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

Jagdish Meghwal (231040054)  
*M.Tech(Power Engineering),IIT Kanpur*  
jagdishm23@gmail.com

*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 Ease of Use. THE WORK THAT WERE USED IN THE ORIGINAL PAPER IS ALSO MENTIONED.

# 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.

II. Method -

**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 Clustering-

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

It 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.

III. Analysis-

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

Here's a breakdown of the information-

**Cluster 0:**

Number of points: 4304

Mean values:

v1 237.642414

v2 237.615409

v3 237.577691

Standard deviation:

v1 6.519722

v2 5.726260

v3 5.513261

**Cluster 1:**

Number of points: 14

Mean values:

v1 2.500000

v2 442.714286

v3 3.428571

Standard deviation:

v1 1.160239

v2 39.322001

v3 1.554858

##### Reference

Osten Anderson, and Nanpeng Yu, “Distribution System Bad Data Detection Using Graph Signal Processing,” Department of Electrical and Computer Engineering University of California, Riverside Riverside, CA, 2018