | x_i \; and \; x_j \; have \; the \; same \; labels\}, \tag{9}$$

$$N = \{(x_i, x_j) | x_i \; and \; x_j \; have \; the \; different \; labels\} \tag{10}$$

Based on this, there are two kinds of weighted adjacency matrices, one is zero-one weighting method: The distance between the nodes that are not connected is zero and the distance between the connected nodes is one. The other one is called Gaussian weighting method. That's to say, if the node $x_i$ and $x_j$ are connected, the distance is a specific value, otherwise, the distance is zero. Both of the two methods could be represented as below [43]:

$$W1_{ij} = \begin{cases} 1 & if(x_i, x_j) \in M \\ 0 & if(x_i, x_j) \in N \end{cases} \tag{11}$$

$$W2_{ij} = \begin{cases} exp\left(-\frac{[dist(i,j)]^2}{2\theta^2}\right) & if\ dist(i,j) \in M \\ 0 & otherwise \end{cases} \tag{12}$$

where $dist(i,j)$ is the distance between two nodes.

An essential operator in graph signal processing (GSP) is the non-normalized graph Laplacian. The graph Laplacian [44] is defined as $L := \boldsymbol{D} - \boldsymbol{A}$, $A = [a_{ij}] \in \mathbb{R}^{N \times N}$ is the adjacency matrix. $D = diag(d_1, d_2, \ldots, d_N)$ is the degree matrix of $A$ where $d_i = \sum_j a_{ij}$ is the degree of node $i$. The normalized graph Laplacian is defined as $\hat{L} := I_N - D^{\frac{-1}{2}} A D^{-\frac{1}{2}}$, where $I_N$ is the identity matrix. Due to the non-normalized and normalized graph Laplacians being the positive semi-definite matrices, it has a complete set of orthonormal eigenvectors $\{(u)_i\}_{i=0}^{n-1}$ and non-negative eigenvalues$\{(\lambda)_i\}_{i=0}^{n-1}$[6].

Then the eigen-decomposition of $\hat{\boldsymbol{L}} = \boldsymbol{U}\boldsymbol{\lambda}\boldsymbol{U}^{\boldsymbol{T}}$ could be got through the eigenvectors $u_i$ and eigenvalues $\lambda_i$, where $\boldsymbol{U} = [\boldsymbol{u_1}, \boldsymbol{u_2}, \boldsymbol{u_3}, \ldots \boldsymbol{u_n}]$ are the graph Fourier bases and $\boldsymbol{\lambda} = [\boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}, \boldsymbol{\lambda_3}, \ldots \boldsymbol{\lambda_n}]$ are the graph frequencies.

*4.2.2. Preliminaries to Convolutions on Graphs*

The goal of GCN is to learn a function of signals/features on a graph $G = (V, \mathrm{E})$ which takes as input:

- A feature description $x_i$ for every node $i$ summarized in a $N \times D$ feature matrix $X$ ($N$: number of nodes, $D$: number of input features)
- A representative description of the graph structure in matrix form, typically in the form of an adjacency matrix $A$ (or some other format) and produces a node-level output $Z$ (an $N \times F$ feature matrix, where $F$ is the number of output features per node). Graph-level outputs can be modeled by introducing some form of pooling operation.

Every neural network layer can be written as a non-linear function

$$H^{(l+1)} = f\left(H^{(l)}, A\right) \tag{13}$$

where $H^{(0)} = X$ and $H^{(L)} = Z$ (or $z$ for graph-level outputs), $L$ is the number of layers. The specific models then differ only in how $f(\cdot, \cdot)$ is chosen and parameterized.

## 4.3. Graph Convolution Network

### *4.3.1. Structure of Graph Convolution Network*

A multi-layer Graph Convolutional Network (GCN) is considered with the following layer-wise propagation rule:

$$H^{(l+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{14}$$

with $\tilde{A} = A + I_N$ is the adjacency matrix of the undirected graph G with added self-connections. $I_N$ is the identity matrix, $\widetilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and $W^{(l)}$ is a layer-specific trainable weight matrix. $\sigma(\cdot)$ denotes an activation function, such as the $ReLU(\cdot) = max(0,\cdot)$. $H^{(l)} \in \mathbb{R}^{N\times D}$ is the matrix of activations in the $l^{th}$ layer; $H^{(0)} = X$. The whole process of the propagation rule is shown below. The rule can be motivated through the first-order approximation of those localized spectral filters on graphs. The structure of a graph convolution network is shown in Figure 4.

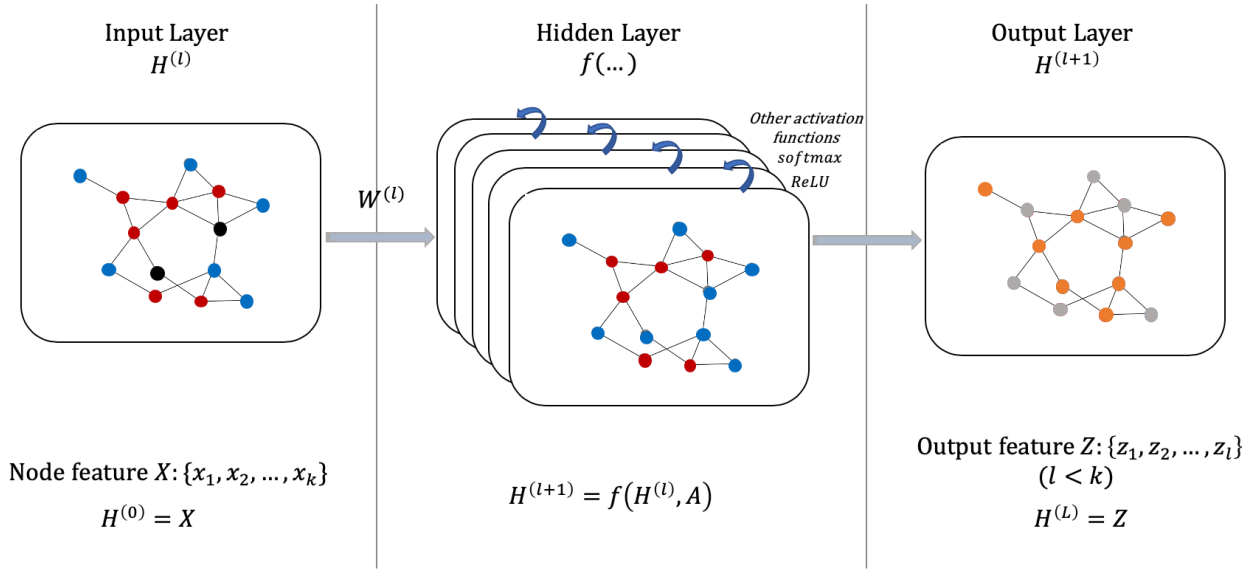



Figure 4 The Structure of a Graph Convolution Network

### *4.3.2. Vertex Domain Approach*

The vertex domain approach does not use the Fourier transform but it uses the definition of graph convolution as defined in graph signal processing (GSP). A graph shift is an operation that replaces a graph signal at a graph vertex by a linear weighted combination

of the values of the graph signal at the neighboring vertices: $x = Ax$ where $A$ is the adjacency matrix. The graph shift A extends the time shift in traditional signal processing to graph-structured data. A graph filter G is shift-invariant, i.e., the filter and shift commute: $A(Gx) = G(Ax)$ only when G is a polynomial in the adjacency matrix A. Thus, the formulation could be defined as [43]

$$G = \sum_{k=0}^{K} \alpha_k A^k \tag{15}$$

where $\alpha_k$ are the coefficients of the polynomial. Graph convolution is the matrix vector multiplication $y = Gx$.

*4.3.3. Spectral Convolution*

The classical Fourier transform

$$F(\omega) = \mathcal{F}[f(t)] = \int f(t) e^{-i\omega t}\, dt \tag{16}$$

is the expansion of a function $f$ in terms of the complex exponentials, which are the eigenfunctions of the one-dimensional Laplace operator

$$\Delta e^{-i\omega t} = \frac{\partial^2}{\partial t^2} e^{-i\omega t} = -\omega^2 e^{-i\omega t} \tag{17}$$

The convolution theorem states that [46] under suitable conditions, the Fourier transform of a convolution of two signals is the pointwise product of their Fourier transforms, which can be written as

$$\boldsymbol{f} \hat{*} \boldsymbol{g} = \hat{\boldsymbol{f}} \odot \hat{\boldsymbol{g}} \tag{18}$$

where $\boldsymbol{f} \in R^N$ and $\boldsymbol{g} \in R^N$ denotes two signals, the operator * and $\odot$ represent the convolution operator and elementwise Hadamard product, respectively, and the operator $\hat{}$ denotes the Fourier transform.

With the convolution theorem, we could realize graph convolution by achieving the Fourier transform and inverse Fourier transform on graph. As mentioned before, Laplacian $\boldsymbol{L}$ is a real symmetric matrix, it could ne decomposed and its eigenvectors $\boldsymbol{U} = \{\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{u_n}\}$ are orthogonal. Besides, all these eigenvectors have corresponding real and non-negative eigenvalues $\boldsymbol{\Lambda} = diag\{\boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}, \ldots, \boldsymbol{\lambda_n}\}$**.** Similar to the ordinary Fourier transform, the graph Fourier transform of a signal $\boldsymbol{f} \epsilon R^N$ on the graph is defined as

$$\hat{\boldsymbol{f}}(l) = \sum_{n=1}^{N} \boldsymbol{u_l}(n) \cdot \boldsymbol{f}(n) \tag{19}$$

The inverse graph Fourier transform on the graph is

$$\boldsymbol{f}(n) = \sum_{l=1}^{N} \hat{f}(l) \cdot \boldsymbol{u_l}(n) \tag{20}$$

Using the matrix form $\hat{\boldsymbol{f}} = \boldsymbol{U^T} \cdot \boldsymbol{f}$ and $\boldsymbol{f} = \boldsymbol{U} \cdot \hat{\boldsymbol{f}}$, respectively. Therefore, the convolution operation could be further defined as follows,

$$\boldsymbol{f} * \boldsymbol{g} = \boldsymbol{U} \cdot \left( (\boldsymbol{U^T} \cdot \boldsymbol{g}) \odot (\boldsymbol{U^T} \cdot \boldsymbol{f}) \right) \tag{21}$$

by using the convolution theorem and the graph Fourier transform, where $\boldsymbol{f} \in R^N$ denotes a signal, $\boldsymbol{g} \in R^N$ denotes the filter, and operator $\odot$ denotes the Hadamard product. In particular, a diagonal filter $\boldsymbol{g_\theta} = diag\{\theta_1, \theta_2, \ldots, \theta_N\} \in R^{N \times N}$ in the spectral domain is defined directly so that the element-wise product can be transformed into a common matrix multiplication. As a result, the spectral convolution can be written as

$$\boldsymbol{f} * \boldsymbol{g} = \boldsymbol{U} \cdot \left( \boldsymbol{g_\theta} \cdot (\boldsymbol{U^T} \cdot \boldsymbol{f}) \right) \tag{22}$$

*4.3.4. Spectral Graph Convolutions*

According to David and Sunil[47], spectral convolutions on graphs are defined as the multiplication of a signal $x \in \mathbb{R}^N$ (a scalar of every node) with a filter $g_\theta = diag(\theta)$ parameterized by $\theta \in \mathbb{R}^N$ in the Fourier domain based on the aforementioned contents, i.e.:

$$g_\theta * x = U g_\theta U^T x \tag{23}$$

where $U$ is the matrix of eigenvectors of the normalized graph Laplacian $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^T$, with a diagonal matrix of its eigenvalues $\Lambda$ and $U^T x$ being the graph Fourier transform of $x$. $g_\theta$ could be considered as a function of the eigenvalues of $L$, i.e.

$g_\theta(\Lambda)$. Evaluating Equation (23) is computationally expensive, as multiplication with the eigenvector matrix $U$ is $O(N^2)$. Furthermore, computing the eigen decomposition of $L$ in the first place might be prohibitively expensive for large graphs which means more storage space is needed to store the result of eigenvalues and eigenvectors. To circumvent this problem, Hammond et al. [48] suggested that $g_\theta(\Lambda)$ can be well-approximated by a truncated expansion in terms of Chebyshev polynomials $T_k(x)$ up to $K^{th}$ order:

$$g'_\theta(\Lambda) \approx \sum_{k=0}^{K} \theta'_k T_k(\widetilde{\Lambda}) \tag{24}$$

with a rescaled $\widetilde{\Lambda} = \frac{2}{\lambda_{max}}\Lambda - I_N$. $\lambda_{max}$ denoted the largest eigenvalue of $L$. $\theta' \in \mathbb{R}^K$ is now a vector of Chebyshev coefficients. The Chebyshev polynomials are recursively defined as $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, with $T_0(x) = 1$ and $T_1(x) = x$.

The optimized spectral filter uses the form of K-order polynomial to express the neighbor information that K steps away from a sample. Going back to the definition of a convolution of a signal $x$ with a filter $g'_\theta$, and a new definition of spectral convolution could get:

$$g'_\theta * x \approx U \sum_{k=0}^{K} \theta'_k T_k(\tilde{\Lambda}) U^T x = \sum_{k=0}^{K} \theta'_k U T_k(\tilde{\Lambda}) U^T x \tag{25}$$

with $\tilde{\mathrm{L}} = \frac{2}{\lambda_{max}}\mathrm{L} - I_N$. as can easily be verified by noticing that $(U\Lambda U^T)^k = U\Lambda^k U^T$. Note that this expression is now K-localized since it is a $K^{th}$-order polynomial in the Laplacian,

i.e. it depends only on nodes that are at maximum $K$ steps away from the central node ($K^{th}$-order neighborhood). The complexity of evaluating Equation (25) is $\mathrm{O}(|\mathcal{E}|)$, i.e. linear in the number of edges. With formulation (25), it could be seen that the graph convolution relies on the buses of K neighboring instead of the whole graph.

### 4.4. Layer-Wise Linear Model

A neural network model based on graph convolutions can therefore be built by architecting multiple convolutional layers in the form of Equation (25), however, each layer is non-linear and pointwise. To realize the layer-wise convolution operation, the value of K is limited to 1, i.e. a function that is linear with respect to $\mathrm{L}$ and therefore a linear function on the graph Laplacian spectrum.

In this linear formulation of a GCN, another parameter is further limited to $\lambda_{max} \approx 2$, neural network parameters will adapt to this change in scale during training. Under these approximations Equation (25) simplifies to:

$$g'_\theta * x \approx \theta'_0 x + \theta'_1 (L - L_N) x = \theta'_0 x - \theta'_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x \tag{26}$$

with two free parameters $\theta'_0$ and $\theta'_1$. The filter parameters can be shared over the whole graph. Successive application of filters of this form then effectively convolve the $k^{th}$-

order neighborhood of a node, where $k$ is the number of successive filtering operations or convolutional layers in the neural network model.

Actually, the number of parameters could be further constrained to deal with overfitting and to minimize the number of operations (such as matrix multiplications) per layer. This leaves us with the following expression:

$$g_\theta * x \approx \theta\left(I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}\right)x \tag{27}$$

with a single parameter $\theta = \theta_0' = -\theta_1'$. Note that $I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ now has eigenvalues in the range of [0,2]. The problem is that it will cause instabilities and exploding/vanishing gradients when used in a deep neural network model if the process is repeated too many times. The renormalization trick is introduced to mitigate the problem: $I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \rightarrow \widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}$, with $\tilde{A} = A + I_N$ and $\widetilde{D}_{ii} = \sum_j \tilde{A}_{ij}$.

And then the definition to a signal $X \in \mathbb{R}^{N\times C}$ with $C$ input channels (i.e. a $C$-dimensional feature vector for every node) and $F$ filters or feature maps could be generalized as follows:

$$Z = \widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}\mathrm{X}\Theta \tag{28}$$

where $\Theta \in \mathbb{R}^{C\times F}$ is now a matrix of filter parameters and $\mathrm{Z} \in \mathbb{R}^{N\times F}$ is the convolved signal matrix. This filtering operation has complexity $\mathrm{O}(|\mathcal{E}|FC)$, as $\tilde{A}X$ can be implemented as a product of a sparse matrix with a dense matrix.

### 4.5. Semi-Supervised Node Classification

A flexible model $f(X, A)$ for efficient information propagation on graphs have been introduced. To solve the problem of semi-supervised node classification, some assumptions typically made in graph-based semi-supervised learning by conditioning the model $f(X, A)$ both on the data $X$ and on the adjacency matrix $A$ of the underlying graph structure could be solved. The multi-layer GCN for semi-supervised learning is schematically depicted in Figure 4.

In this thesis, a two-layer GCN for semi-supervised node classification on a graph with a symmetric adjacency matrix A (binary or weighted) is considered. The forward model is shown as follows:

$$Z = f(X, A) = softmax\big(\hat{A}\, ReLU\big(\hat{A}XW^{(0)}\big)W^{(1)}\big) \tag{29}$$

where $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ could be get in a pre-processing step.

Figure 4 shows a basic structure and hidden layers of GCN. The whole graph of Figure 4 is a schematic depiction of multi-layer Graph Convolution Network (GCN) for semi-

supervised learning with $N$ input channels and $X$ feature maps in the output layer. The graph structure (edges shown as black lines) is shared over layers, labels are denoted by $x_i$. The middle part of Figure 4 is a visualization of hidden layer activations of a two-layer GCN including the activation functions. With each different layer, the input data is trained and get the final output as normal or abnormal finally.

Here, $W^{(0)} \in \mathbb{R}^{C\times H}$ is an input-to-hidden weight matrix for a hidden layer with $H$ feature maps. $W^{(1)} \in \mathbb{R}^{H\times F}$ is a hidden-to-output weight matrix. The softmax activation function, defined as $softmax(x_i) = \frac{1}{\mathcal{Z}} exp(x_i)$ with $\mathcal{Z} = \sum_i exp(x_i)$ is applied row-wise. For semi-supervised multi-class classification, the cross-entropy error over all labeled examples could be evaluated as follows:

$$\mathcal{L} = -\sum_{l\in y_L}\sum_{f=1}^{F} Y_{lf} ln Z_{lf} \tag{30}$$

where $y_L$ is the set of node indices that have labels. The neural network weights $W^{(0)}$ and $W^{(1)}$ are trained using gradient descent.

# Chapter 5

# Architecture of the Detection Framework

## 5.1. Feature Selection

Detection performance usually depends on the appropriate selection of the basis of feature space. Feature space is a hyperspace which exists training data and testing samples. Higher dimensional feature space could offer more detailed information about one system. PMU, located at the substation of the power generation and transmission system, are capable of measuring the real-time status of the power system, the real-time amplitude and phase angle of voltage at the bus, of current on the transmission line, and of the power at each branch [50] at a relatively high sampling rate. These measurement data are then periodically transmitted to the PDCs, usually in 50 Hz, through the local area network (LAN). Then, the aggregated data at the phasor data concentrators (PDCs) are delivered to the CC via the wide area network (WAN) for further data analysis, such as state estimation, event diagnostics, and contingency analysis.

In normal operation circumstances, the power grid works in a stable status. That's to say, all state variables vary in a mutual balanced manner based on Kirchhoff's law, demand-

response constraints and so on. As a result, ang changes of a state on a bus or transmission line will lead to either normal demand variation or system faults. This will result in corresponding state changes of the same and/or other variables on interconnected buses or transmission lines [50].

Consider a power system with $n+1$ buses. Assuming the resistance of the transmission line between bus $i$ and $j$ is small compared to its reactance, the active power-flow model from bus $i$ to bus $j$ can be expressed as

$$P_{ij} = \frac{V_i V_j}{X_{ij}} \sin\left(\theta_i - \theta_j\right) \tag{31}$$

where $V_i$ and $\theta_i$ denote the voltage magnitude and phase angle at bus $i$, respectively, and $X_{ij}$ denotes the reactance between bus $i$ and $j$. With the active power $P_i$ which is injected into bus $i$, the conservation of energy for all buses should be

$$P_i = \sum_{j \in \mathcal{A}_i} P_{ij} \tag{32}$$

where $\mathcal{A}_i$ denotes the set of buses directly connected to bus $i$.

In DC power flow studies, the difference of phase angles $\theta_i - \theta_j$ is assumed to be small between any pair of buses, and the voltage magnitudes are close to unity. Therefore, the model for dc power flow is like:

$$P_i \approx \sum_{j \in \mathcal{A}_i} \frac{\theta_i - \theta_j}{X_{ij}} \tag{33}$$

Based on formula (28), (29) and (30), the feature spaces for a dc power is chosen as voltage angles, active power and reactive power.

### 5.2. Detection Mechanism

The proposed method is used to help bad data detection module to find the stealthy data. The proposed detection mechanism is depicted in Figure 5. The mechanism considers the system states and measurements from consecutive discrete sampling time instances, i.e., the time instances when the conventional state estimation takes place. These sampling time instances may have an interval $\Delta$ ranging from milliseconds (PMU-based measurement systems) to a few seconds (conventional supervisory control and data acquisition (SCADA) system). At an arbitrary sampling time instance $t$, the mechanism takes real-time measurements $\boldsymbol{z}_t$ and the utility's knowledge of the power network $h(\cdot)$ as inputs and develop FDIA attack detection result as the output. The input data first go through

a state estimator, which estimates the current system state as $\hat{x}_t$. The estimated state is then tested with the bad data detector to prune any measurements with bad data. In this step, bad data caused by sampling and communication errors can be effectively detected, since they generally do not satisfy the circuit laws, rendering high residual values [51].

After these conventional state estimation processes, the proposed GCN mechanism introduces a new detector to further analyze the estimate system states. The method uses spatial feature of power system like convolution neural network. The input of GCN is the nodes feature including voltage amplitude, load active power and reactive power. The feature is demonstrated by binary number '0' and '1'. '1' means the feature is manipulated and '0' means the feature is under normal situations. The output of the deep learning model is normal or abnormal which is converted into binary format by graph convolution network. Once a data is defined as alert, it will send back to the bad data detector and the control center to check whether the node is attacked.

This information of measurements including power and voltage is stored in the history system state, the weights and bias are chosen randomly at first. And they are going to process offline training and adjusted these parameters repeatedly by graph convolution network using training samples. The structure of a power system is constructed by the graph

convolution network at first based on the connecting information of buses. Eighty percent of the data set is chosen as training set to train the weight and bias of the graph convolution network. Twenty percent of the data is chosen as testing set to check the accuracy of the deep leaning network.

Based on features and connection information (adjacency matrix), the network could learn the relationship or preference of each node. Therefore, another feature of graph convolution network is classifying abnormal nodes like the classical Zachary's Karate Club network. That's to say, this deep learning network could recognize the type of nodes by learning features and connecting information. As we all know, the number of PMUs installed in a power system is limited. Using the proposed graph convolution network method could detect which nodes are suspicious once one bus is confirmed as abnormal.

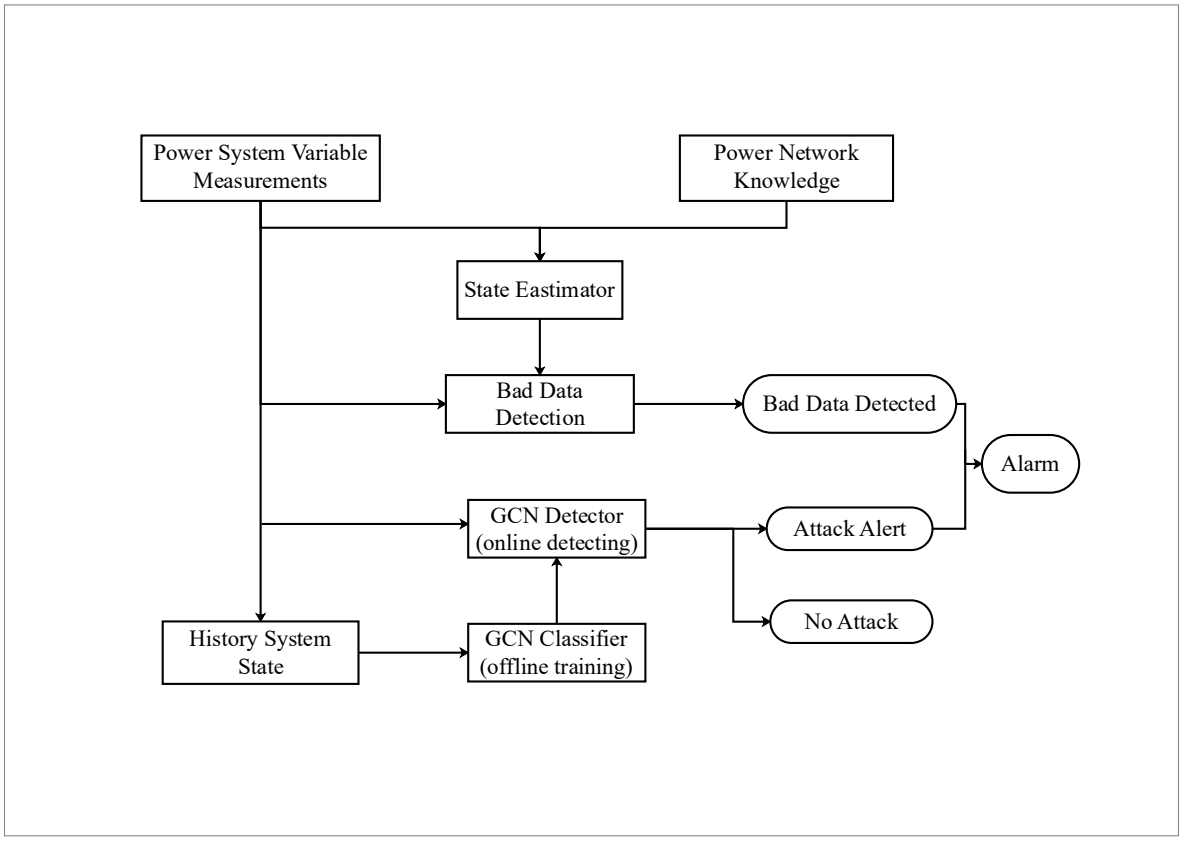



Figure 5 The Flowchart of Detection Mechanism

### 5.3. Algorithm

The operation rule is shown in Algorithm 1. It could be defined as four important procedure: data-preprocessing, model construction, loss function definition and train/test module. Two files are given to the graph neural network for building one system structure. One includes the connect information of a transmission line's nodes, the other one includes features of a node. These features are summarized by the strategy mentioned in last section and the nodes are divided into two classes to verify whether the node is attacked.

The main purpose of data pre-processing is to convert the original file into a readable python readable file, change the diagraph structure of power system into an undirected

graph and build a symmetric adjacency matrix. To reduce the computation complexity and eliminate the effect of singularity, both feature and adjacency matrix needs to be normalized. The GCN model includes two hidden layers. The layers used in the thesis is 3-128-128-2. The first layer is trained by function $relu()$ which is defined by

$$ReLU(x) = \begin{cases} x & if\ x > 0 \\ 0 & if\ x \leq 0 \end{cases} \tag{34}$$

The second layer is trained by function $log\_softmax()$ which is defined by

$$\log\ \_softmax(x_i) = log\frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{35}$$

The loss function of GCN include two parts: one is for classification loss; another one is about weight regularization. $weigh\_decay$ presents the regularization coefficient. The definition of regularization is given at the time of defining optimizer. A $NLLloss()$ is defined synergy with the second activation function which is calculated by

$$loss(\boldsymbol{x_i}, \boldsymbol{y_i}) = (\boldsymbol{x_i} - \boldsymbol{y_i})^2 \tag{36}$$

The result got by $NLLloss()$ goes backward to the first layer to update weights and bias given randomly.

Table 2 Algorithm of Proposed Mechanism

| Algorithm 1: Graph convolution network algorithm |
|---|
| **Input:** Graph $G(V, E)$ including degree matrix $D$ and adjacency matrix $A$; input features $\{\boldsymbol{X}_v, \forall v \in V\}$; |
| **Output:** Classified nodes |

**Initialize** randomly chosen weight matrix $W$ and bias; Set initial training rate, the number of hidden units, dropout rate and regularization parameter $weigh\_decay$.

**1** Build symmetric adjacency matrix and convert the diagraph to undirected graph via building symmetric

**2** Normalize feature matrix $\boldsymbol{X}_v$ and adjacency matrix $A$

**3** Define optimizer, the training dataset, evaluation dataset and test dataset

**4** Architect GCN model using $relu()$ for the first layer and $log\_softmax()$ for the second layer:

$x \leftarrow Relu(\boldsymbol{X}_v, A)$

$x \leftarrow dropout(x)$

$x \leftarrow log\_softmax(x, A)$

**5** Calculate loss function $NLLloss()$

$$loss \leftarrow NLLloss(result)$$

**6** Optimize weights and bias and output the final result

**7** Train and test samples

**for** $epoch$ **in** $ranges$

$\boldsymbol{H}^{(l)} \leftarrow \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l-1)}W)$

$\mathbf{Z} \leftarrow H^{(l)}$

$loss \leftarrow nll\_loss(Z, A)$

Update parameter (W and bias) with losses

**end**

# Chapter 6

# Experiment and Case Study

The proposed method performance of each anomaly node is directly related to the overall performance of the proposed detection framework. In this chapter, the 14-bus test system and 300-bus teste system are implemented to evaluate the detection performance of the proposed detection mechanism. The detailed confusion matrix works as an indicator to test the result of the GCN-based detector.

## 6.1. Evaluation Indicator

Accuracy, precision, recall and F1-measure are chosen as the evaluation indicators in the thesis.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{37}$$

$$precision = \frac{TP}{TP + FP} \tag{38}$$

$$recall = \frac{TP}{TP + FN} \tag{39}$$

$$F1 - measure = 2 \cdot precision \cdot \frac{recall}{precision + recall} \tag{40}$$

where $TP$ (True Positive) presents true positive, which stands for the number of correctly detected abnormal samples in the testing dataset. The same as $TP$, $TN$ (True Negative) represents the number of correctly detected normal samples. $FP$ (False Positive) is the number of normal samples misclassified as abnormal samples by the proposed method, and $FN$ (False Negative) represents the number of misclassified abnormal samples. The definition of recall describes the sensitivity of the model to the positive case category. The F1 score calculates the harmonic mean of precision and recall.

## 6.2. Experiment Analysis

### *6.2.1. Data Sets*

The 118-bus test system which is shown in Figure 6 and 300-bus test system are used in MATLAB to simulate the power system and get enough data for training and testing the anomaly nodes. Attackers might modify the PMU measurements in different ways depending on their purpose. For example, if the attacker would like to imitate a short-circuit fault, the bias from normal values need to be large enough. Opposite to this, if the attackers want to theft power, the bias need to be moderate to evade the power system detection mechanism. During the process of simulation using MATLAB, the possibility of all PMU

measurements is same no matter they measure which part of a system, and both magnitudes and angles could be modified. The attack of magnitudes is realized by adding random number generated by Gaussian distribution function with variation equals to 0 to the original value. The attack of angles is realized by intercepting random degree between -90 to 90. The threshold of PMU needs to be defined after simulation to detect which PMU is attacked by adversaries.

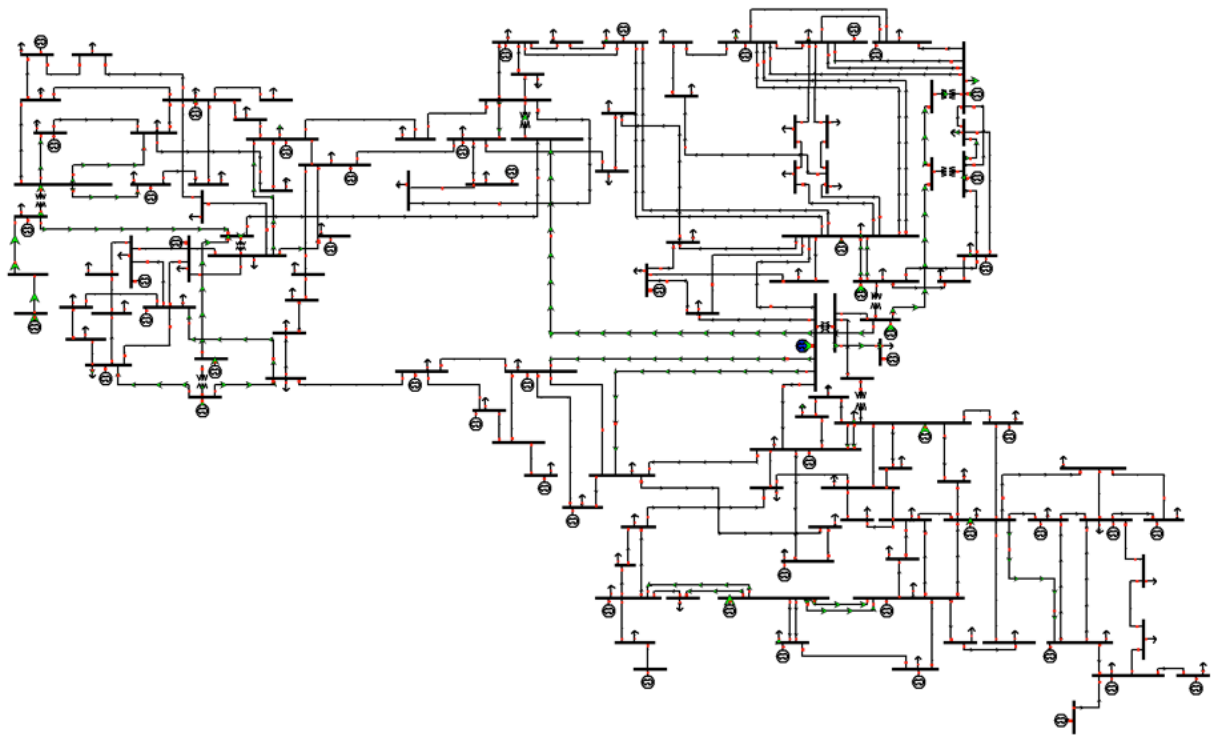


Figure 6 IEEE-118 Test System [61]

### *6.2.2. Parameter Settings*

Pytorch is implemented to create the GCN structure. The number of hidden units is a very important parameter of the GCN model because it will affect the prediction precision.

Finally, the hidden layers are chosen by experimenting with different units and select the optimal one. In the experiment, for the specific data, the number of hidden units is chosen as [16, 32, 64, 128, 168] and analyze the change of precision and accuracy. The result is shown in Figure 7 and Figure 8. The horizontal axis represents the number of hidden units and the vertical axis represents the change of different metrics (accuracy and precision). It can be seen that the precision and accuracy is best when hidden units are 128.

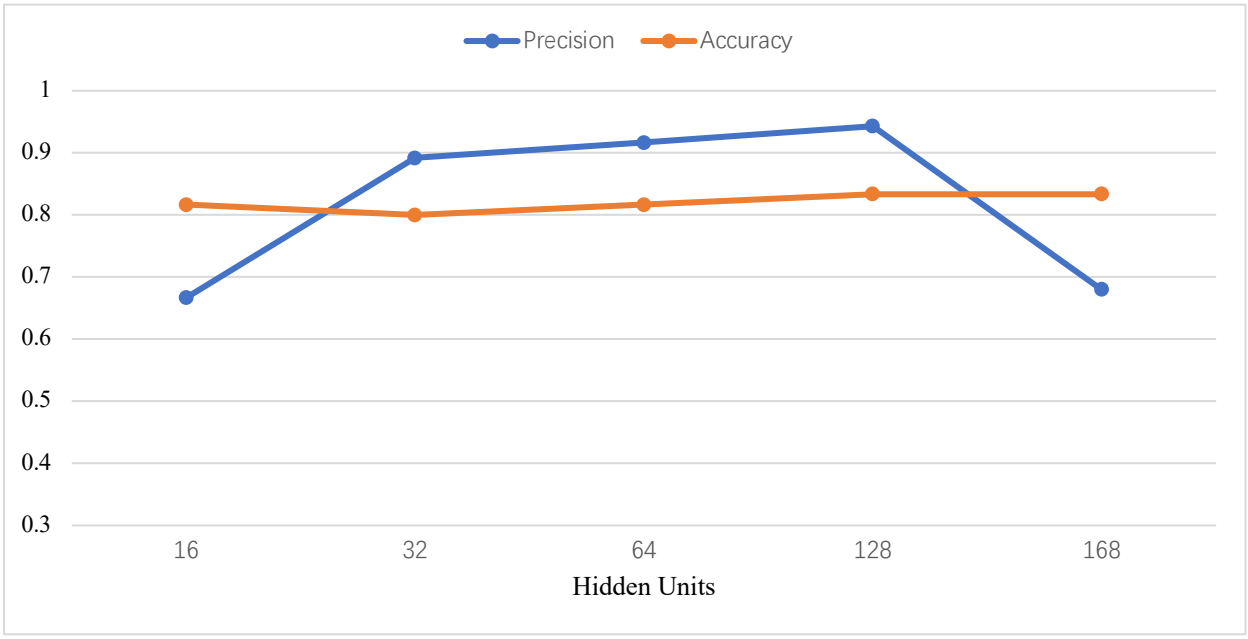



Figure 7 Accuracy VS Hidden Units for Testing Samples

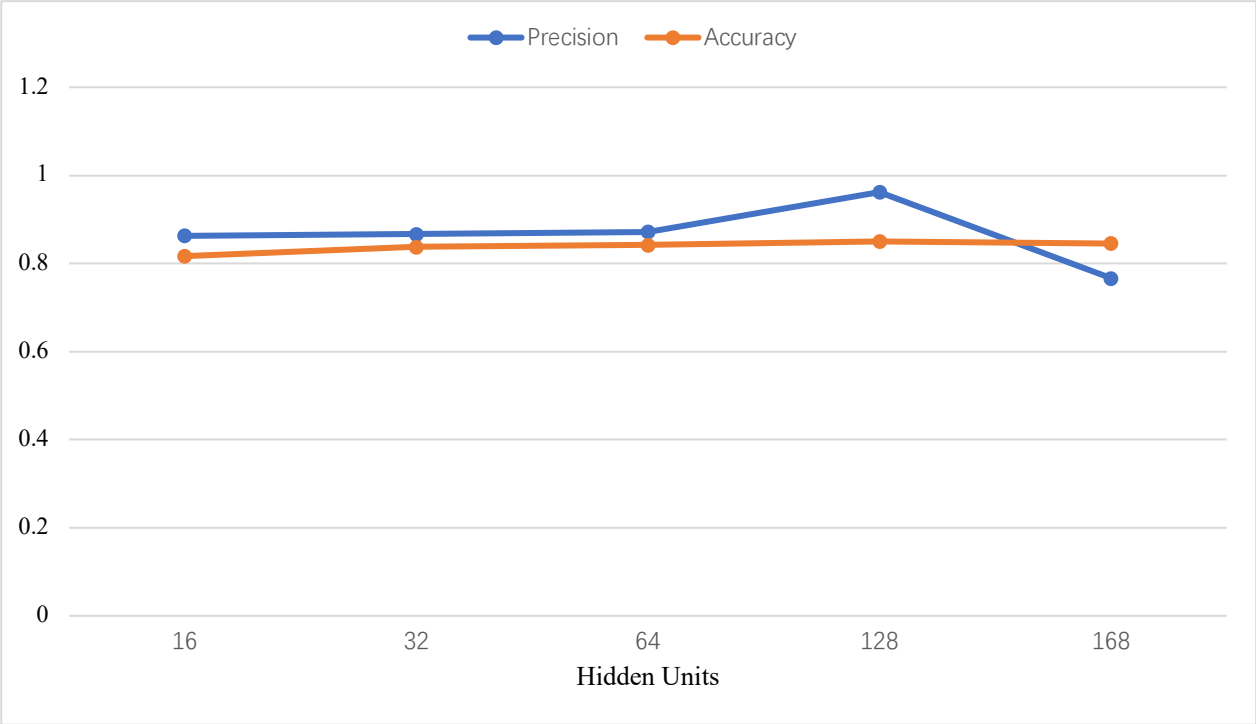



Figure 8 Accuracy VS Hidden Units for Train Samples

The loss and accuracy figures of training set are shown in Figure 9 and 10. The line becomes stable at epoch 200. As a result, the epoch is chosen to 1000.

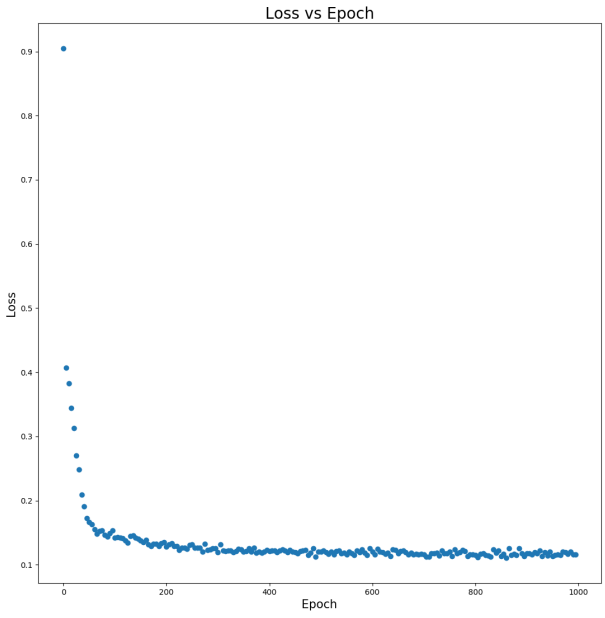



Figure 9 Epoch versus Loss

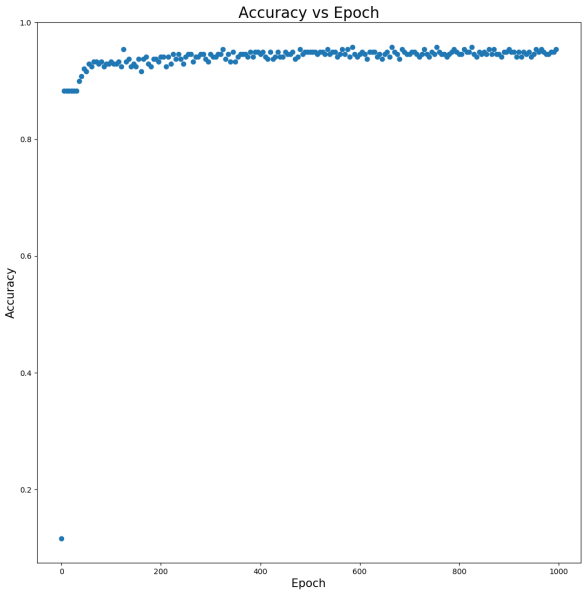



Figure 10 Epoch versus Accuracy

80% and 20% of samples are chose as training dataset and testing dataset respectively.

The parameters of GCN are set as follows:

Table 3 The Hyperparameters of Graph Convolution Network

| Hyperparameters | Setting number |
|---|---|
| Learning rate | 0.01 |
| Training epoch | 1000 |
| The number of hidden units | 128 |
| Weight decay | 5e-4 |

### 6.3. Detection Performance

*6.3.1. Case Study I*

The performance of the proposed method is studied at first with the pre-mentioned hyperparameters of GCN. The IEEE-118 bus and IEEE-300 bus test power system are assessed, and the simulation results are shown in Table 4.

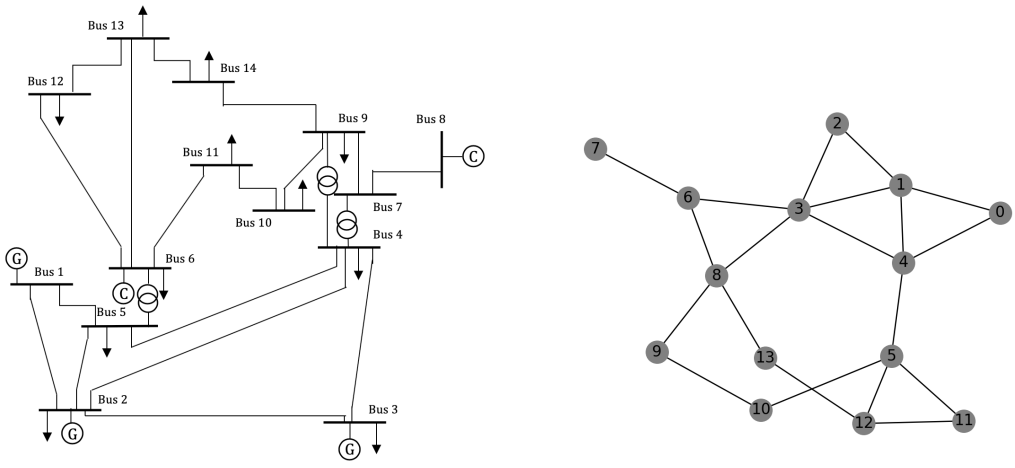
It can be observed that the proposed method can develop a satisfactory detection accuracy from the table. The detection accuracy is about 90%-95% for both testing cases and training cases of IEEE-118 bus and IEEE-300 bus, and the mechanism works slightly better on 118-bus system because of the less complex topology. The accuracy of 300 bus is higher compared with 118 bus because 300 bus has more detailed information compared with 118 bus. It could be seen that the precision of both test systems is around 80% or higher in the testing cases. The overfitting is under control with the regularization and loss function introduced into the algorithm based on the comparison of detection results of testing and training cases. The detection time and training time of both test systems are also summarized in the table. The detection time could represent the overhead introduced by the proposed detection method, and the training time describes the complexity of the adopted GCN. The detection time increased a lot accompanying with the growing scalability of power

systems, therefore, the process is typically conducted offline. The precision and recall increase as the complexity of power system boost which means the graph convolution network works better under a huge power system.

Table 4 The Detection Performance of the Proposed Method

| Indicators | | 118 bus | 300 bus |
|---|---|---|---|
| Training cases | Accuracy | 0.9545 | 0.9577 |
| | Precision | 0.8636 | 0.9700 |
| | Recall | 0.9048 | 0.9826 |
| | F1 | 0.8837 | 0.9762 |
| Testing cases | Accuracy | 0.9070 | 0.9250 |
| | Precision | 0.8182 | 0.9444 |
| | Recall | 0.8182 | 0.9714 |
| | F1 | 0.8182 | 0.9577 |
| Training Time | | 10.0694s | 14.6751s |
| Detection Time | | 0.0114s | 0.0147s |

*6.3.2. Case study II*




(a) (b)

Figure 11 IEEE-14 Test Power System and Topology by Python

The IEEE-14 bus test power system is shown in Figure 11 (a). The system is drawn to its topology format using Pytorch as shown in Figure11 (b). The counting method of Python is from 0 to the specific one. Therefore, number "0" is used to present bus 1. The nodes installed PMUs is bus 2, bus 9, bus 10 and bus 13 which could realize observability of the system . In this experiment, bus 2 and bus 13 is defined as abnormal nodes, and the other two buses are defined as normal states. The proposed method begins to classify nodes based on the power system structure and adjacency matrix constructed by graph convolution neural network. And the result is shown as Figure 12. Red circles represent susceptible buses including abnormal PMUs. This function is used to help redetect normal buses which evade the detection of bad data detection module.

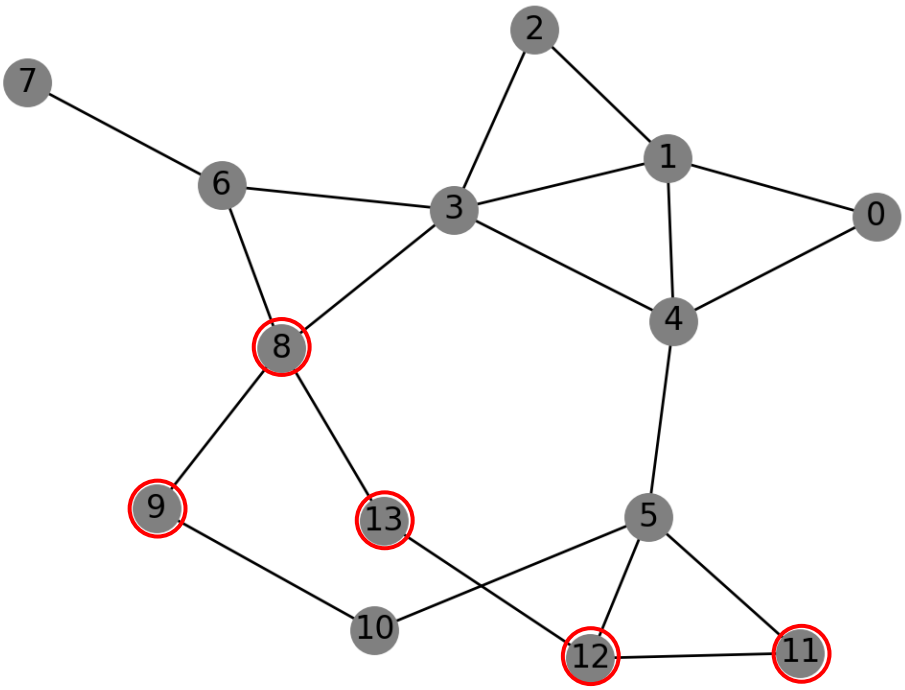



Figure 12 The Test Result of IEEE-14 Buses Power System

# Chapter 7

# Conclusion and Future Work

In this thesis, a power system is considered as a topology network structure. The reality that locations installing PMU is limited because of the high cost. And graph convolution neural network could utilize the graph structure to extract features and classify node. Based on these, the thesis combines the bad data detection module with a semi-supervised graph deep learning method GCN. The basic theory of graph convolution neural network is introduced at first with symmetric normalized Laplacian. Features of a system node is defined based on the power flow formulation. And then a physical synergy with state estimator to detect false data injection is created. IEEE-118, IEEE-300 test system is implemented to verify the efficiency of the proposed method. In both test systems, the proposed method could realize excellent attack detection performance. IEEE-14 bus is also used to show the result of node classification. That's to say, once the node connected with PMU violates the rule specifications, the PMU will be defined as abnormal state. These neighbor nodes could be classified to abnormal states based on the graph convolution neural network theory. The method is used to help bad data detection module find undetected abnormal nodes and stealthy attacks. Each node changes their own states

affected by neighbor nodes or distant relatives to find the final balance. The closer nodes are, the more effect those adjacent nodes give to a specific node.

Graph convolution neural network is a promising method with many aspects to improve. The whole topology structure of a system needs to be drawn in advance to realize the purpose of nodes classification. Therefore, future research of this thesis could be divided into two parts. The first one will focus on finding methods to detect false data injection in parts of a system. Graph attention network (GAT) gives another way to analysis the topology structure of power system. The advantage of GAT is that only partial structure is considered. Therefore, GAT might be implemented for future research. The second one is about to consider the impedance of transmission lines into the structure. Besides, graph convolution network utilizes spatial structure of power system. Sequential data could be introduced into the detection mechanism using recursive neural network or gated convolution network.

# References


[1] L. Monostori. "Cyber-physical production systems: Roots, expectations and r&d challenges," Procedia CIRP, 2014.

[2] G. Simard, "IEEE Grid Vision 2050", IEEE PES, 2013.

[3] R. Kudo, "Detection and Mitigation of False Data Injection Attacks for Secure Interactive Networked Control System," 2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR), Shenyang, 2018, pp. 7-12.

[4] G. Liang, J. Zhao, F. Luo, S. R. Weller and Z. Y. Dong, “A Review of False Data Injection Attacks Against Modern Power Systems,” IEEE Transactions on Smart Grid , vol. 8, no. 4, pp. 1630-1638, 2017.

[5] T. L. Hardy, Software and System Safety: Accidents, Incidents, and Lessons Learned, AuthorHouse, 2012.

[6] “Analysis of the Cyber Attack on the Ukrainian Power Grid,” Electricity Information Sharing and Analysis Center, 2016.

[7] Z.-H. Pang, G.-P. Liu, D. Zhou, F. Hou, and D. Sun, "Two-channel false data injection attacks against output tracking control of networked system," IEEE Transactions on Industrial Electronics, vol. 63, no. 5, pp. 3242-3251, 2016.

[8] C. Konstantinou and M. Maniatakos, "A Case Study on Implementing False Data Injection Attacks Against Nonlinear State Estimation," in Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy, 2016.

[9] R. Macwan, C. Drew, P. Panumpabi, A. Valdes, N. Vaidya, P. Sauer and D. Ishchenko, “Collaborative defense against data injection attack in IEC61850 based smart substations,” in IEEE Power and Energy Society General Meeting (PESGM), 2016.

[10] G. Liang, J. Zhao, F. Luo, S. R. Weller and Z. Y. Dong, “A Review of False Data Injection Attacks Against Modern Power Systems,” IEEE Transactions on Smart Grid , vol. 8, no. 4, pp. 1630-1638, 2017.

[11] T. T. Kim and H. V. Poor, “Strategic protection against data injec- tion attacks on power grids,” IEEE Trans. Smart Grid, vol. 2, no. 2, pp. 326–333, Jun. 2011.

[12] J. Wang, D. Shi, Y. Li, J. Chen, H. Ding and X. Duan, "Distributed Framework for Detecting PMU Data Manipulation Attacks With Deep Autoencoders," in IEEE Transactions on Smart Grid, vol. 10, no. 4, pp. 4401-4410, July 2019.

[13] S. Pal, B. Sikdar and J. H. Chow, "Classification and Detection of PMU Data Manipulation Attacks Using Transmission Line Parameters," in IEEE Transactions on Smart Grid, vol. 9, no. 5, pp. 5057-5066, Sept. 2018.

[14] Y. Liu, P. Ning, and M. K. Reiter, “False data injection attacks against state estimation in electric power grids,” in Proc. 16th ACM Conf. Comput. Commun. Security (CCS), Chicago, IL, USA, 2009, pp. 21–32. [Online]. Available: http://doi.acm.org/10.1145/1653662.1653666

[15] H. Sandberg, A. Teixeira, and K. H. Johansson, “On security indices for state estimators in power networks,” in Preprints 1st Workshop Secure Control Syst. CPSWEEK, Stockholm, Sweden, 2010, pp. 1–6.

[16] G. Hug and J. A. Giampapa, “Vulnerability assessment of AC state es- timation with respect to false data injection cyber-attacks,” IEEE Trans. Smart Grid, vol. 3, no. 3, pp. 1362–1370, Sep. 2012.

[17] M. Netto, J. Zhao, and L. Mili, “A robust extended kalman filter for power system dynamic state estimation using pmu measurements,” in IEEE Power and Energy Society General Meeting (PESGM), July 2016, pp. 1–5.

[18] S. Wang, W. Gao, and A. P. S. Meliopoulos, “An alternative method for power system dynamic state estimation based on unscented transform,” IEEE Transactions on Power Systems, vol. 27, no. 2, pp. 942–950, May 2012.

[19] J. Zhao, M. Netto, and L. Mili, “A robust iterated extended kalman filter for power system dynamic state estimation,” IEEE Transactions on Power Systems, vol. 32, no. 4, pp. 3205–3216, July 2017.

[20] J. Zhao, A. Gomez-Exposito, M. Netto, L. Mili, A. Abur, V. Terzija, I. Kamwa, B. C. Pal, A. K. Singh, J. Qi, Z. Huang, and A. P. S. Meliopoulos, “Power system dynamic state estimation: Motivations, definitions, methodologies and future work,” IEEE Transactions on Power Systems, vol. IEEE early access, pp. 1–1, 2019.

[21] A. F. Taha, J. Qi, J. Wang, and J. H. Panchal, “Risk mitigation for dynamic state estimation against cyber attacks and unknown inputs,” IEEE Transactions on Smart Grid, vol. 9, no. 2, pp. 886–899, March 2018.

[22] Y. Chakhchoukh and H. Ishii, "Enhancing Robustness to Cyber-Attacks in Power Systems Through Multiple Least Trimmed Squares State Estimations," in IEEE Transactions on Power Systems, vol. 31, no. 6, pp. 4395-4405, Nov. 2016.

[23] W. Ao, Y. Song, and C. Wen, “Adaptive cyber-physical system attack detection and reconstruction with application to power systems,” IET Control Theory & Applications, vol. 10, no. 12, pp. 1458–1468, 2016.

[24] H.M.Merrill,F.C.Schweppe,Baddatasuppressioninpowersystemstaticstate estimation, IEEE Trans. Power Appar. Syst. (6) (1971) 2718–2725.

[25] T.V. Cutsem, M. Ribbens-Pavell, L. Mili, Hypothesis testing identification: A new method for bad data analysis in power system state estimation, IEEE Trans. Power Appar. Syst. (11) (1984) 3239–3252.

[26] 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.

[27] Y. Chakhchoukh, S. Liu, M. Sugiyama, and H. Ishii, “Statistical outlier detection for diagnosis of cyber attacks in power state estimation,” in Proc. IEEE PES General Meeting, pp. 1–5, July 2016.

[28] Y. He, G. J. Mendis, and J. Wei, “Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent me- chanism,” IEEE Transactions on Smart Grid, 2017.

[29] S. Marsland, Machine learning: an algorithmic perspective. CRC press, 2015.

[30] A. S. Musleh, G. Chen and Z. Y. Dong, "A Survey on the Detection Algorithms for False Data Injection Attacks in Smart Grids," in IEEE Transactions on Smart Grid. doi: 10.1109/TSG.2019.2949998.

[31] A. Monticelli, “Electric power system state estimation,” Proceedings of the IEEE, vol. 88, no. 2, pp. 262 - 282, 2000.

[32] M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni and H. V. Poor, "Machine Learning Methods for Attack Detection in the Smart Grid," in IEEE Transactions on Neural Networks and Learning Systems, vol. 27, no. 8, pp. 1773-1786, Aug. 2016.

[33] M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni, and H. V. Poor, “Smarter security in the smart grid,” in Proc. 3rd IEEE Int. Conf. Smart Grid Commun., Tainan, Taiwan, Nov. 2012, pp. 312–317.

[34] Y. Li, Y. Wang and S. Hu, "Online Generative Adversary Network Based Measurement Recovery in False Data Injection Attacks: A Cyber-Physical Approach," in IEEE Transactions on Industrial Informatics, vol. 16, no. 3, pp. 2031-2043, March 2020.

[35] A. Ayad, M. Khalaf and E. El-Saadany, “Detection of False Data Injection Attacks in Automatic Generation Control Systems Considering System Nonlinearities,” in IEEE Electrical Power and Energy Conference (EPEC), 2018 .

[36] J. G. Sreenath, A. Meghwani, S. Chakrabarti, K. Rajawat and S. C. Srivastava, “A recursive state estimation approach to mitigate false data injection attacks in power systems,” in IEEE Power & Energy Society General Meeting, 2017.

[37] M. G. Kallitsis, S. Bhattacharya, S. Stoev and G. Michailidis, “Adaptive statistical detection of false data injection attacks in smart grids,” in IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2016.

[38] R. Moslemi, A. Mesbahi and J. M. Velni, “A Fast, Decentralized Covariance Selection-Based Approach to Detect Cyber Attacks in Smart Grids,” IEEE Transactions on Smart Grid, vol. 9, no. 5, pp. 4930 - 4941, 2018.

[39] D. Wang, X. Wang, Y. Zhang and L. Jin, “Detection of power grid disturbances and cyber-attacks based on machine learning,” Journal of Information Security and Applications, vol. 46, pp. 42-52, 2019.

[40] Y. Wang, M. M. Amin, J. Fu and H. B. Moussa, “A Novel Data Analytical Approach for False Data Injection Cyber-Physical Attack Mitigation in Smart Grids,” IEEE Access, vol. 5, pp. 26022 - 26033, 2017.

[41] R. Deng, G. Xiao, R. Lu, H. Liang and A. V. Vasilakos, "False Data Injection on State Estimation in Power Systems—Attacks, Impacts, and Defense: A Survey," in IEEE Transactions on Industrial Informatics, vol. 13, no. 2, pp. 411-423, April 2017.

[42] D. Zhu, T. Zhang and G. Mao, "Back-propagation artificial neural networks for water supply pipeline model," in Tsinghua Science and Technology, vol. 7, no. 5, pp. 527-531, Oct. 2002.

[43] F. Sichao, L. Weifeng, L. Shuying and Z. Yicong, "Two-order graph convolutional networks for semi-supervised classification," in IET Image Processing, vol. 13, no. 14, pp. 2763-2771, 12 12 2019.

[44] F. R. Chung and F. C. Graham, Spectral graph theory. American Mathematical Soc., 1997, no. 92.

[45] Lin, G., Wang, J., Liao, K., Zhao, F., & Chen, W. (2020, March 4). Structure Fusion Based on Graph Convolutional Networks for Node Classification in Citation Networks. Retrieved from https://www.mdpi.com/2079-9292/9/3/432/htm

[46] P. Gong and L. Ai, "Neighborhood Adaptive Graph Convolutional Network for Node Classification," in IEEE Access, vol. 7, pp. 170578-170588, 2019.

[47] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83-98, May 2013.

[48] Hammond, D.K., Vandergheynst, P., Gribonval, R.: 'Wavelets on graphs via spectral graph theory', Appl. Comput. Harmon. A, 2011, 30, (2), pp. 129–150.

[49] Kipf, T. N., & Welling, M. (2017, February 22). Semi-Supervised Classification with Graph Convolutional Networks. Retrieved from https://arxiv.org/abs/1609.02907

[50] Beibei Li, Rongxing Lu, Wei Wang, Kim-Kwang Raymond Choo, "Distributed host-based collaborative detection for false data injection attacks in smart grid cyber-physical system," Journal of Parallel and Distributed Computing, Volume 103, 2017.

[51] J. J. Q. Yu, Y. Hou and V. O. K. Li, "Online False Data Injection Attack Detection With Wavelet Transform and Deep Neural Networks," in IEEE Transactions on Industrial Informatics, vol. 14, no. 7, pp. 3271-3280, July 2018.

[52] R. Xu, R. Wang, Z. Guan, L. Wu, J. Wu and X. Du, "Achieving Efficient Detection Against False Data Injection Attacks in Smart Grid," IEEE Access, vol. 5, pp. 13787 - 13798, 2017.

[53] B. Li, G. Xiao, R. Lu, R. Deng and H. Bao, "On Feasibility and Limitations of Detecting False Data Injection Attacks on Power Grid State Estimation Using D-FACTS Devices," IEEE Transactions on Industrial Informatics, in press, 2019.

[54] H. Zhao, H. Liu, W. Hu and X. Yan, "Anomaly detection and fault analysis of wind turbine components based on deep learning network," Renewable Energy, vol. 127, pp. 825-834, 2018.

[55] M. N. Kurt, O. Ogundijo, C. Li and X. Wang, "Online Cyber-Attack Detection in Smart Grid: A Reinforcement Learning Approach," IEEE Transactions on Smart Grid, pp. 1-12, in press.

[56] S. Ntalampiras, "Fault Diagnosis for Smart Grids in Pragmatic Conditions," IEEE Transactions on Smart Grid, vol. 9, no. 3, pp. 1964 - 1971, 2018.

[57] S. Pan, T. Morris and U. Adhikari, “Classification of Disturbances and Cyber-Attacks in Power Systems Using Heterogeneous Time-Synchronized Data,” IEEE Transactions on Industrial Informatics, vol. 11, no. 3, pp. 650 - 662, 2015.

[58] M. Zanetti, E. Jamhour, M. Pellenz, M. Penna, V. Zambenedetti and I. Chueiri, “A Tunable Fraud Detection System for Advanced Metering Infrastructure Using Short-Lived Patterns,” IEEE Transactions on Smart Grid, vol. 10, no. 1, pp. 830 - 840, 2019.

[59] J. J. Q. Yu, Y. Hou and V. O. K. Li, "Online False Data Injection Attack Detection With Wavelet Transform and Deep Neural Networks," in IEEE Transactions on Industrial Informatics, vol. 14, no. 7, pp. 3271-3280, July 2018.

[60] A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar and S. Mishra, “Decision Tree and SVM-Based Data Analytics for Theft Detection in Smart Grid,” IEEE Transactions on Industrial Informatics, vol. 12, no. 3, pp. 1005 - 1016, 2016.

[61] Electric Grid Test Case Repository. (n.d.). Retrieved from https://electricgrids.engr.tamu.edu/electric-grid-test-cases/ieee-118-bus-system/