Supervised Machine Learning in Digital Power Line Communications

Kushal Thapa*, Stan McClellan†, Damian Valles§
Ingram School of Engineering
Texas State University
San Marcos, TX, USA
email: *k_t260@txstate.edu, †stan.mcclellan@txstate.edu, §dvalles@txstate.edu

***Abstract*—Power Line Communications (PLC) is a technology that uses power lines to transport communication data alongside the AC electric signals. Due to the highly penetrative pre-existing power grid infrastructure, PLC has a huge networking potential, especially in the implementation of smart grid technologies. However, PLC medium poses a major hindrance in the form of poor signal propagation. Traditional signal processing measures are not enough to demodulate these poor signals at the receiver end. To overcome this challenge, we are investigating Machine Learning (ML) as a supplement to the traditional digital signal processing techniques in this project. Our project focuses on testing and comparing various supervised machine learning and deep learning algorithms for the purpose of digital PLC bit classification.**

Keywords-power line communications; PLC; machine learning; ML; smart grid.

I. INTRODUCTION

The use of electrical wiring and power lines for network communication is not new. Since the early 1920s, this technology has been used to automate meter reading by utility companies [1]. Beyond this application, the potential of Power Line Communications (PLC) was conceptualized as a universal networking solution mainly because of the pre-existing power-grid [2]. This power grid would obviate the need for building other types of dedicated communication infrastructures like phone lines and optical fibers, thereby saving billions in cost [2]. However, over the years, such high expectations of this technology have not been realized due to many factors. One of the primary culprits is signal propagation.

The power grid infrastructures, including the power cables, were not designed for communication purposes. Thus, communication signals face various hindrances in this medium, including highly variant and dynamic noise, radiation leakage, undesired modulation, etc., [3]. All of these problems aggregate to cause poor propagation of the signal. One approach to solving this problem is to devise ways to cancel out these causes and maintain a better quality of signal throughout its communication path. A different approach would be to design a better, more sensitive receiver that could extract information even from the poorly propagated communication signals. The latter approach has an advantage because only the receiver needs modification, while the former might need engineering improvements in the transmitter and the medium.

Traditional communication receivers work primarily by implementing Digital Signal Processing (DSP) techniques, such as demodulation, filtering, digitization, etc., [4]. However, these methods alone are not sensitive enough to extract the information signals in PLC. Machine Learning (ML), which is a technique that probes data for information, might be a good supplement to traditional signal processing in creating more sensitive receivers for PLC. Therefore, in our study, we have designed a PLC network architecture and used ML with signal processing features to extract the transmitted information from the raw PLC signal captured at the receiver.

The signal workflow of our project is shown in Figure 1. Digital information is modulated onto an analog carrier at the transmitter in one of several well-known approaches. This analog signal is injected into the power line where it combines with the dominant power signal plus highly variant noise. The output is collected with a Data Acquisition Device (DAQ) at the receiver, and it consists of a raw signal that resembles a power signal. Various features are extracted from this raw signal using DSP. These features, along with the corresponding digital labels, are arranged into a dataset. This dataset is then fed into ML algorithms, which creates a model. Lastly, we use this ML model to classify and thus, extract the transmitted digital information.

The rest of the paper is organized as follows. In Section 2, we provide a description of the data-capture methodology, feature extraction process used in the raw-data, and the setup of the ML models. In Section 3, we present the outcomes of ML model optimization, performance of these models, and validation of the results. Finally, we summarize the paper and provide conclusions in Section 4.

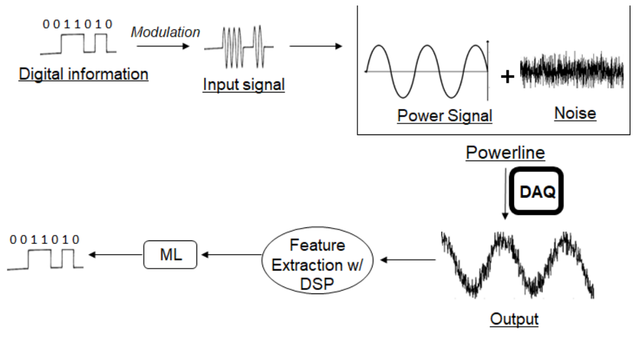

Figure 1. The flow of signals from digital input to ML output.

II. PROCEDURE

A. Data Capture

The experimental setup used to capture the PLC data is shown in Figure 2. As shown in the figure, we used a current source to inject a low amplitude signal with a frequency of 1595Hz into the power line via a current-modulator and a stabilizing filter. This signal first passes into a single-phase lab wiring, then into a three-phase distribution power grid via transformers. Some signature of the signal gets ingrained on all three-phases during this transition [3]. In the substation, the power signal and the injected signal go through more transformation, primarily due to Current Transformers (CT). These transformed signals were then collected at the substation using DAQ. The communication signal originated at the low voltage region (the lab) and traveled towards the high voltage region (substation) of the distribution grid, thereby making the PLC path upstream.

B. Raw Data

The raw data, captured using DAQ, was a three-phase time-series data consisting of a power signal at around 60Hz, communication signal at around 1595Hz, and time-variant noise at all frequencies. The power signal dominated the time-domain plot of this raw data because of its relatively high amplitude. Thus, the time-domain plot did not show any trace of our communication signal. The power signal and its harmonics also dominated the frequency-domain spectrum plot. However, a small peak was present at 1595Hz that showed the presence of our transmitted signal. However, the spectrum plot cannot show the time-varying nature of the signal, and thus, did not provide us information about the digital data that was transmitted. A spectrogram, which is a plot of signal energies in a time vs. frequency graph, helps acquire this information. Figure 3 shows the spectrogram of the Phase A raw data. We can see the dominant power signal and its harmonics at low frequencies. More significantly, there is a clear dotted band above 1500Hz, which is our communication signal. In this frequency band, the short bright dashes represent the 1s, and the gap between these dashes represents the 0s. These discrete amplitude (energy) shifts correspond to the data (bits) modulated and transmitted by the current source.

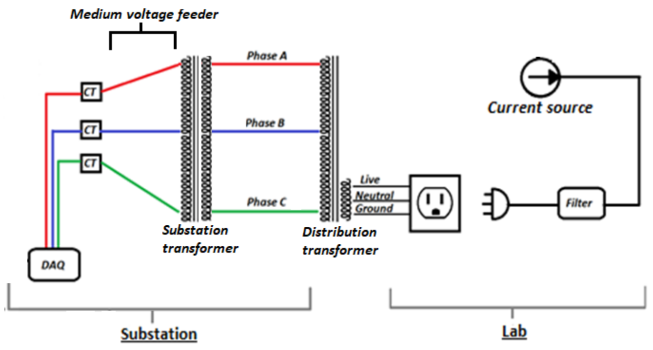

Figure 2. Experimental setup for sending and receiving a current signal through power lines in a distribution power grid.

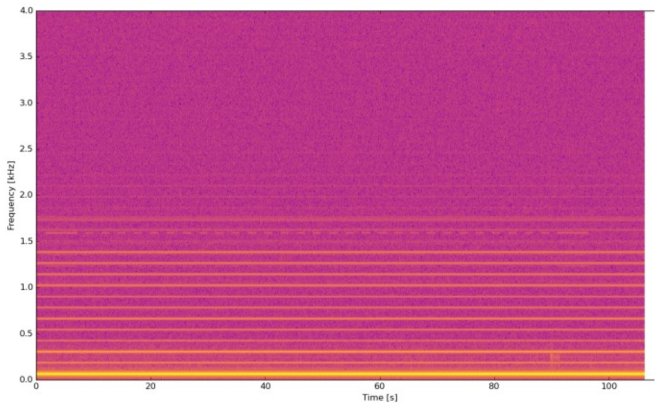

Figure 3. Spectrogram of Phase A of captured raw data.

C. Feature Extraction

After the raw data was collected, this data needed to be converted to ML-ready format. First, we divided the raw data into numerous frames corresponding to the resolution frame of the labels (sample length of a single bit in the transmitted analog signal). Each of these frames would be a sample row in our final dataset. Then, from each of these frames, we extracted various features as described below:

1) Amplitude Envelope

This feature gives the change in the amplitude of the signal over time [5]. It effectively traces the outline of the signal in the time-domain. In our case, the raw signal's amplitude envelope, as is, would not provide any meaningful information as the 60Hz power signal dominates all other superimposed sinusoidal signals. Therefore, we filtered the raw frames with band pass filters of 100Hz bandwidth starting from 1Hz and up to 2000Hz with no overlap (1Hz-100Hz, 101Hz-200Hz,...,1901-2000Hz). Hence, we divided each frame into twenty frequency-separated signals and calculated each of these signals' amplitude envelope. Our expectation was that the amplitude envelope of one of these signals which contains our communication frequency (1595Hz in our case) would provide information about the bit that was transmitted in that frame.

2) RMS (Root Mean Square) Energy

The energy of a signal is the measure of the "strength" of the signal. A signal's energy is defined as the sum of the square of its magnitude [6]. Thus, RMS Energy (RMSE) is the square root of the mean energy of a signal. Equation (1) [7] shows the formula for RMSE where x_i is the ith sample of signal x and N is the total number of samples.

$$RMSE = \sqrt{\frac{(x_1^2 + x_2^2 + \cdots + x_N^2)}{N}} \tag{1}$$

In our case, the raw signal's energy (or each frame) would again be dominated by the power signal. Hence, we frequency separated the frames as before and calculated RMSE for each of the twenty bandpass filtered signals of each frame. Like the amplitude envelope, we were expecting variations in the RMSE of 1,501-1,600Hz signals of different frames corresponding to the bit these frames were carrying.

3) Spectral Centroid

Amplitude envelope and RMSE are time-domain features, and thus, they were extracted from the time-series data. We decided to use spectral centroid to probe the frequency-domain of the raw data for important signal characteristics. The spectral centroid compares the center of mass of the signal's spectrum [8]. Our raw signal's spectrum had a primary peak at around 60Hz and secondary harmonic peaks at multiples of 60Hz because of the dominant power signal. Whenever the communication signal was present in the raw signal, there should also be a peak at 1595Hz (our communication frequency). We assumed that the presence and absence of the communication signal (corresponding to 1 and 0, respectively) would noticeably shift the center of the spectrum's mass, thereby providing a classification measure of the transmitted bit. Therefore, we included the spectral centroid of each frame as one of the features.

D. Machine Learning

After the dataset was formed by compiling the features from the raw data and labels were recorded, it was used in machine learning models with a 70% training split. To form the models with various supervised algorithms, Python Sci-kit learn used for Logistic Regression (LR) [9], Support Vector Machines (SVM) [10], and Decision Tree (TREE) [11]. The hyperparameters for these algorithms were optimized using the grid search [12] method. A majority voting model [13] was also created from the optimized LR, SVM, and TREE to check if such ensemble model would outperform the individual models. ROC AUC scores [14], precision [15], recall [16], and f1 scores [17] were computed to evaluate and compare these various models, training, and testing accuracy scores. Learning curves [18] were plotted and evaluated to ensure the models were not overfitting or underfitting. Confusion matrices [19] were also plotted to visualize the accurate label versus the predicted label.

Besides these basic "one neuron" ML models, multi-neuron, multilayer Artificial Neural Network /Deep Neural Network (ANN/DNN) model [20] was also tested using python's Tensor Flow and Keras. The various hyperparameters of these ANN/DNN models were optimized by manual trial and error method. Accuracy scores, loss and validation curves, and confusion matrix were generated to evaluate this ANN/DNN model's performance and this performance was compared with the other ML models.

ML was performed on the full dataset (with combined phase A, B, and C data). However, the accuracy and other performance metrics were low for this full dataset. Hence, the same ML techniques were applied for the phase A data only as well. The comparisons on the various metrics between these two datasets and other significant results are presented in Section 3.

III. Results and Discussions

A. Grid Search

A grid search was performed on the LR, SVM, and TREE algorithms to optimize the models' hyperparameters. Tables I and II show the optimized parameters and their corresponding values for each of these algorithms. These tables also show the training and testing accuracy values for respective algorithms. Table I is for the phase A data only, while Table II is for the combined phase A, B, and C data (full dataset).

As shown in Table I, all three algorithms, after grid search optimization, had similar performance in terms of training and testing accuracy for phase A data. The accuracy values were in the mid ninety percent, which indicates that the ML was successful in learning and classifying the samples into binary digital bits.

On the other hand, Table II shows that the ML models were not as relatively successful in the same regard for the full dataset. This might be because the phase B and C dataset did not have the same amount of information on the communication signal as phase A, or the features that we extracted did not work as well for phase B and C data. As shown in Figure 2, the communication signal is injected into a single phase, and the image of this signal gets ingrained into the other two phases when the signal transitions through a distribution transformer. From our accuracy result, we can infer that the signal was injected directly into phase A, and the images were produced in phase B and C later in the PLC path.

B. Feature Selection

Next, to examine the most impactful features and to plot a 2D graph with decision regions for each model, we used the Sequential Backward Selection (SBS) [21] method to filter out the two most essential features from a total of 41 (20 each of amplitude envelope and RMSE plus one spectral centroid). The results are presented in Tables III and IV.

As shown in these tables, one of the two best features for every algorithm in both datasets was 'RMSE 1501-1600'. This is the RMS energy feature of the samples after being filtered with a 1501Hz-1600Hz band pass filter. This frequency range is significant because our input communication signal is at 1595Hz. This result shows that the ML models can correctly identify the frequency band location of our communication signal. Further, the tables also show that RMSE was consistently the best feature in all cases. This is expected because the main difference between the 1s and 0s in our input signal is the signal strength, and RMSE is the measure of this signal strength.

The tables also show the accuracy values of the models with just the two best features. Comparing these values to the values in Tables I and II, we can see that reducing the dataset features from 41 to 2 did not have a significant impact on the accuracy of the models.

A 2D plot of labels with decision regions was produced using the two best features for each model. Figure 4 shows such a 2D plot of the Phase A training and testing set with SVM decision regions.

TABLE I. GRID SEARCH RESULTS FOR PHASE A DATA

Classifiers	Optimized parameters	Training accuracy	Testing accuracy
Logistic Regression	C=1.0, solver=lbfgs	94.06	93.99
SVM	C=1000, gamma=0.001	94.45	95.19
Decision Tree	Max_depth=1, Min_samples_split=1.0	94.19	95.19

TABLE II. GRID SEARCH RESULTS FOR FULL DATASET

Classifiers	Optimized parameters	Training accuracy	Testing accuracy
Logistic Regression	C=1.0, solver=lbfgs	77.27	76.73
SVM	C=10, gamma=0.1	77.15	75.52
Decision Tree	Max_depth=5, Min_samples_split=7	73.84	72.22

TABLE III. FEATURE SELECTION RESULTS FOR PHASE A DATA

Classifiers	Two best features	Training accuracy	Testing accuracy
Logistic Regression	RMSE 201-300 and RMSE 1501-1600	93.29	94.58
SVM	RMSE 1301-1400 and RMSE 1501-1600	95.53	96.78
Decision Tree	RMSE 1-100 and RMSE 1501-1600	95.7	95.19

TABLE IV. FEATURE SELECTION RESULTS FOR FULL DATASET

Classifiers	Two best features	Training accuracy	Testing accuracy
Logistic Regression	RMSE 501-600 and RMSE 1501-1600	73.43	72.21
SVM	RMSE 701-800 and RMSE 1501-1600	76.6	74.92
Decision Tree	RMSE 501-600 and RMSE 1501-1600	78.86	74.51

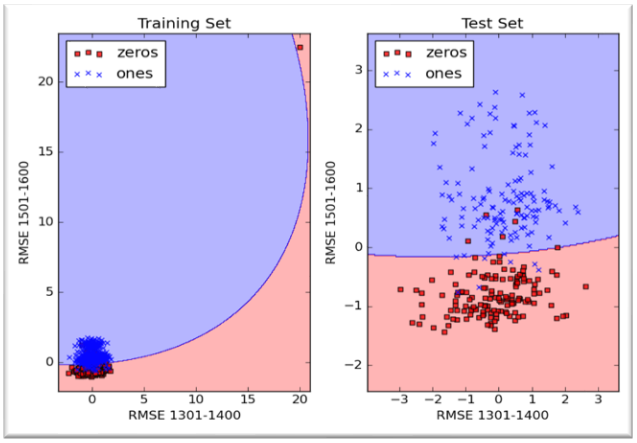

Figure 4. 2D feature plot showing labels and decision boundary of the SVM model for phase A data.

C. Learning Curve

To check if the ML models were overfitting or underfitting, we produced learning curves for each model. Overfitting is caused by high variance when models train with the noise and the appropriate data and produce a disproportionate result in the training and testing set [20]. In learning curves, overfitting can be implied if the training and validation accuracy curves do not converge and are far apart. On the other hand, underfitting is caused by high bias when the models do not consider all relevant data with appropriate weight. Underfitting can be implied in learning curves if the training and validation accuracy is consistently low [20].

Figure 5 shows the learning curve of the LR model for the phase A dataset. This shows that the model was not overfitted or underfitted. The two other models for the phase A dataset also had similar learning curves showing no overfitting or underfitting.

D. Ensemble model

After the LR, SVM, and TREE models were optimized, they were assembled into one classifier by soft (with probabilities) majority voting. The results of the individual classifier along with the ensemble model for phase A dataset and full dataset are shown in Table V. As shown in this table, both the phase A and full dataset had a slight decrease in the accuracy of their corresponding majority voting model compared to the best individual model.

E. Confusion Matrix

For all the models, including the ANN/DNN, confusion matrices were produced. The confusion matrix of the decision tree model for the phase A dataset is shown in Figure 6. It shows the number of True Positive (TP) on the top left quadrant, False Negative (FN) on the top right, False Positive (FP) on the bottom left, and True Negative (TN) on the bottom right. From these values, other metrics, including accuracy, can be calculated. In Figure 6, Precision = [TP/(TP+FP)], Recall =[TP/(TP+FN)] and F1 score =[2*(Precision*Recall)/ (Precision + Recall] [22] are calculated and shown on the plot title.

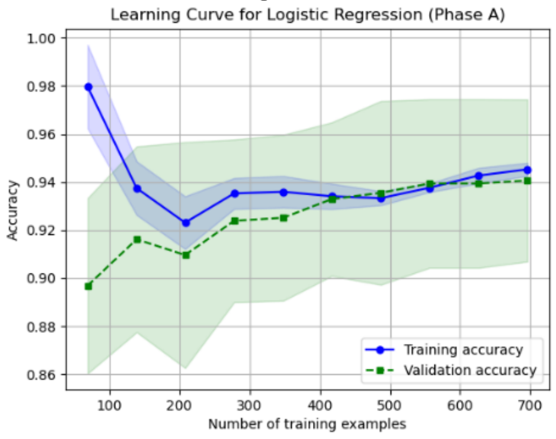

Figure 5. Learning curve for logistic regression model of phase A dataset.

As shown in this figure, the decision tree model correctly classified most of the labels (shown in blue quadrants) while it wrongly classified eight samples of each of the two labels (shown in white quadrants). SVM was the best model for the phase A dataset, while for the full dataset, LR was the best.

F. ROC AUC Curve

The Receiver Operating Characteristics (ROC) curve is a graph of model probabilities of False Positive Rate (FPR) versus True Positive Rate (TPR). FPR is the ratio of the number of False Positives (FP) to the total number of negatives (FP+TN), while TPR is the ratio of True Positive (TP) to the total number of positives (TP+FN) [23]. ROC curve shows a model's performance at all classification thresholds, and the Area Under this Curve (AUC) provides a metric for this performance measure [23]. Figure 7 shows the ROC curve and ROC AUC scores of LR, SVM, TREE, and Majority voting models for the phase A dataset. Please note that the straight diagonal line in the middle of the plot is a hypothetical model which cannot distinguish between the two classes and is equivalent to "guessing" the classification. Therefore, its AUC is 0.5. This diagonal line represents a threshold, and if a model falls below this threshold, it is performing worse than guesswork. As seen in Figure 7, our LR, SVM, TREE and Majority voting models' respective curves are close to AUC of 1. The performance of these models in this metric is very similar.

G. ANN/DNN model and its Loss Curve

After testing the three basic ML algorithms, we created an ANN/DNN model with the phase A dataset. The number of hidden layers in this model, number of nodes in each layer, activation functions for each layer, optimizer, and hyperparameters for the model were all tuned and optimized by trial and error. The results of this optimization are shown in Table VI.

With these parameters, the ANN/DNN model was trained with phase A data, and it produced a final training accuracy of 98.19% and testing accuracy of 94.29%. These accuracy values are slightly better than the corresponding accuracy values of LR, SVM, TREE, or Majority voting models. Figure 8 shows the training versus test (validation) curve of this ANN model. As shown in this figure, the model's loss decreased and stabilized as the model trained for more epochs.

TABLE V. ACCURACY VALUES FOR INDIVIDUAL AND ENSEMBLE MODEL IN PHASE A AND FULL DATASET

Classifiers	Phase A Dataset Accuracy	Full Dataset Accuracy
Logistic Regression	0.94	0.77
SVM	0.93	0.76
Decision Tree	0.92	0.74
Majority Voting	0.93	0.76

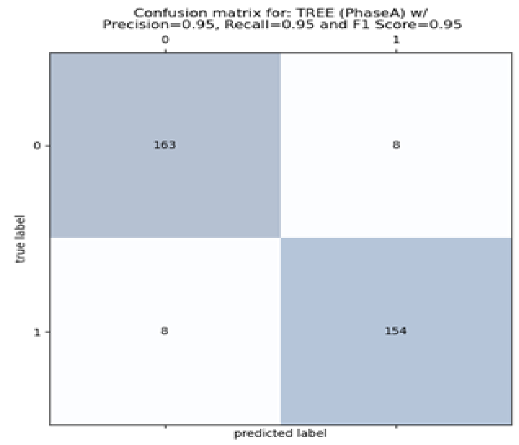

Figure 6. Confusion matrix of the decision tree model for phase A dataset.

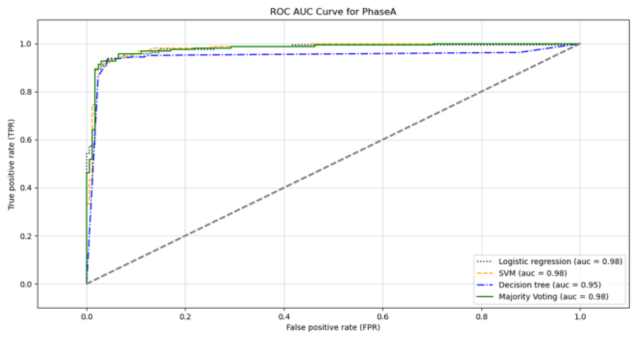

Figure 7. ROC AUC curve of LR, SVM, TREE and majority voting model.

TABLE VI. ANN/DNN OPTIMIZED HYPERPARAMETER VALUES

Number of hidden layers	2
Number of nodes in each hidden layer	50, 50
Activation function for each layer	tanh, tanh, sigmoid (for output layer)
Optimizer	Adam
Optimizer parameters	Learning rate of 0.01 and beta decay (beta_1) of 1e-5
Number of epochs	100
Validation ratio	0.01

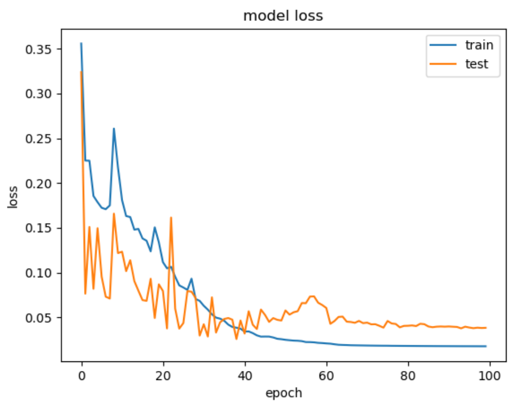

Figure 8. Training and testing (validation) loss curve for the ANN model.

IV. Conclusion

This research study extracted time-domain (amplitude envelope and RMS energy) and frequency-domain (spectral centroid) feature from raw PLC data to generate an ML-ready dataset. Then, we used the dataset in three supervised machine learning algorithms: logistic regression, support vector machine, and decision tree, to generate classification models. We optimized these models using the grid search method, investigated impactful features in each model using sequential backward analysis, checked for model's overfitting and underfitting using learning curves, and used accuracy, ROC AUC scores, precision, recall, and f1 score metrics to evaluate and compare the performance of the models. Using these performance metrics, we found out that all three models (and an ensemble model made by majority voting of the three) performed similarly, with SVM being slightly better than the rest because of its non-linear classification.

Then, we used an artificial neural network/deep neural network model with two hidden layers to perform the same classification task on the PLC dataset. This ANN model performed slightly better than the aforementioned basic ML models.

We also observed that all the models performed significantly better with the standalone phase A dataset than the full dataset containing data from all three phases. This is most likely because the input signal was initially transmitted through the phase A power line, and the phases B and C only got images of this signal along the PLC path. Hence, the deteriorated signal data in phases B and C diluted the full dataset and caused the model to be less accurate. In future works, the signal reception from the secondary phases can be improved, for example, by using a Rake receiver. This could result in a better performance from the full dataset.

References

[1] K. Dostert, "Telecommunications over the power distribution grid–possibilities and limitations," IIR-Powerline, vol. 6, no. 97, 1997.

[2] A. M. Tonello, N. A. Letizia, D. Righini, and F. Marcuzzi, "Machine Learning Tips and Tricks for Power Line Communications," IEEE Access, vol. 7, pp. 82434–82452, 2019, doi: 10.1109/ACCESS.2019.2923321.

[3] S. U. Ercan, O. Ozgonenel, Y. E. Haj, C. Christopoulos, and D. W. P. Thomas, "Power line communication design and implementation over distribution transformers," in 2017 10th International Conference on Electrical and Electronics Engineering (ELECO), Nov. 2017, pp. 190–194.

[4] D. Kakati and S. C. Arya, "A full-duplex optical fiber/wireless coherent communication system with digital signal processing at the receiver," Optik, vol. 171, pp. 190–199, Oct. 2018, doi: 10.1016/j.ijleo.2018.05.140.

[5] T. Smyth, "Amplitude Envelopes", Department of Music, UCSD, 2019. [Online]. Available from: http://musicweb.ucsd.edu/~trsmyth/sinusoids171/Amplitude_Envelopes.html [retrieved: April, 2021]

[6] B. Boashash, "Chapter 4 - Advanced Time-Frequency Signal and System Analysis," in Time-Frequency Signal Analysis and Processing (Second Edition), Oxford: Academic Press, 2016, pp. 141–236.

[7] Energy and RMSE. musicinforetrieval.com. [Online]. Available from: https://musicinformationretrieval.com/energy.html#:~:text=The%20root%2Dmean%2Dsquare%20energy,x%2C%20sr%20%3D%20librosa [retrieved: April, 2021]

[8] J. M. Grey and J. W. Gordon, "Perceptual effects of spectral modifications on musical timbres.," Journal of the Acoustical Society of America, vol. 63, pp. 1493–1500, May 1978.

[9] sklearn.linear_model.LogisticRegression. Scikit-learn. [Online]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html [retrieved: April, 2021]

[10] sklearn.svm.SVC. Scikit-learn. [Online]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html [retrieved: April, 2021]

[11] sklearn.tree.DecisionTreeClassifier. Scikit-learn. [Online]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html [retrieved: April, 2021]

[12] sklearn.model_selection.GridSearchCV. Scikit-learn. [Online]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html [retrieved: April, 2021]

[13] sklearn.ensemble.VotingClassifier. Scikit-learn. [Online]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html [retrieved: April, 2021]

[14] sklearn.metrics.roc_auc_score. Scikit-learn. [Online]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html [retrieved: April, 2021]

[15] sklearn.metrics.precision_score. Scikit-learn. [Online]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html [retrieved: April, 2021]

[16] sklearn.metrics.recall_score. Scikit-learn. [Online]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html [retrieved: April, 2021]

[17] sklearn.metrics.f1_score. Scikit-learn. [Online]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html [retrieved: April, 2021]

[18] Plotting Learning Curves. Scikit-learn. [Online]. Available from: https://scikit-learn.org/stable/auto_examples/model_selection/plot_learning_curve.html [retrieved: April, 2021]

[19] sklearn.metrics.confusion_matrix. Scikit-learn. [Online]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html [retrieved: April, 2021]

[20] S. Raschka, Python machine learning. Packt publishing ltd, 2015.

[21] sklearn.feature_selection.SequentialFeatureSelector. Scikit-learn. [Online]. Available from: https://scikit-learn.org/dev/modules/generated/sklearn.feature_selection.SequentialFeatureSelector.html [retrieved: April, 2021]

[22] K.P. Shung. Accuracy, Precision, Recall or F1? Towards data science. Mar. 15, 2018. [Online]. Available from: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9 [retrieved: April, 2021]

[23] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," arXiv preprint arXiv:2010.16061, 2020.