**Distribution System Bad Data Detection Using Graph Signal Processing**

**Abstract-** Effective data-driven methods for managing distribution systems are becoming more and more necessary as the use of sophisticated metering infrastructure increases. I read a paper on graph signal processing-based system for faulty data detection. It begins with a physics-based technique for creating three-phase power system graphs and then presents a data-driven methodology for clustering low-dimensional voltage signal transform representations in order to identify problems. I changed some parameters in it according to my understanding, that can be understood further.

**Introduction-** Large volumes of data on electricity distribution systems are produced by the large installation of smart meters. Abnormalities like malfunctioning meters or lost communications could threaten this data. Finding these kinds of mistakes is essential to trustworthy data analysis. This paper suggest a technique for locating inaccurate voltage measurements in distribution networks that is based on fourier transforms and clustering bases method. This method offers a fresh approach to anomaly detection by clustering voltage signals in the frequency domain.

**Related Work**- Diverse techniques have been put to identify irregularities in power systems, specifically in distribution networks. Some rely on redundant readings, but as advanced metering technology is increasingly deployed, these aren't always available. Other strategies make use of PCA-based techniques or neural networks, however they don't take into account the physical characteristics of the system or require clean training data. Even if some models seem promising, they might rely on single-phase approximations or precise mapping models. Although graph signal processing has not been thoroughly studied for imbalanced three-phase systems, it has been investigated for load disaggregation and attack detection.

**Method of the actual paper-**

**Graph Signal Processing- A** graph is defined by a set of N nodes or vertices V ∈ [v1, ..., vN ] connected by M edges E ∈ [e1, . . . , eM]. The adjacency matrix A of a graph= symmetric matrix with the ijth element aij = 1 (if eij exists, otherwise 0). In general, these connections may or may not be directed. That is, aij may or may not equal aji. An extension of the adjacency matrix is the weight matrix W, in which the wij may assume some value relating to the strength of the connection between nodes i and j.The degree matrix D is a diagonal matrix wherein the entries dii are defined as the sum of all weights related to node i: dii = PN j=1 eij . As the basic building block of graph signal processing, the graph Laplacian can then be defined as L = D − W. Graph signal processing extends traditional graph theory by defining a signal on a graph as a set of nodal measurements x = {x1, ..., xN }. The graph Laplacian may be decomposed by eigenvalue decomposition. The decomposition is obtained as: L = UΛU (1) where the columns of U are the eigenvectors of L and Λ is a diagonal matrix comprised of the eigenvalues, with the eigenvalues sorted in ascending order. The Laplacian has at least one zero eigenvalue. The magnitude of the eigenvalues corresponds to the frequency component of the signal, with eigenvectors corresponding to small eigenvalues being part of the low-frequency component. Using the eigenvalue decomposition of the Laplacian, the graph Fourier transform (GFT) X of signal x, also referred to as the spectral signal, is defined as X = U −1x (2). Arising from this transformation is a notion of ‘smoothness’, a measure of the overall variation across the graph. A signal is smooth if it varies slowly in the frequency domain, and thus has mostly low-frequency components.

For us, measurement points are nodes in the graph. The nodal signals are the voltage magnitude measurements. In power distribution systems, most buses are served by more than one phase wire, and accordingly there can be more than one voltage magnitude measurement at every bus. To address this issue, a distribution grid is split into multiple graphs, each corresponding to a certain phase connection. The manner in which the phase connection is defined in each graph depends on the phases across which the nodal voltage magnitude is measured. If all measurements are phase-to-neutral, then the grid may be separated into three graphs, each corresponding to a particular phasing φi1 −φi2 ∈ [A−N, B−N, C−N]. If all measurements are phase-to-phase or some combination of phase-to-phase and phase-to-neutral, then the grid may be separated into three graphs corresponding to phasings φi1 − φi2 ∈ [A − B, B − C, C − A]. For example, a certain bus voltage may be measured on phase C − N, but served by a distribution line which at some point only carries phases B and C, but not A. In this scenario, the bus will only appear in the graph of phase B − C. Note that we assume voltage magnitude measurements are available at each bus on each phasing served. With the nodes defined, it remains to define the weights between these nodes. Physical systems often present intuitive means of constructing the adjacency matrix and weight matrix. For example, the weight matrix can be constructed using the Euclidean distance between nodes or the similarity between nodal measurements. For power distribution systems, it is natural to leverage physical line parameters. The edge weights should model the similarities between voltage measurements. Nodes that are not physically connected, intuitively, should have an edge weight of 0. Nodes connected by lines with small voltage drops should have more similar voltage measurements than nodes connected by lines with large voltage drops.

1) Weight Matrix Definition: The weight matrix of a particular phasing of the distribution system graph is constructed as follows. If a line connecting node i and j exists, the weight between node i with phasing φi1−φi2 and node j with phasing φj1 − φj2 can be defined as: wi,j =    (vdropφi1 ) −1 , if φi2 or φj2 = N and φi1 = φj1 or φj2 (vdropφi2 ) −1 , if φj2 = N and φi2 = φj1 (vdropφi1 − vdropφi2 ) −1 , if φi2 = φj2 6= N (3) where the per-phase voltage drop with 1 per unit current flow on phase Φ ∈ [A, B, C] can be calculated as: vdropΦ = Z Φ · Ilinephase (4) where Z Φ = z ΦA z ΦB z ΦC denotes the row of the 3 by 3 line impedance matrix that corresponds to phase Φ. Ilinephase = 1∠120° 1∠0° 1∠ − 120° denotes the per unit balanced current vector.

Nodes in a graph can only carry the same phase. Line-to-line and line-to-neutral voltages are taken into consideration by the weight matrix when accounting for voltage loss. It differentiates between several node kinds according to their connection type and phase. Without taking into account the current amount, the per-phase voltage drop roughly corresponds to the voltage loss in a balanced system. It is uncommon for the weight matrix to contain complex values in graph signal processing, as real-valued matrices are more typical.

Accordingly, we define weight matrices corresponding to the real part Wr = real(W) and imaginary part Wi = imag(W).

**2. Spectral Clustering Method-** Nodes in a graph can only carry the same phase. Line-to-line and line-to-neutral voltages are taken into consideration by the weight matrix when accounting for voltage loss. It differentiates between several node kinds according to their connection type and phase. Without taking into account the current amount, the per-phase voltage drop roughly corresponds to the voltage loss in a balanced system. It is uncommon for the weight matrix to contain complex values in graph signal processing, as real-valued matrices are more typical.

Since changes in voltage measurements are caused by variations in power consumption, we have chosen clustering since these fluctuations should be similar at connected nodes. Normal signals will have identical spectral components if the weight matrix takes voltage correlations into account. Nevertheless, defective signals resulting from inaccurate voltage data will display distinct spectral components. By accurately accounting for the contributions of impedance to voltage drop, this method also permits the use of various weight matrices, taking into account both real and imaginary components. This provides a more robust approach to error identification.

1.Feature Extraction and Dimensionality Reduction: The number of nodes in each graph is equal to the number of spectral signal components. By utilizing both the real and imaginary components of the weight matrix, the hybrid signal's components are doubled. But not every part is equally helpful in identifying inaccurate voltage data. To distinguish erroneous data from normal data, feature extraction and dimensionality reduction are required. For nonlinear dimensionality reduction, T-distributed Stochastic Neighbor Embedding (t-SNE) is selected because it can see and represent comparable points as clusters in the embedded space. In order to effectively tune hyperparameters without labeled data, t-SNE minimizes the Kullback-Leibler divergence between high-dimensional and low-dimensional probability matrices. t-SNE should cluster normal measurements if the majority of the data are normal, and it should display erroneous measurements as smaller clusters or outliers.

2. Clustering- We apply a clustering approach on the low-dimensional embedding of the spectral signal to detect false positives. High-density zones are formed by normal measurements, whereas low-density areas separate erroneous measurements. Because density-based spatial clustering of applications with noise (DBSCAN) can handle clusters with unusual shapes, that is the method we have chosen. Two parameters are used by DBSCAN to identify clusters: ε, or maximum distance between points, and MinPts, or minimum number of points needed to form a cluster. By adding nearby locations that meet density restrictions, clusters grow. Small clusters and outliers are marked as errors based on the low likelihood of faulty data.

Experimental Setup-

1. Data Preparation-

A time-series dataset of real power consumptions from smart meters is obtained. 100 customers are randomly selected from this dataset to represent the load at each node in the test feeder. The aggregated loads are scaled to match the specified loads in the IEEE test feeder. Three-phase load flow analysis is performed to calculate the voltage at each node.

1. Adding noise-

Gaussian noise is added to the voltage measurements to simulate random measurement errors. Synthetic bad measurements are introduced by randomly adding or subtracting errors to the original measurements with a certain probability.

1. Test scenarios-

Two scenarios are created: one assuming smart meters measure line-to-neutral voltage, and the other assuming line-to-line voltage except for single-phase lines. A modification is made to the feeder's topology by removing switches that don't affect the system's behavior.

1. Dimensionality Reduction-

t-SNE (t-distributed Stochastic Neighbor Embedding) is used for dimensionality reduction. The perplexity parameter, which influences the distribution of points in the reduced space, is set to 50 after experimentation. The embedded spectral signals help identify clusters in the data.

1. Cluster Identification-

Values for parameters ε (epsilon) and MinPts for DBSCAN clustering are chosen based on the t-SNE plots. ε is set to 2.5 and MinPts to 3. Clusters with at least 25 points are considered significant.

Numeric Results-

In summary, the proposed data-driven bad data detection algorithm is evaluated using two metrics: F1 score and accuracy. The F1 score is the harmonic mean of precision and recall, where precision measures the proportion of correctly identified positive cases among all cases identified as positive, and recall measures the proportion of correctly identified positive cases among all actual positive cases.

Given that the number of normal measurements significantly outweighs the number of erroneous data points in the test scenarios, both F1 score and accuracy are reported. F1 score gives equal weight to false positives and false negatives, while accuracy measures the proportion of correctly classified data points.

Method that I applied-

1. Firstly, I chose the data of three phases of a single bus in a time series format. Then I convert it into fourier transform so that we can analysis the data in frequency domain. As we know during most of the times, our voltage data will be in limits that are pre-described and only some of the times it will vary from the limit. This this can be easily understood by fourier transform by seeing it as most of the data is at zero frequency of the fourier curve and if erroneous data is there in the table we will see fluctuation in the fourier transform graph as it proceeds further.

2. Then because we have a large amount of data that it is not possible to do analysis so this high dimensional data points can be converted into low dimensional so that easier analysis can be done.

Definition of t-SNE- t-SNE (t-distributed Stochastic Neighbor Embedding) is a dimensionality reduction technique used primarily for visualizing high-dimensional data in a lower-dimensional space, typically two or three dimensions. It aims to capture the underlying structure of the data by preserving local similarities between data points.

Here's how t-SNE works:

1. Similarity Calculation-

t-SNE first constructs a probability distribution that represents similarities between pairs of high-dimensional data points. It measures similarity using a Gaussian distribution centered around each data point, where closer points have higher probabilities of being similar.

1. Mapping to lower dimensionality-

Next, t-SNE defines a similar probability distribution in the lower-dimensional space (usually two or three dimensions).

It iteratively adjusts the positions of points in the lower-dimensional space so that the distributions in the high-dimensional and low-dimensional spaces are as similar as possible. This is done by minimizing the Kullback-Leibler divergence between the two distributions.

1. Gradient Descent Optimization-

t-SNE uses gradient descent optimization to minimize the mismatch between the distributions.

It updates the positions of the points in the lower-dimensional space based on the gradient of the cost function, which measures the mismatch between the distributions.

1. Preservation of local structure-

t-SNE tends to preserve the local structure of the data, meaning that nearby points in the high-dimensional space are typically represented by nearby points in the lower-dimensional space

3. Then clustering was done of the t-sne data to analysis the data with the help of clusters. Density-based spatial clustering of applications with noise (DBSCAN), is selected as the clustering algorithm.

DBSCAN- DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise. It's a popular clustering algorithm used in machine learning and data mining for identifying clusters of points in a dataset. Unlike traditional clustering algorithms like k-means, DBSCAN does not require the number of clusters to be specified in advance, making it particularly useful for datasets where the number of clusters is not known beforehand or where the clusters have varying shapes and densities.

Here's how DBSCAN works:

1. Density Bases Clusterring-

DBSCAN operates based on the idea of density reachability. It defines clusters as areas of high density separated by areas of low density.

t starts by randomly selecting a point from the dataset. Then, it identifies all neighboring points within a specified distance (epsilon) from this point.

If the number of neighboring points is greater than or equal to a specified threshold (MinPts), the point is considered a core point, indicating that it's in a dense region of the dataset.

If a core point is found, DBSCAN expands the cluster by recursively finding all reachable points from this core point, adding them to the cluster.

1. Border Points-

Points that are within epsilon distance of a core point but don't have enough neighbors to be considered core points themselves are called border points.

Border points are added to the cluster of their nearest core point.

1. Noise points-

Points that are not core points and not reachable from any core point are considered noise points or outliers.

1. Resulting Clusters-

The DBSCAN algorithm identifies clusters as groups of connected core points, along with their associated border points.

Noise points are not assigned to any cluster.

Epsilon- The maximum distance between two points for one to be considered as being in the neighborhood of the other.

MinPts- The minimum number of points (a threshold) within ε distance to define a core point.

4. Analysis-

Our clustering is resulted in two clusters: Cluster 0 and outliers labeled as Cluster -1.

Here's a breakdown of the information-

Cluster Counts-

Cluster -1 (outliers): 293 points

Cluster 0: 7 points

Cluster 0-

Number of points: 7

Mean values:

v1: 1.714286

v2: 364.141429

v3: 0.000000

Standard deviation:

v1: 4.535574

v2: 51.846167

v3: 0.000000