IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676, p-ISSN: 2320-3331, Volume 18, Issue 5 Ser. I (Sept. – October. 2023), PP 01-08
www.iosrjournals.org

Enhancing Power System Analysis Through Graph Topology

Babita Gupta [1], Nikitha Ch [2], Thanmai M [3], Manogna CH [4], Mounika S [5]
[1] Assistant Professor, BVRIT HYDERABAD College of Engineering for Women, Hyderabad, Telangana, India
[2,3,4,5] BVRIT HYDERABAD College of Engineering for Women, Hyderabad, Telangana, India

Abstract—
Power system analysis is a critical aspect of ensuring the reliable and efficient operation of electrical grids. Traditionally, power system analysis has focused on mathematical models and optimization techniques to analyse the behaviour of individual components and the overall system. However, with the increasing complexity of modern power systems, there is a growing need to incorporate the inherent graph topology of the grid into the analysis process. This paper explores the use of graph theory and graph topology in power system analysis. The graph topology represents the interconnectedness of power system components, including generators, transmission lines, substations, and loads. In recent years, there has been a notable surge of interest in utilizing graph topology analysis for power system analysis. Scientists and scholars have dedicated their efforts to crafting graph-based models, like Graph Neural Networks (GNNs), which effectively capture the intricate interconnections among various components within the power system. Researchers have developed graph-based models, such as Graph Neural Networks (GNNs), to capture the complex relationships between power system components. In Power flow analysis, we calculate the unknown variables using known variables using electrical formulas. Similarly, Graph Neural Networks (GNNs) employ the prediction of unmeasured variables using measured variables to effectively compute the power flow. In this work, the performance of GNN will be compared with the traditional neural network.

Keywords— *Neural Networks, Power Flow, Fully Connected Neural Network, Graph neural network*

Date of Submission: 23-08-2023 Date of acceptance: 03-09-2023

I. INTRODUCTION

Electric power system analysis is a fundamental topic extensively covered in the field of power system analysis. Finding the system's known and unknown variables is the first step in solving difficulties with power flow. As illustrated in Table 1, buses are divided into three categories based on these factors: slack, generation, and load buses [1].

The analysis of electric power flow typically involves the identification and resolution of a set of non-linear algebraic equations using iterative numerical analysis methods. Over time, the power flow problem has been investigated using various techniques within the realm of iterative numerical analysis. The Newton Raphson Method, a potent technique for solving non-linear algebraic equations, is one of the approaches to load flow analysis. Compared to the Guass-Seidal approach, it operates more quickly and is almost always guaranteed to converge [3].

Table-I: Type of buses in the power flow problem.

Bus type	Voltage (|Vi| ∠δi)		Real power	Reactive power
	Magnitude	Angle	Net (Pi)	Net (Qi)
Slack/ Swing	Specified	Specified	Unknown	Unknown
Generator/ Regulated/ PV	Specified	Unknown	Specified	Unknown
Load/PQ	Unknown	Unknown	Specified	Specified

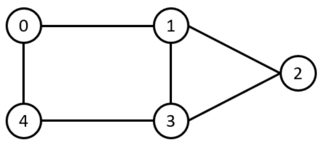
Artificial Neural Network (ANN) algorithm [5][2], Fully Connected Neural Networks are applied to power flow calculation. Fully Connected Neural Networks (FCNN) are used in the study [6] in order to imitate the results of AC Power A fully connected neural network makes use of the node's information which is not adjacent to the network. Because of this trait, it is used to imitate the results of AC Power Flow and hence the FCNN will frequently overfit.

[18][19] Graph neural network (GNN) technology offers an innovative solution to the AC Power Flow (ACPF) problem [7]. Buses and lines can be viewed as the nodes and edges respectively, of a power grid. While the line features may include line current and line resistance, the node features are voltage, angle of voltage, real power, and imaginary power [8].

The paper contrasts the performance of many Neural Network models on AC Power Flow and inferences that the Graph Neural Network (GNN) model surpasses the Fully Connected Neural Network (FCNN) model. This paper has the following organization brief overview of the graph and graph neural networks, and fully connected neural networks, provided in Section 2 and Section 3 respectively. How the experiment is carried out is explained in Section 4. The outcome and analysis are discussed in Section 5 and in Section 6 the conclusions are drawn.

II. GRAPH

A graph, as a data structure, consists of a countable set of nodes and edges, which can be either ordered or unordered pairs, depending on whether the graph is oriented or unoriented. A graph is a graphic depiction of a group of objects where related object pairs are connected. Vertices or nodes of a graph are connected to one another, and the connections between these vertices are referred to as edges. It consists of an ordered pair of sets (V, E). Graph operations, types of graphs, and graph representations are shown in [9]. One of the common ways to represent graphs is an adjacency matrix. Figure 1 illustrates an example of an adjacency matrix representation.

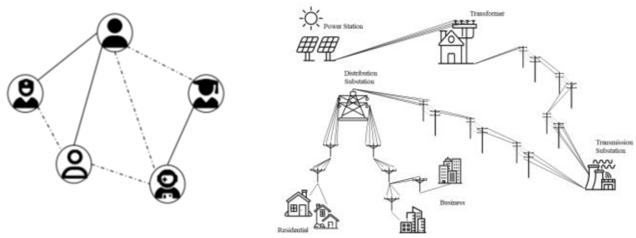


$$\begin{array}{c|ccccc|} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 2 & 0 & 1 & 0 & 1 & 0 \\ 3 & 0 & 1 & 1 & 0 & 1 \\ 4 & 1 & 1 & 0 & 1 & 0 \\ \hline \end{array}$$

Figure 1: Adjacency Matrix Representation of a given Graph

The world fundamentally represents itself through graphs. Graphs are commonly used to represent data that is generated spontaneously. By observing figure 2, it can be concluded that the power grids and social networks are similar to graph structure.

Figure 2: Social network and electrical power grids are all graphs



Data in Euclidean space is referred to as Euclidean data. Euclidean data examples Non-euclidean data, which deviates from the principles of Euclidean geometry. Euclidean data examples include text and photos. In the realm of non-euclidean data, the straight line is not always the shortest route between two points. if Euclidean distance is used as the metric, entities that are comparable to one another may not always be close. Since it is impossible to determine the distance between nodes physically using the Pythagorean theorem, graph data is non-Euclidean data [10].

III. POWER FLOW

Power flow analysis, also known as load flow analysis, is a fundamental method used in electrical engineering to study and analyse the steady-state behaviour of power systems. It is utilized to ascertain diverse electrical parameters, including voltage magnitudes, phase disparities, active and reactive power transmissions, and network losses.

The power flow analysis is conducted to assess the performance and reliability of power systems, identify potential issues, and optimize their operation. It helps engineers understand how power is distributed and utilized within a network, ensuring that generation, transmission, and distribution capacities are efficiently allocated.

Here is a basic overview of how power flow analysis works:

Network Representation: The power network is depicted as a network structure comprising of buses (nodes) interconnected by transmission lines, transformers, and additional apparatus. Each bus is characterized by its voltage magnitude and phase angle.

Formulating Equations: Based on the network topology and electrical characteristics of the components, a set of mathematical equations is derived to describe the power flow within the system. These equations are typically based on Kirchhoff's laws, Ohm's law, and power equations.

Initialization: The power flow analysis begins by initializing the system. Initial values for voltage magnitudes and phase angles are assigned to all buses. In general, voltage measurements are commonly considered as 1 per unit (p.u.), while phase angles are frequently set to zero as a standard practice.

Iterative Process: The power flow equations are solved iteratively to calculate the unknown variables (voltage measurements and Angular phase) at each bus. The iterative process involves updating the variables based on the calculated values from the previous iteration until convergence is achieved. Convergence occurs when the difference between successive iterations falls below a specified tolerance level.

Active and Reactive Power Flows: Once the voltage magnitudes and phase angles are determined, the active and reactive power flows on the transmission lines and at each bus can be computed using the power flow equations. These calculations provide information about power losses, voltage drops, and loading conditions within the system

Analysis and Optimization: The outcomes derived from the power flow analysis are utilized for evaluating the system's operational characteristics, detecting potential limitations, and evaluating voltage stability and power quality aspects. Engineers can also use the analysis to optimize the system's operation by adjusting generator setpoints, transformer taps, and other control parameters.

Power flow analysis plays a vital role in the effective planning, operation, and control of power systems. It helps ensure the reliable and efficient delivery of electrical power while maintaining voltage stability and minimizing losses.

In this work, the IEEE 14 bus system will be employed for analysis purposes. Figure 3 shows the IEEE 14-bus system, which only consists of PQ buses and clack buses. The dataset required for this experiment is created using the PyPSA toolbox. The IEEE 14-bus system will be converted into a PyPSA network using the components mentioned in the above section. All required components and their accompanying specifications will be added to the network in accordance with the transformation.

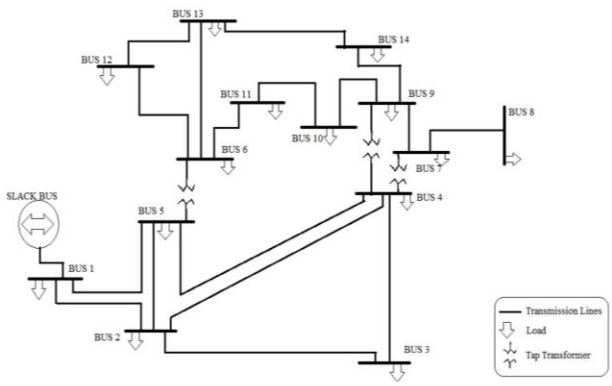


Figure 3: IEEE 14-bus power network

To produce this dataset, the programme iteratively computes power flow. The network is consecutively executed under power flow analysis present in the PyPSA library and the values of PQ bus loads are changed in each iteration by ±50% to produce random and unique data. After the completion of power flow analysis, the known and unknown values are stored. There will be 2000 different power flow data calculations in one dataset as a result. Simulation is performed and 102 datasets will be produced: 100 datasets for training, 1 dataset for testing, and 1 dataset for validation.

IV. NEURAL NETWORKS

Machine Learning, as a subset of Artificial Intelligence, revolves around uncovering latent patterns within data and leveraging these patterns to classify or predict events associated with a given problem. In order to learn data and recognize patterns for the goal of responding to an environment, machine learning algorithms can either be supervised learning or unsupervised learning. In supervised learning, the machine or model is guided and trained using data that has been appropriately labeled, with predetermined correct answers. The algorithm analyzes the provided training data to generate accurate outputs. Models are not supervised using a training dataset while utilizing the machine learning technique known as unsupervised learning. Instead, models themselves find the hidden patterns and insights from the given data [11].

Fully Connected Neural Network

In the domain of artificial neural networks, the Fully Connected Neural Network (FCNN) represents a specific architecture in which neurons are interconnected across multiple layers, enabling comprehensive connectivity throughout the network. The FCNN consists of three primary layer types: the input layer, hidden layers, and the output layer, forming the fundamental components of its architecture. The architecture of FCNN can be seen in Figure 4.

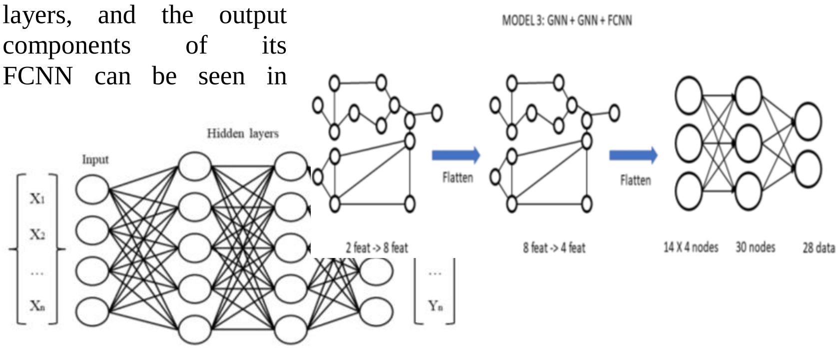


Figure 4. FCNN Architecture

Each neuron in an FCNN layer has a bias term and a weight assigned to each of its connections. During the training process, these weights and biases are iteratively adjusted using optimization algorithms like gradient descent, in order to minimize a predefined loss function that measures the network's prediction error. This process is known as training or learning, and it allows the FCNN to learn the underlying patterns and relationships in the data.

The training process of a Fully Connected Neural Network (FCNN) involves several steps, including data preparation, forward propagation, loss computation, backpropagation, and weight updates. Here is a brief overview of the FCNN training process:

The training data comprises known values of the buses, which are divided into training and validation sets to evaluate the performance of the model. The network receives the input data, which initiates sequential computations layer by layer, with the weights and biases being initially assigned random values. Within the network, individual neurons perform a calculation that involves multiplying the inputs by their corresponding weights, summing them up with appropriate weights, applying an activation function to this sum, and forwarding the resulting output to the subsequent layer. To measure the dissimilarity between the predicted and actual values, the output of the network is compared with the true labels, resulting in the computation of a loss function. To evaluate regression tasks, mean squared error (MSE) is a commonly used loss function that computes the average squared difference between predicted and actual values. For classification tasks, cross-entropy loss is frequently employed to quantify the dissimilarity between predicted and true labels. Following the loss computation, the network undergoes backpropagation, a process that involves calculating gradients of the loss function with respect to the network's weights and biases. By applying the chain rule of calculus, backpropagation systematically propagates the error back through the network, layer by layer. The gradients computed during backpropagation are used to update the network's weights and biases. This whole process is repeated for multiple iterations. Each iteration involves passing the training data through the network, adjusting the weights based on the computed gradients, and updating the loss. Periodically, the performance of the trained model is evaluated using the validation set. If the performance stops improving or starts to degrade, early stopping can be applied to halt the training process and select the best-performing model.

Graph Neural Network

In the realm of data structured as graphs, Graph Neural Networks (GNNs) stand out as a neural network type designed to process and analyze such information. It leverages the graph structure to capture dependencies and relationships between entities in the data. Through the process of message passing, GNNs update the representations of nodes iteratively, incorporating information from neighboring nodes. Message passing is a fundamental operation in Graph Neural Networks (GNNs) that allows information to be propagated between nodes in a graph. GNN layers may basically be divided into these three steps: 1. A message is computed for each of the neighbors of each node in the graph. The node, the neighbor, and the edge between them all influence how messages are sent. 2. A permutation-invariant function is used by each node to aggregate the messages it receives, regardless of the sequence in which they are received. 3. Each node modifies its properties after receiving the messages based on both its existing attributes and the aggregated messages.

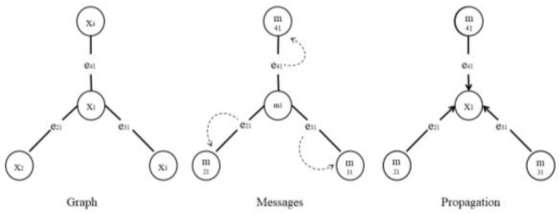


Figure 5. Message passing in Neural Network

Every node in the network receives an update as a result of this method, which occurs synchronously for all nodes in the graph [12].

In power system analysis, Optimal Power Flow (OPF) emerges as a crucial problem, aiming to identify the most efficient generation, transmission, and consumption settings to minimize costs while meeting operational constraints. GNNs can be utilized to model and optimize the power flow in the network, considering the complexities and interdependencies of the system.

V. EXPERIMENT: POWER FLOW ANALYSIS USING GNN

This paper seeks to explore the potential of utilizing a Graph Neural Network on an electrical power grid whose structure is similar to a graph, primarily for the purpose of power flow analysis. In this study, the GNN model makes predication by utilizing both the unknown variables of a power system and the power grid connectivity information in the form of a regression model.

Architecture of Models

In this study, a comparison of three model architectures has been done which are - 1. a combination of two fully connected neural networks, 2. a combination of graph neural network along with fully connected neural network, 3. two layers of graph neural network with a single layer of fully connected neural network. The architecture of the above mention models can be seen in Figure 6. The models are designed carefully to maintain the same complexity level to get accurate results when compared. The total parameters and number of layers are similar to each other as the aim of the comparison is to evaluate all the models and determine the best performance. In the training phase, to ease the gradient descent process the data generated is normalized using mean and standard deviation. [15]. The models employ the ADAM optimizer and all the hidden nodes use the tanh activation function, except for the output layer nodes.

Dataset Generation

The PyPSA library, an open-source Python 3 library, is used to generate the power flow dataset. In Figure 3, we observe the representation of the IEEE 14 bus system network is created, and the power flow calculation is iteratively executed to generate the dataset. For every iteration, the PQ bus loads are varied by ±50% to produce random and unique data. The known and unknown variables are computed and saved. There are 2000 different outputs in a single dataset as a result of the power flow calculation from the PyPSA program. The program is simulated to generate 102 datasets.

Experiment Steps

The experiment involves several steps. Initially, the model receives training using a merged dataset that combines both the train and validation data. The training and validation errors are recorded after each iteration and the best model parameter is saved. The best model parameter, which corresponds to the lowest validation error, is saved. This saved parameter is then applied to 100 different test datasets. Finally, a histogram is plotted to display the distribution of the test errors across the 100 datasets.

The experiment incorporates two distinct error calculation methods: Normalized Root Mean Squared Error (NRMSE) and Mean Square Error (MSE). [16] NRMSE is utilized to verify the validity of predicted values, ensuring they are not merely a result of data averaging.

VI. RESULT AND ANALYSIS

The findings of the experiment conducted on a 14-bus power system are presented in this section. The training process of models 1, 2, and 3 for a training dataset is illustrated in Figure 7. The validation loss and train loss for each iteration in the training phase are represented as orange and blue lines in the figure below. The best model is saved in every iteration and has the lowest validation loss. It is observed that there is no consistency in the lowest validation loss found at the lowest training loss or highest training loss.

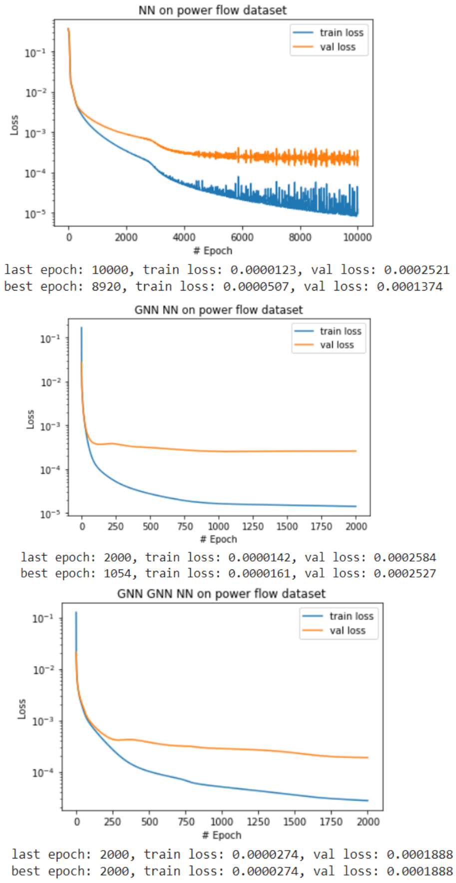


Figure 7: The training process of all three models.

After acquiring the best model, it is tested over 100 test datasets. To observe the distribution of the error values, The histogram is plotted for hundred test errors. Instead of utilizing a histogram, the distribution of the 100 test errors is displayed using the Probability Density Function (PDF) for a simplified visualization that maintains similar objectives. The Probability Density Function graph represents the probability of values along the x-axis, with the total area under the PDF curve being equal to one.

The NRMSE and MSE distributions are plotted for hundred test errors in Figures 8 and 9 respectively. Both error (MSE and NRMSE) types are identical in pattern. In the NRMSE test loss histogram, it can be observed that the x-axis values are less than 1, which indicates that the model prediction is credible.

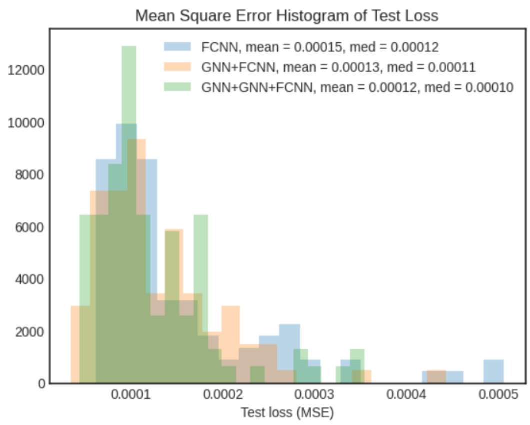


Figure 8: The histogram of the three models of 100 test errors

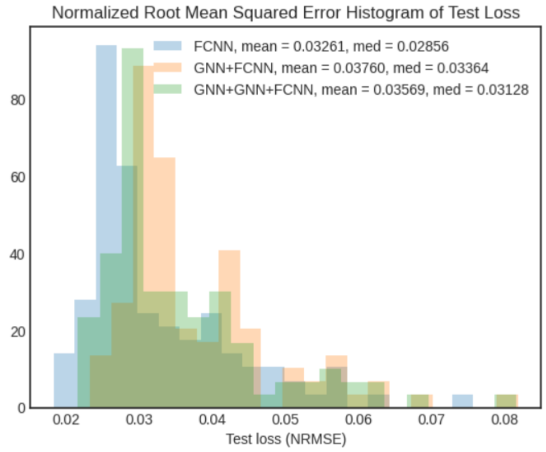


Figure 9: The histogram of the three models of 100 normalized test errors

From the above figures 8 and 9, it can be interpreted that for the less value of test loss, the bins count is more for the GNN model whereas the FCNN model's test loss is spread over a range. Hence, GNN-based models outperform FCNN when having similar trainable parameters. To match the performance with the FCNN model, the GNN model requires fewer parameters as it utilizes parameter sharing.

VII. CONCLUSION

This study offers a thorough analysis of GNN's use in power flow applications. The outcome leads to several inferences.

GNN models effectively utilize the connectivity information of a graph by incorporating the adjacency matrix in their operations. This characteristic allows for parameter sharing, which improves model accuracy, particularly when the size of the training dataset is limited. In scenarios where data availability is restricted, such as when acquiring new data is expensive, GNN outperforms traditional FCNN models.

In conclusion, the utilization of graph topology in power system analysis offers valuable insights and benefits for understanding the behaviour and operation of electrical grids.

Further research and advancements are still needed to fully harness the potential of graph topology analysis in power system analysis. This includes developing more sophisticated graph-based models, refining data representation and pre-processing techniques, and exploring novel applications that leverage the inherent structure of power systems.

References

[1] M. S. Albo Kareem And S. Gheorghe, "Power Flow Analysis In Power System Planning. Case Study," *2019 International Conference On Energy And Environment (Ciem)*, Timisoara, Romania, 2019, Pp. 142-146, Doi: 10.1109/Ciem46456.2019.8937565

[2] Tania B. Lopez-Garcia, José A. Domínguez-Navarro, Power Flow Analysis Via Typed Graph Neural Networks, Engineering Applications Of Artificial Intelligence, Volume 117, Part A, 2023 105567, Issn 0952-1976

[3] Eltamaly, A.M., Sayed, Y., El-Sayed, A.H.M. And Elghaffar, A.N.A., 2018. Optimum Power Flow Analysis By Newton Raphson Method, A Case Study. Ann Fac Eng Hunedoara, 16(4), Pp.51-58s.

[4] M. Fikri, B. Cheddadi, O. Sabri, T. Haidi, B. Abdelaziz And M. Majdoub, "Power Flow Analysis By Numerical Techniques And Artificial Neural Networks," 2018 Renewable Energies, Power Systems & Green Inclusive Economy (Reps-Gie), Casablanca, Morocco, 2018, Pp. 1-5, Doi: 10.1109/Repsgie.2018.8488870.

[5] N. Kumar, R. Wangneo, P. K. Kalra And S. C. Srivastava, "Application Of Artificial Neural Networks To Load Flow Solutions," Tencon '91. Region 10 International Conference On Ec3-Energy, Computer, Communication And Control Systems, New Delhi, India, 1991, Pp. 199-203, Doi: 10.1109/Tencon.1991.712546.

[6] Neel Guha, Z Wang, Matt Wytock, Arun Majumdar, “Machine Learning For Ac Optimal Power Flow”, Climate Change Workshop At Icml,2019

[7] Balthazar Donon, Rémy Clément, Benjamin Donnot, Antoine Marot, Isabelle Guyon, Et Al.. Neural Networks For Power Flow: Graph Neural Solver. Electric Power Systems Research, 2020, 189,Pp.106547. Ffhal-02372741f

[8] Yegnanarayana, B., 2009. Artificial Neural Networks. Phi Learning Pvt. Ltd..

[9] Jayesh Kudase1, Priyanka Bane2, B.E. Student, Dept Of Computer Engineering, A Brief Study Of Graph Data Structure, Fr. Conceicao Rodrigues College Of Engineering, Maharashtra, India1, 2

[10] M. M. Bronstein, J. Bruna, Y. Lecun, A. Szlam, And P. Vandergheynst. Geometric Deep Learning: Going Beyond Euclidean Data. Corr, Abs/1611.08097, 2016.

[11] Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., Aljaaf, A.J. (2020). A Systematic Review On Supervised And Unsupervised Machine Learning Algorithms For Data Science. In: Berry, M., Mohamed, A., Yap, B. (Eds) Supervised And Unsupervised Learning For Data Science. Unsupervised And Semi-Supervised Learning. Springer, Cham. Https://Doi.Org/10.1007/978-3-030-22475-2_1

[12] Zhiqiang Zhong1 · Cheng-Te Li2 · Jun Pang1,3 Hierarchical Message-Passing Graph Neural Networks,

[13] Brown, T., Hörsch, J. And Schlachtberger, D., 2017. Pypsa: Python For Power System Analysis. Arxiv Preprint Arxiv:1707.09913.

[14] Phongtrakul, T., Kongjeen, Y. And Bhumkittipich, K., 2018, July. Analysis Of Power Load Flow For Power Distribution System Based On Pypsa Toolbox. In 2018 15th International Conference On Electrical Engineering/Electronics,Computer, Telecommunications, And Information Technology (Ecti-Con) (Pp. 764-767). Ieee.

[15] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, And L. Shao. Normalization Techniques In Training Dnns: Methodology, Analysis, And Application. Corr, Abs/2009.12836, 2020.

[16] D. K. Molzahn And I. A. Hiskens. Convex Relaxations Of Optimal Power Flow Problems: An Illustrative Example. Ieee Trans. Power Syst.,63(5):650–660, May 2016

[17] P. Von Hippel. Mean, Median, And Skew: Correcting A Textbook Rule. Journal Of Statistics Education, 13:965–971, 07 2005. Doi: 10.1080/10691898.2005.11910556

[18] W. Liao, B. Bak-Jensen, J. R. Pillai, Y. Wang, And Y. Wang. A Review Of Graph Neural Networks And Their Applications In Power Systems. Jan.2021.

[19] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, And M. Sun. Graph Neural Networks: A Review Of Methods And Applications. Ieee Transactions On Visualization And Computer Graphics, July 2019.