Power Line Fault Detection Project Report

Astral Cai - 10177331, Adams Liu - 10187602, Jeff Peng - 10185016

Abstract—Partial discharges (PD) in power grids can cause damages to the electrical assets, as well as power outages or even fires. Therefore, it is crucial to detect partial discharges early so that they can be repaired. The goal of this project is to design a model that detects partial discharges in voltage readings. In this paper, we propose a manual feature extraction based model using characteristics of peaks in the signal. Finally, we obtained a prediction precision of 0.815 with the LightGBM model using the features we extracted.

Index Terms—Group 4, Data Analytics

1 Introduction

Partial discharges (PD) are localized dielectric breakdowns formed due to the degradation of insulators. The lack of insulation results in electrons travelling through unwanted mediums such as gas voids, or transformer oil. The PD phenomenon introduces instability into an electric system, potentially causing short circuits. The discharged electricity is especially dangerous if the electric potential is high. High-voltage environments such as power generating stations and transmission lines are extremely vulnerable to partial discharges. If left unattended, they can lead to power outages or even fires, posing a significant safety concern.

Furthermore, partial discharges can cause damages to electrical assets, costing hundreds of thousands of dollars to replace. However, with an increasing demand for electricity and the ageing electrical infrastructure, it is infeasible to inspect the transmission lines to look for these defects manually. As a result, it is imperative to detect partial discharges early on, as it can help power companies save millions of dollars in asset repairs and compensation fees for businesses affected by power outages.

In 2019, a data analytics competition named *VSB Power Line Fault Detection* was held on Kaggle to address this issue. The competition provided signals acquired from power lines with a new meter designed at the ENET Centre. As the problem of PD detection is somewhat open-ended, we chose to address this problem as framed in this competition. The goal of this project is to build an analytic model that detects PD patterns from voltage readings.

We used signal processing methods such as the discrete wavelet transform to de-noise the voltage readings and manually extracted several features from each sample. Then we tested different classifiers, including regression, support vector machines, and ensemble learning. We have also made unsuccessful attempts to use deep learning models such as convolutional neural networks and recurrent neural networks to solve this problem. Finally, we achieved a test accuracy of 0.839 with the LightGBM classifier on the features we extracted.

- *Members 1, 2, and 3 are with School of Computing at Queen's University: astral.cai@queensu.ca, adams.liu@queensu.ca, jeff.peng@queensu.ca*

2 Related Work

There have been numerous proposed methods for partial discharge recognition. Many of them involve distinguishing between different types of PD pulses or studying the relationship between the physical defects related to each PD signal. We discovered that the problem of partial discharge recognition is very complicated. The choice of model is highly dependent on the nature of the data, sometimes even the physical properties of the transmission lines, as well as the sensors used to collect the signals [1]. As a result, most papers we reviewed were not directly applicable to our task. However, they can still be used as a reference for the available tools.

In 2017, Mišák et al. [2] from the ENET Center proposed a simple method for online power line fault detection. The proposed solution in this paper applies to voltage readings collected using a single layer inductor designed by the researchers. In this paper, they applied univariate wavelet analysis and hard thresholding to denoise the data. Then, they selected a small section from each voltage cycle based on their understanding of the physical properties of the PD pulse. Finally, they extracted features related to the peaks in the signal, such as the heights and widths of peaks, and used a random forest classifier to generate predictions. They also used the self-organized migrating algorithm (SOMA) to optimize the performance of their classifier. They achieved an overall precision of 0.836 on a data set generated in the lab environment.

In 2018, Adam and Tenbohlen [3] proposed and compared two different models for PD classification. The main focus of their research was classifying different types of PD pulses as opposed to detecting the existence of a PD pulse in a continuous voltage signal. The first approach was to apply the Fourier transform to the raw signal and extract statistical features from the frequency space. They proposed to segment the frequency spectrum of the signal and extracted different features from each segment, which requires domain knowledge. Finally, they used the random forest classifier to achieve an overall classification accuracy of 0.9825. The second approach was to use the long short term memory (LSTM) feature extractor connected to a dense classifier. The input to the LSTM model is the original signal

down-sampled to 100 data points. This model achieved an overall accuracy of 0.9704.

Another popular choice of model for PD classification is the convolutional neural network (CNN). In 2018, Wan et al. [4] proposed a time-domain waveform pattern recognition method based on a one-dimensional convolutional neural network to solve the partial discharge detection problem. Image processing and pattern recognition techniques are applied to extract features from waveform images. The data were gathered from on-site sensor readings and simulation experiments. However, instead of raw voltage readings, the data were stored in the form of waveform images. The proposed model achieved average recognition accuracy of 88.9% across five different defect types, outperforming the SVM, the BPNN, and the 2D-CNN.

3 Dataset

The data set used in this project is obtained from the Kaggle competition mentioned before [5]. Each sample in the data set is a series of 800,000 voltage readings sampled over a period of 20 milliseconds. The power grid operates at a frequency of 50 Hz, which means each signal covers a full grid cycle. Each data point is stored as an 8-bit integer ranging from -127 to 128. Figure 1 shows an example of a signal in the given data set.

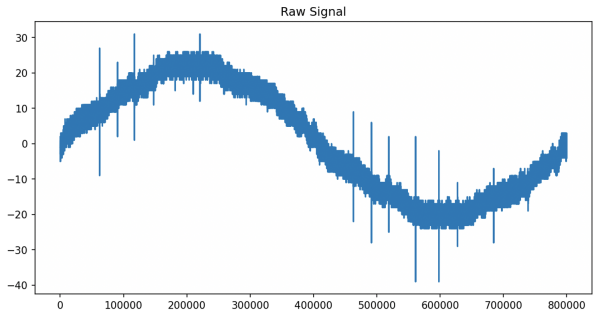

Fig. 1. An example of a signal in the data set.

This data set consists of 8712 signals in total, 525 of which contain partial discharge patterns. This is a highly imbalanced data set. Therefore, we decided to down-sample the data set by keeping all 525 positive samples, and 1050 randomly selected negative samples, which is twice the size of the positive samples. The 2:1 ratio, as opposed to the typical 1:1 ratio between negative and positive samples, was chosen to reflect that in real life, partial discharges occur less frequently. This gives us a total of 1575 samples.

4 Methodology

4.1 Data Reprocessing

After reading multiple research papers and Kaggle submissions, we found that there is a somewhat standard procedure for signal pre-processing. We used the discrete wavelet transform (DWT) to obtain the frequency spectrum of the signal. The DWT is a variation of the Fourier transform

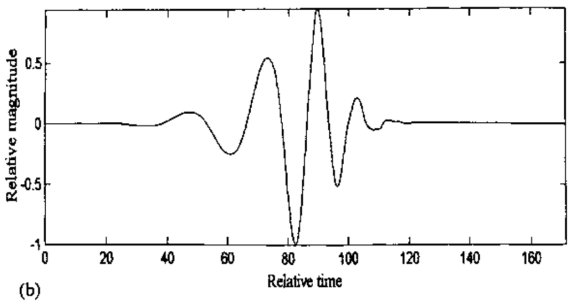

Fig. 2. An example of a wavelet used in discrete wavelet transform [1]

designed for discrete signal analysis. Wavelets are small wave packets localized in time [1], illustrated in Figure 2.

After obtaining the frequency-domain representation of the signal, we apply thresholding to remove noise in the signal. This involves inspecting the coefficients obtained using DWT, and removing the ones that does not exceed a certain threshold, which is determined using the following equations proposed by Tomas Vantuch [6]:

$$\sigma = 1/0.6745\mathrm{MAD}\left(|c_d|\right), \tag{1}$$

$$T_d = \sigma\sqrt{2\log(n)}, \tag{2}$$

where c_d is the coefficients to the frequency spectrum of the signal, MAD is the mean absolute deviation function, and T_d is the final threshold. Then, we reconstruct the signal using the filtered coefficients and apply a process called detrending, where we take the numerical derivative of the signal, which highlights the large variations in the voltage level, illustrated in Figure 3.

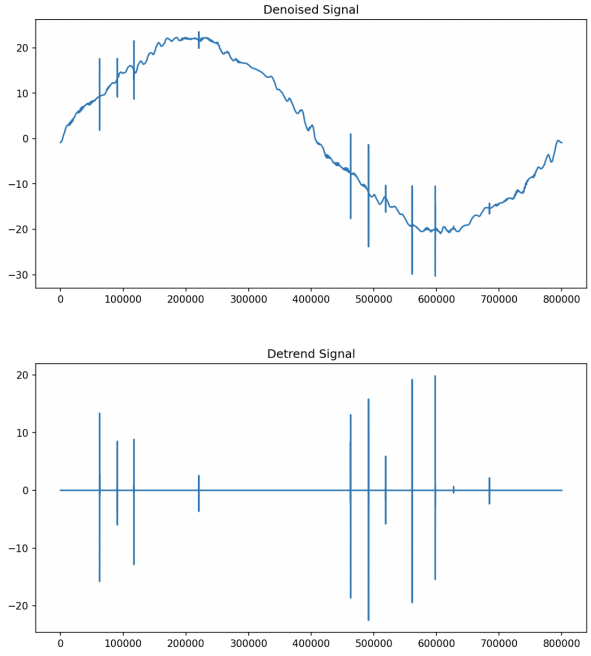

Fig. 3. The signal after thresholding, and the signal after detrending.

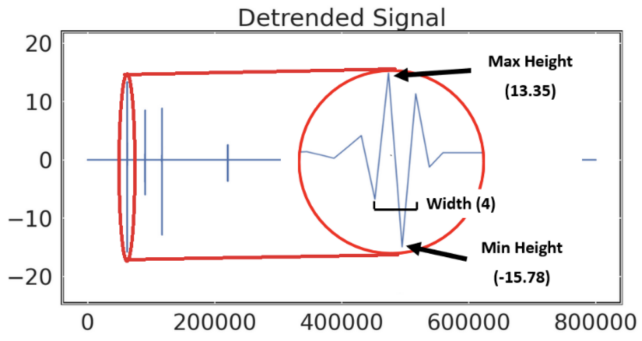

Fig. 4. An example of a spike and its characteristics in a detrended signal

4.2 Feature Extraction

The most interesting characteristics of the signals are obtained by evaluating the interruptions. In this case, we decided to focus on the spikes. There are two main types of spikes: peaks and spikes, one has positive amplitudes, and the other has negative amplitudes. By inspecting the detrended signals, we decided to set the threshold for what is considered a spike at 4. Figure 4 shows a zoomed-in snippet of the first spike in one of the detrended signals. We can extract some interesting features from the spike characteristics. The maximum and minimum amplitudes of a spike can be taken, as well as the widths.

In the case of Figure 4, the maximum height of the spike is 13.35, the minimum height is -15.78, and the width is 4. Note that even though the first zoomed-in segment of the signal contains 7 distinct interruptions, only 4 of them have magnitudes greater than the threshold of 4, thus making make the width of this spike 4. Other features are related to the statistics of the entire signal, such as the total number of peaks and valleys, as well as the mean, median, standard deviation, variance, and skewness. Note that each spike contains several peaks and valleys, as shown in Figure 4 This sums up to a total of 16 features, listed below:

- Number of True Spikes
- Number of Peaks
- Number of Total Peaks and Valleys
- Number of Valleys
- Maximum Spike Width
- Minimum Spike Width
- Maximum Spike Height
- Minimum Spike Height
- Mean Spike Width
- Mean Spike Height
- Variance
- Standard Deviation
- Height Difference
- Skewness
- Median
- Mean

4.3 Baseline and Feature Engineering

We decided to use the random forest classifier as a baseline, replicating the approach proposed by Mišák et al. [2]. We used 5-fold stratified cross-validation to evaluate the performance of the model and included all 16 features. We set the number of estimators to 100, applying the gini impurity metric, and setting the minimum sample split to 2. The final performance of the baseline model was a precision of 0.815, and an F1-score of 0.779.

The results show that the baseline model is already showing high performance, which is proof that the features we extracted have good predicting power. Using the baseline results, we were able to perform feature selection based on the importance score for each feature provided by the random forest classifier. We also performed multicollinearity analysis with the features. We calculated Pearson's correlation coefficients for all pairs of features and removed the ones that showed high correlations (> 0.95) with the others that had higher importance scores. We also removed unimportant features (importance score < 0.01). Finally, we selected the top 8 most relevant features:

- Number of True Spikes
- Mean Spike Width
- Max Spike Width
- Mean Spike Height
- Minimum Spike Height
- Variance
- Skewness
- Median

4.4 Classifiers

Four classifiers were chosen for the purpose of this study. These classifiers include random forest (RF), LightGBM (LGBM), support vector machine (SVM), and logistic regression (LR). The RF classifier was chosen as a benchmark to see if selecting specific features resulted in any performance improvement compared to the original baseline RF model. LightGBM was used due to its flexibility and speed to perform well in various machine learning problem domains. And finally, SVM and Logistic Regression was chosen for their popularity use in many types of binary classification problems.

All classifiers were trained using the default hyperparameters provided by the SciKitLearn toolkit, and the performance was evaluated for all PD signals using precision, recall, AUC, and F-measure as shown in Table 1.

	RF	LGBM	SVM	LR
Precision	0.811	0.790	0.741	0.759
Recall	0.727	0.735	0.577	0.473
AUC	0.811	0.807	0.727	0.690
F-measure	0.765	0.769	0.648	0.434

TABLE 1
Cross validation results for the random forest classifier (RF), LightGBM (LGBM), support vector machines (SVM), and logistic regression (LR)

The results indicates that the ensemble models scored much better than SVMs and LR. However, the ensemble

models still performed similarly to our initial baseline model indicating that selecting specific features did not pose any significance to improving results.

4.5 Hyperparameter Tuning

The next step was to do hyperparameter tuning using grid search. This was done for only the ensemble models. For RF, Table 2 shows the hyperparameters and the range of values that was used on for the grid search.

Parameters	Values
Number of trees	[100,200,500]
Split criterion	[gini, entropy]
Min samples for node split	[2,4,6,8,10]

TABLE 2
Hyperparameters used for random forest classifier

With the 30 candidates that was tested, the best candidate had an accuracy score of 0.834, with 200 decisions trees, using the entropy split criterion, and 10 minimum sample split. Table 3 shows the hyperparameters tuning metrics for the LightGBM classifier.

Parameters	Values
Number of trees	[100,200,500]
Max number of leaves	[31,50,100]
Minimum number of child samples	[2,4,8,10,20]
L1 Regularization Weight	[0,0.5,1]
L2 Regularization Weight	[0,0.5,1]

TABLE 3
Hyperparameters used for LightGBM classifier

For LightGBM 405 candidates were tested with the best candidate having an accuracy score of 0.839, using 100 trees, and using 31 as the max number of leaves,2 minimum child samples, and a L1 weight of 0.5. All in all, the accuracy of the results showed that both ensembles performed almost identically.

5 Results and Discussions

The final classification precision of the LightGBM model was 0.815, and the F1-score was 0.778. Although this project was based on a Kaggle competition, the submissions on the Kaggle website were evaluated with a separate data set, for which the labels were not released to the public. Since we do not have access to the same test data set used by the competition, we will not be comparing our results to submissions on Kaggle.

It is also worth noting that the metric we care about most for PD detection is the precision, which is the true positive rate of the model. A precision of 0.815 proves that the features we extracted do have prediction power. However, our model did not obtain state-of-the-art accuracy. On the other hand, since each researcher in this field uses a different data set, not much information can be obtained by comparing our model's performance to the existing results.

6 Future Work

Throughout this project, we have made several failed attempts at solving this problem, which are all potential fields to explore in the future. The first attempt was to perform multi-level discrete wavelet transform (DWT) as suggested by Zhou et al. [7] and remove the frequency ranges that are irrelevant to PD pulses. According to the paper, partial discharges reside in the hundred-kilohertz frequency range. Theoretically, by isolating that frequency range and reconstructing the signal solely from these frequencies, we should be able to obtain a much cleaner signal that still preserves information related to the PD pulse. Unfortunately, as shown in Figure 5, the result was very noisy, and did not seem to contain any useful information.

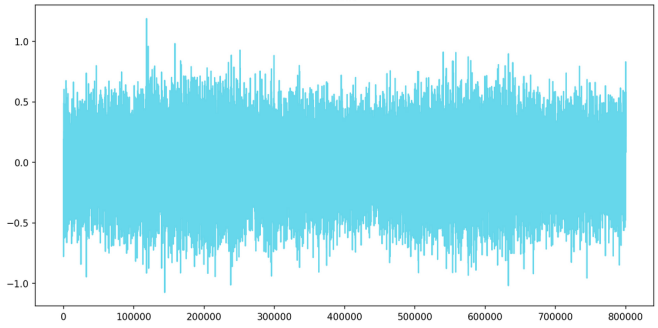

Fig. 5. An example of a PD signal reconstructed from the hundred-kilohertz frequency range of the signal's spectrum after DWT.

Another attempt was to use a convolutional neural network to replace manual feature extraction. However, as mentioned before, each sample in the data set contains 800,000 measurements, which is too large for any network to handle. As a result, down-sampling is needed. However, we lack the domain knowledge to understand if down-sampling will remove any useful information related to PD pulses. Furthermore, it is difficult to design an optimal network structure (i.e. the number of layers, the number and size of the convolutional filters at each layer). The convolutional neural network we tested did not converge, and we were not able to debug the network due to time constraints. However, research indicates that deep neural networks has shown high performance in PD recognition. Therefore, this is an area worth pursuing in the future.

7 Conclusion

The purpose of this project was to design a data analytics model that detects the existence of partial discharge patterns from voltage readings. The Kaggle *VSB Power Line Fault Detection* data set was used. After referencing research conducted on PD classification and existing Kaggle submissions, we designed a simple pipeline. We used the discrete wavelet transform (DWT) and thresholding to remove noise in the signal, then we used a process called de-trending to flatten the waveform as well as amplify the interruptions in the signal. Then we extracted features using simple statistics, as well as from characteristics of the spikes in the signal. We selected the most rel event features and tested multiple

classifiers using 5-fold cross validation. Finally, we achieved a test F1-score of 0.769 with the LightGBM classifier on the features we extracted.

8 Group Member Contributions

Group member contributions to the project:

- All group members participated in research.
- Adams Liu and Jeff Peng were looking into data pre-processing including denoising and de-trending
- Astral Cai was exploring discrete wavelet transform in more detail to facilitate data pre-processing
- Adams Liu worked on manual feature extraction
- Astral Cai and Adams Liu worked on testing different classifiers with the manually extracted features.
- Astral Cai was exploring the possibility of applying deep learning to this task.

Group member contributions to the report:

- Jeff Peng: Introduction & Dataset & Related Work
- Adams Liu: Related Work & Feature Extraction & Classifiers & Hyperparameter Tuning & Results
- Astral Cai: Related Work & Data pre-processing & Discussions & Overall editing of the entire report

9 Replication Package

https://github.com/adams-liu/VSB-Power-Line-Fault-Detection-Project/blob/master/351_Project.ipynb

References

[1] X. Ma, C. Zhou, and I. J. Kemp. Interpretation of wavelet analysis and its application in partial discharge detection. *IEEE Transactions on Dielectrics and Electrical Insulation*, 9(3):446–457, 2002.

[2] S Misák, J Fulnecek, Tomáš Vantuch, Tomáš Buriánek, and T Jezowicz. A complex classification approach of partial discharges from covered conductors in real environment. *IEEE Transactions on Dielectrics and Electrical Insulation*, 24(2):1097–1104, 2017.

[3] B. Adam and S. Tenbohlen. Classification of multiple pd sources by signal features and lstm networks. In *2018 IEEE International Conference on High Voltage Engineering and Application (ICHVE)*, pages 1–4, 2018.

[4] Xiaoqi Wan, Hui Song, Lingen Luo, Zhe Li, Gehao Sheng, and Xiuchen Jiang. Pattern recognition of partial discharge image based on one-dimensional convolutional neural network. In *2018 Condition Monitoring and Diagnosis (CMD)*, pages 1–4. IEEE, 2018.

[5] Kaggle vsb power line fault detection competition dataset. https://www.kaggle.com/c/vsb-power-line-fault-detection/data, 2019. [Online; accessed February 2020].

[6] Tomáš Vantuch. Analysis of time series data. 2018.

[7] Xiaohong Zhou, Chengke Zhou, and IJ Kemp. An improved methodology for application of wavelet transform to partial discharge measurement denoising. *IEEE transactions on dielectrics and electrical insulation*, 12(3):586–594, 2005.