Multiple classification methods of power quality events in three phase power data.

Master Candidate: Mariam Ali

University Supervisor: Prof. Marco Zanetti
Internship Supervisor: Dr. Ebrahim Balouji

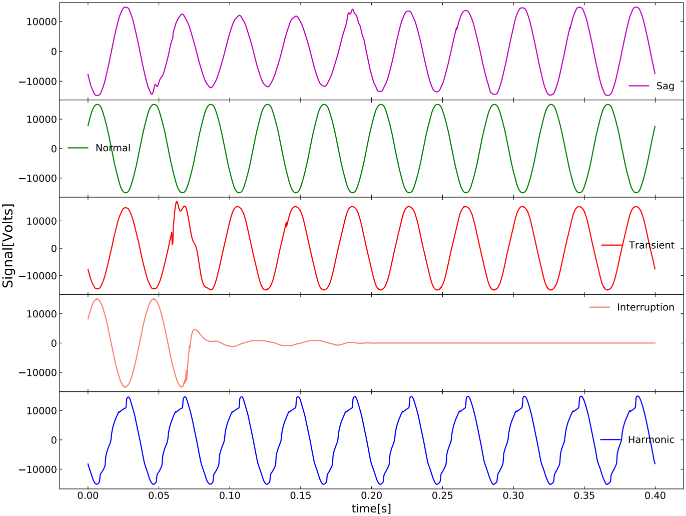
Università degli Studi di Padova

12th April 2021

Outline

Introduction: Power Quality Events

- Sag
- Swell
- Transient
- Interruption
- Harmonic



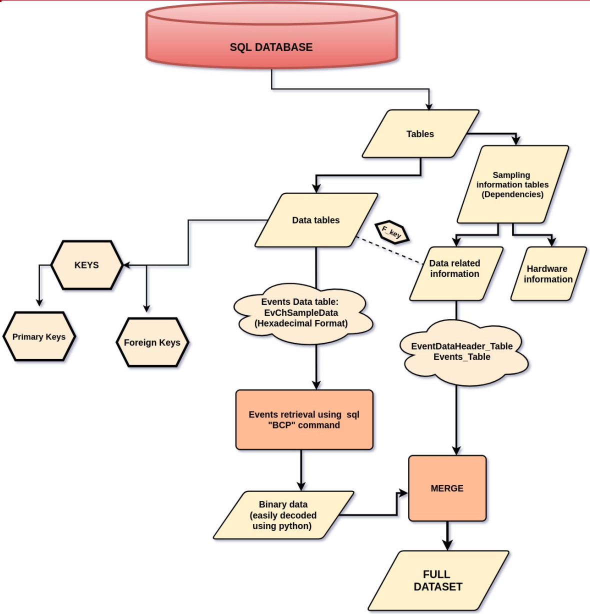
Project Aim

Create a model to classify different power quality events of three phase voltage data.

Project Structure:

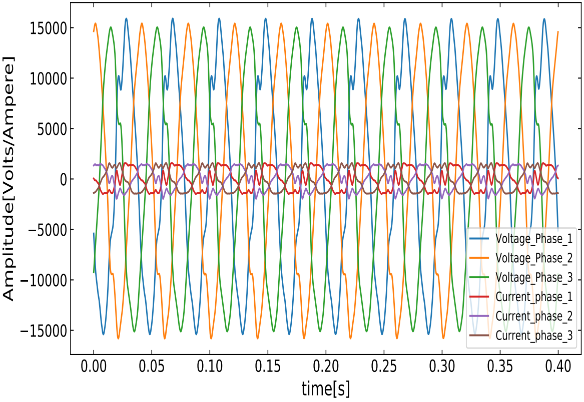
- Understand the structure of an SQL database provided by Eneryeild; collected from electrical power grids in Sweden.
- Retrieve the correct type of data.
- Integrate multiple SQL tables to have complete data set.
- Label the data using multiple techniques to try to solve the ambiguity problem of the data labels.
- Choose and build a suitable classifier to classify the data based on different types of labels.

Methods: Database Structure

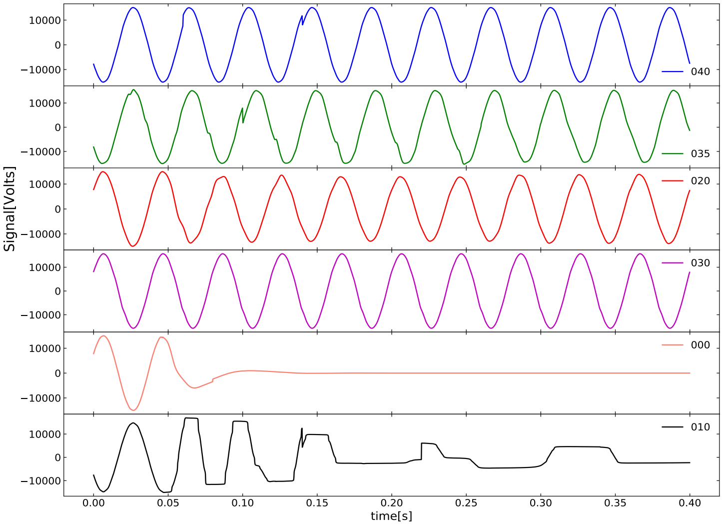


Methods: Full Data set

- 58,000 events of wave-forms and RMS.
- The scope of this project is concerned with wave-forms.
- Each event contains 8 channel; 3-phase Voltage, 3-phase current, Earth voltage & Earth current
- We use $16,000$ data points of the 3-phase voltage wave-forms.

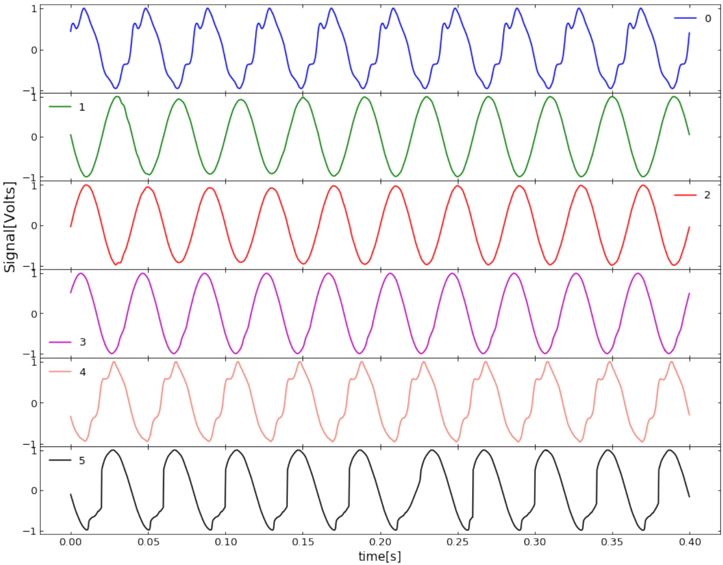


Data Labels

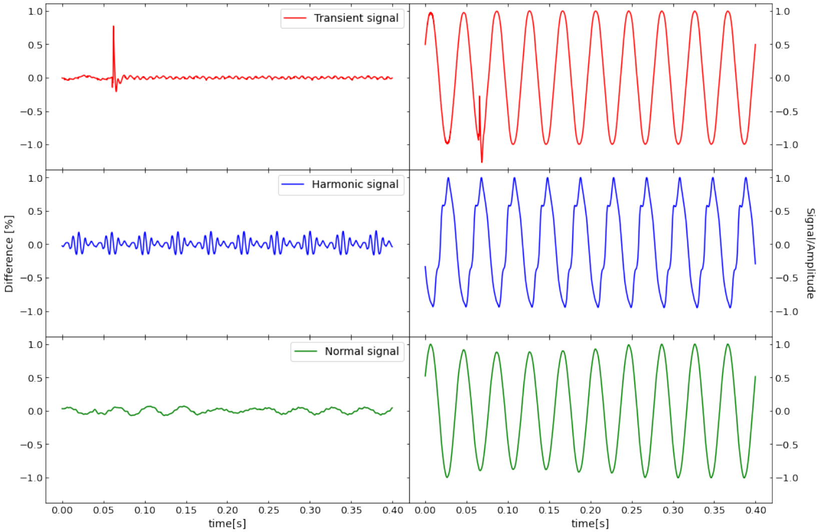


Signal[Volts]
time[s]
040
035
020
030
000
010
10000
0
−10000
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40

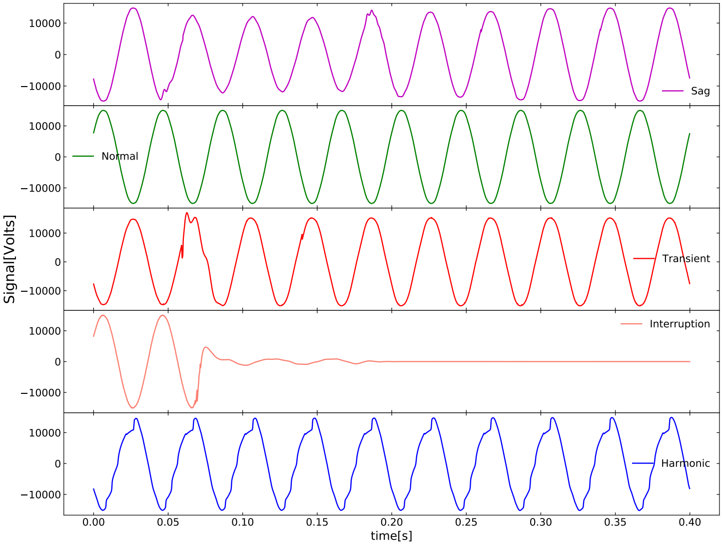
Methods: K-means



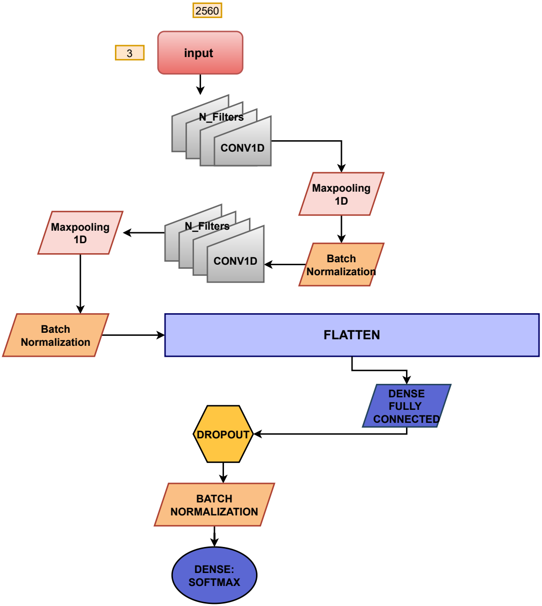
Methods: Thresholds labeling



Methods:Thresholds Labels



Methods: CNN



2560
3
input
N_Filters
CONV1D
Maxpooling 1D
Batch Normalization
N_Filters
CONV1D
Maxpooling 1D
Batch Normalization
FLATTEN
DENSE FULLY CONNECTED
DROPOUT
BATCH NORMALIZATION
DENSE: SOFTMAX

Results: Data labels

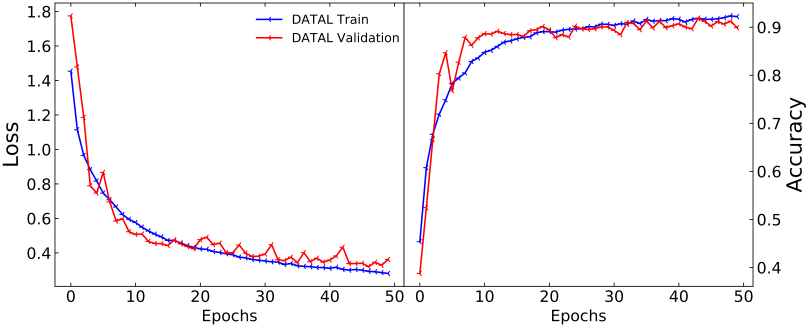


Figure: loss and accuracy of the CNN model learning from the data labels for a duration of 50 epochs.

Results: K-means labels

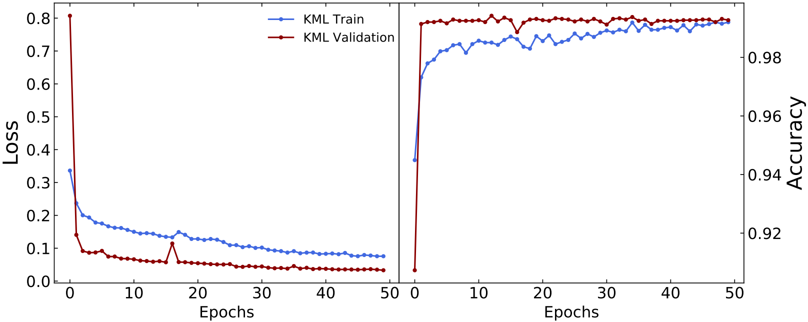


Figure: loss and accuracy of the CNN model learning from the K-means labels for a duration of 50 epochs.

Results:Threshold labels

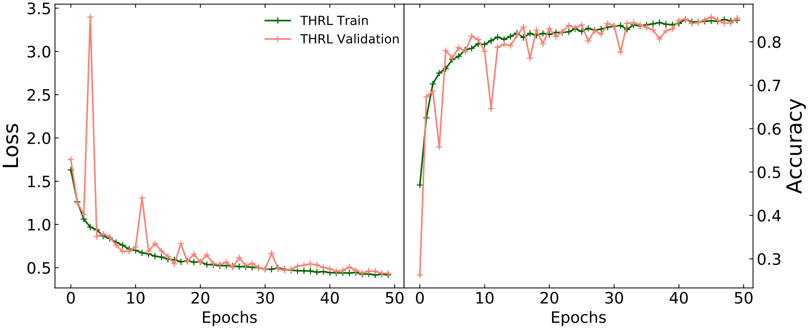


Figure: loss and accuracy of the CNN model learning from the Threshold labels for a duration of 50 epochs.

Results: Random Search

Table: The tuned hyper parameters resulting from the random search based on the two types of labels.

Label	Threshold	Data
num of filters	32	32
units	64	96
Activation	relu	tanh
dropout	0.35	0.0
optimizer	nadam	nadam
learning rate	0.000020	0.00045
Accuracy	90%	95%
loss	0.36	0.1

Results: Random Search

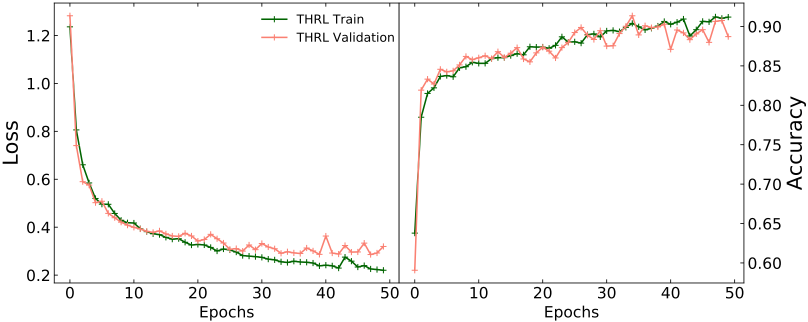
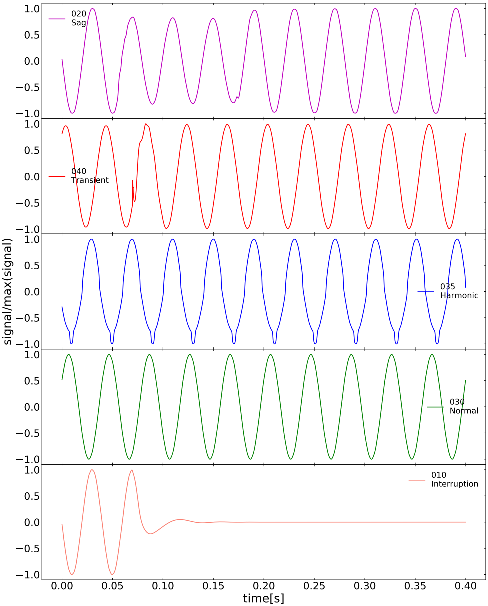


Figure: loss and accuracy of the CNN model with the tuned hyper parameters showing a decrease in loss fluctuations.

Results: Best Model Predictions



Conclusion

- We find that the CNN is a suitable network to fit our data as it accommodates the dimensions of the data without any under or over fitting.
- The CNN we used is useful for our purposes as it reaches at least 83% accuracy in all type of labels and 90% after tuning the parameters of the model.
- Despite the high accuracy reached for the K-means, it is not a suitable labeling algorithm for our purposes.
- The threshold labeling algorithm produces results that have remarkably high accuracy while at the same time it is the most understandable labeling technique.

Future Work:

- Improve the performance of the threshold labeling algorithm by implementing a more sensitive threshold labeling method to be able to differentiate strongly between "Transients", "Harmonics" and normal signals.
- Use a more advanced unsupervised deep learning method to label the data.
- Extend the current network to larger datasets by renaming the data labels according to the predictions agreement.

Thank You!

APPENDIX:Auto-correlation

Auto-correlation is a widely known mathematical tool to identify repeating patterns in signals. Given measurements $Y_1, Y_2, ..., Y_N$ at times $t_1, t_2, ..., t_N$, with lag k, auto-correlation function is defined as:

$$r_k = \frac{\sum_{i=1}^{N-k}(Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^{N}(Y_i - \bar{Y})^2} \ . \tag{1}$$

The time variable,t, is not used in the formula for auto-correlation under the assumption is that the observations are equi-spaced (that the siganl is sampled with a constant sampling rate) .This is true in our the case .

Cross Entropy

$$E_{CE} = \sum_{i=1}^{n} t_i log(p_i), \tag{2}$$

SGD:

$$\nabla_\theta E(\theta) = \sum_{i=1}^{n} \nabla_\theta e_i(x_i, \theta) \Rightarrow \sum_{i \in B_k} \nabla_\theta e_i(x_i, \theta), \tag{3}$$

cycling over all mini batches M

$$\nabla_\theta E^{MB}(\theta) = \sum_{i \in B_k} \nabla_\theta e_i(x_i, \theta), \tag{4}$$

Finally the parameter update equation would be similar to that of the GD:

$$\begin{aligned} v_t &= \eta_t v \nabla_\theta E^{MB}(\theta_t), \\ \theta_{t+1} &= \theta_t - v_t. \end{aligned} \tag{5}$$

APPENDIX: NAG+ADAM

Nestrov Accelerated Gradient (NAG) "Nesterov acceleration optimization is like a ball rolling down the hill but knows exactly when to slow down before the gradient of the hill increases again."

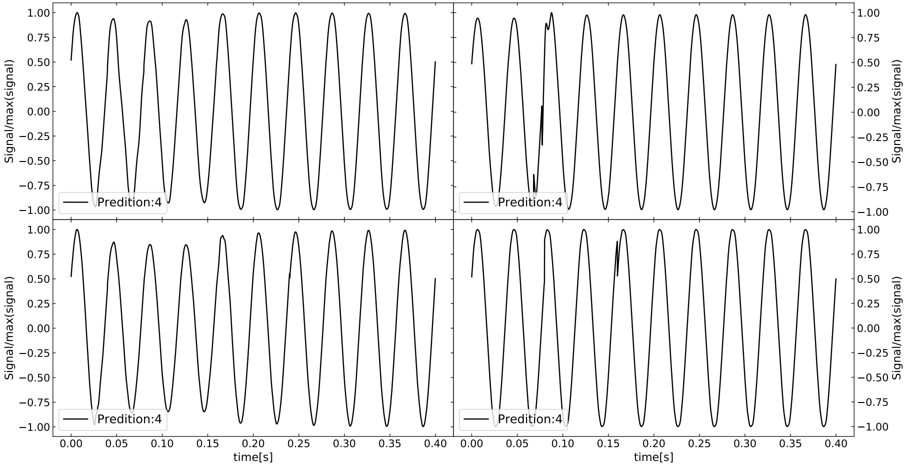
$$\theta = \theta - v_t \tag{6}$$

$$v_t = \gamma v_{t-1} + \eta \nabla J(\theta - \gamma v_{t-1}) \tag{7}$$

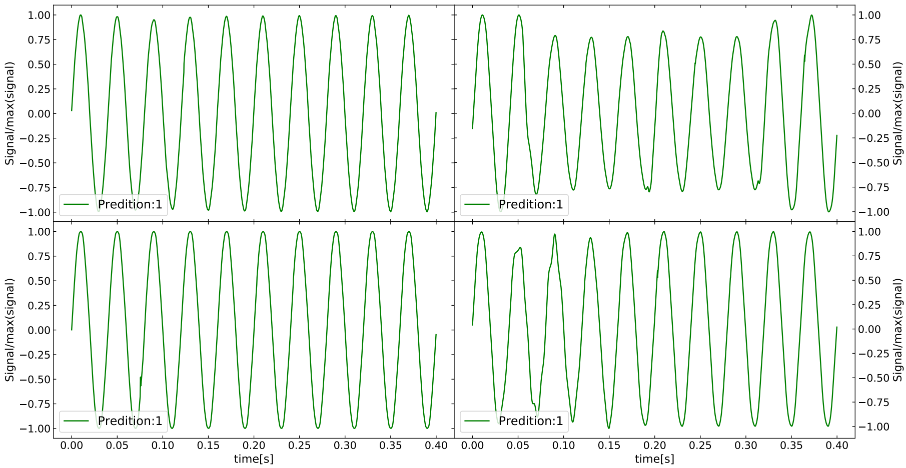
NAG is like going down the hill where we can look ahead in the future; $\theta - \gamma v_{t-1}$. is the looked ahead gradient. This way we can optimize our descent faster. NAG Works slightly better than standard Momentum.

Adam algorithm first updates the exponential moving averages of the gradient.

APPENDIX: K-means Labels Predictions



APPENDIX: K-means Labels Predictions



K-means Labels Predictions; Interruptions

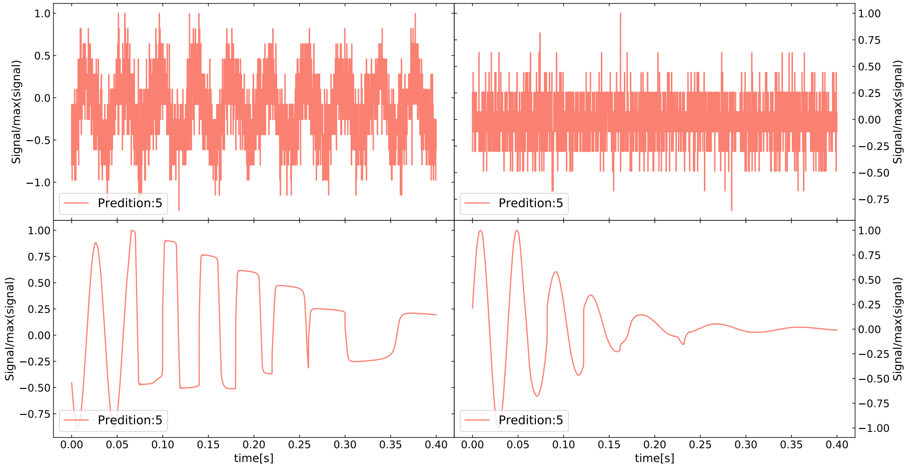


Figure: The last cluster of the k-means algorithm, the algorithm succeeds at clustering interruptions quite well. This is due to the unique spatial distribution of the interruption PQ event.