

# Final Report

Purdue University

**Supervisor:** Prof Stanley H. Chan, stanleychan@purdue.edu

**Graduate Member:** Benjamin Vondersaar

**Undergraduate Members:** Christopher Chow, Gregory Dykes, Alexandria Moore, Xiangyu Qu, Samuel Sowell, Siqing Wei, Fangjia Zhu

## Executive Summary

We report a *multi-harmonic histogram* method for extracting and analyzing electric network frequency (ENF) signals to identify power grids. Given a voltage-time measurement of a power grid with a base frequency  $f_0$ , we compute the ENF signals at multiple harmonic locations  $kf_0$  and extract (i) a histogram of the magnitudes of the ENF; (ii) a histogram of the signal power and noise power surrounding the ENF; (iii) a histogram of the signal-to-noise-ratio (SNR) of the ENF. Different from existing methods which use spectral-combining techniques to compress multiple harmonics into one single ENF, the proposed multi-harmonic histogram method retains significantly richer information which allows one to perform classification. At the classification stage, we propose a histogram matching method with a multi-layer decision rule for identification. On the practice data set consisting of 50 testing samples, we achieve 92% overall accuracy.

## 1 Introduction

This report studies the usage of Electric Network Frequency (ENF) as a mean to classify multimedia signals observed over different power grids [1]-[2]. ENF is the supply frequency of the electric network. In North America, the nominal ENF value is 60Hz, whereas in other countries, the nominal ENF value is 50Hz. Nominal ENF values are typically constants over time and are thus not very useful. The one we use is the *instantaneous* ENF, which has been considered as an important classification signature of power grids [3] because every power grid has a distinctive flow dynamics and frequency control scheme. Besides power, audio recordings are also used as ENF exists and can be extracted from the audio recordings [4]. The goal of this year's SP cup competition is to classify 9 power grids based on the power and audio recordings, and to build a sensing circuit to collect power recordings.

Grid classification based on ENF analysis has been previously reported in [3, 4, 5]. The basic idea is to perform the following steps in sequence: (i) Extract the ENF signal from the raw data; (ii) Extract features (e.g., mean, variance, wavelet coefficients, etc) of the ENF signal; (iii) Build a support vector machine (SVM) for classification, and return "None of the above" if the confidence score is below certain threshold. This method works reasonably well for power recordings, but we observe the following limitations.

- Harmonics. ENF exists at multiple harmonics. While it is possible combine all harmonics into one single ENF using the spectral combining technique [6], we find that the performance gain is limited.
- Weak ENF. In some audio recordings, the ENF is surrounded by heavy noise. Extracting the ENF and finding meaningful features are challenging.

- Limited Training. There are only two half-hour recordings for audio training data, which is inadequate in terms of variety and duration. Training on these data will likely lead to overfit considering the number of features (16 as reported in [5]).

The method we present in this report mitigates the above problems by using a different approach. Our approach, called the multi-harmonic histogram classification method, bypasses the need of combining multiple harmonics into one single ENF, the need of extracting ENF features as in [6]-[5], and the need of using an SVM. On the practice data set consisting of 50 testing samples, we achieve 92% overall accuracy.

The rest of this report is organized as follow. In Section 2 we discuss the extraction of the ENF signals and show examples of the multi-harmonic histograms. In Section 3 we discuss the classification system. Section 4 presents the hardware circuit we built and analysis results of the data collected.

## 2 Extraction of ENF Signals

In this section we discuss how ENF signals are extracted. We will also discuss how to extract the multi-harmonic histogram from the ENFs.

### 2.1 ENF Extraction

ENF is the instantaneous frequency of a given signal. Starting with the voltage-time signal  $x(t)$  with discrete time samples  $x(t_1), \dots, x(t_N)$ , we compute the discrete short-time Fourier transform (STFT) to obtain

$$X(f_\ell, t_n) = \sum_{\tau=-\infty}^{\infty} x(\tau)w(\tau - t_n)e^{-j2\pi f_\ell \tau},$$

where  $w(t)$  is a window function, and  $f_\ell$  denotes the  $\ell$ -th frequency at which the STFT is computed. We call  $X(f, t)$  the spectrogram of the input signal  $x(t)$ , and we define  $S(f, t)$ :

$$S(f, t) \stackrel{\text{def}}{=} 20 \log_{10} |X(f, t)|$$

as the instantaneous power spectral density.

Extracting ENF from  $S(f, t)$  can be done by identifying the frequency at which the maximum  $S(f, t)$  occurs, averaged with its neighboring values [6]:

$$f_{\text{ENF}}(t) = \frac{\sum_{\ell=L_1}^{L_2} S(f_\ell, t)f_\ell}{\sum_{\ell=L_1}^{L_2} S(f_\ell, t)}, \quad (1)$$

where

$$L_1(t) = \frac{(f_{\max}(t) - \Delta)n_{\text{FFT}}}{F_s}, \quad L_2(t) = \frac{(f_{\max}(t) + \Delta)n_{\text{FFT}}}{F_s},$$

with  $F_s$  being the sampling frequency (1000Hz in this report),  $\Delta$  is the frequency perturbation (typically 3 Hz),  $n_{\text{FFT}}$  is the number of Fast Fourier Transform (FFT) points ( $8F_s$  in this report with a window of length  $10F_s$ ), and  $f_{\max}(t) \stackrel{\text{def}}{=} \underset{f}{\text{argmax}} |S(f, t)|$  is the frequency at which maximum  $S(f, t)$  occurs. An example of the extracted ENF is shown in Figure 1.

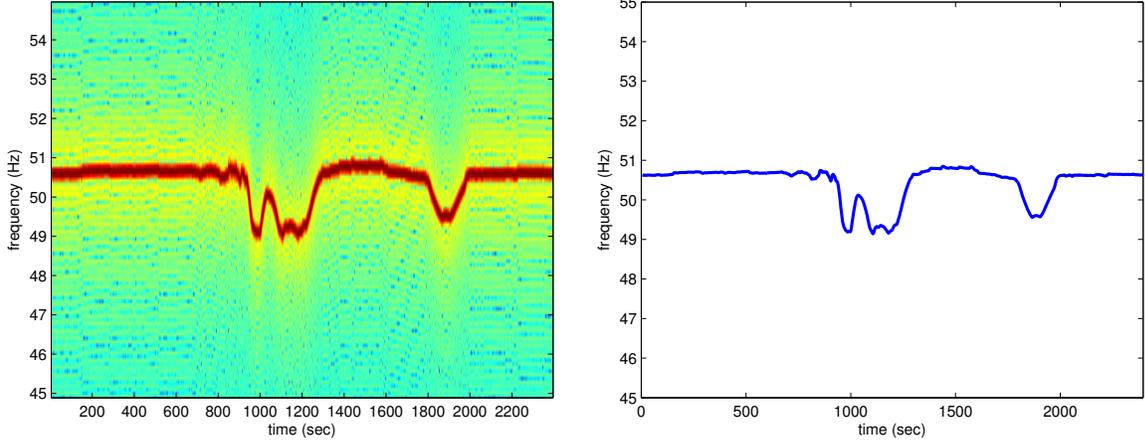


Figure 1: (a) Spectrogram of Grid B-1 / power recording. (b) Extracted ENF.

## 2.2 Multi-Harmonic Histogram

The ENF extracted in (1) is a vector of frequencies  $\mathbf{f}_{\text{ENF}} = [f_{\text{ENF}}(t_1), \dots, f_{\text{ENF}}(t_N)]^T$ . For large  $N$ , one can treat these entries  $\{f_{\text{ENF}}(t_1), \dots, f_{\text{ENF}}(t_N)\}$  as samples drawn from a non-parametric distribution  $p(f)$ . Suppose we discretize  $p(f)$  into  $L$  bins,  $p(f_\ell)$  is then the count of elements falling into the  $\ell$ -th bin ( $\ell = 1, \dots, L$ ), yielding

$$p(f_\ell) = \sum_{n=1}^N \mathbb{I}(f_{\text{ENF}}(t_n) \in [f_\ell - \delta, f_\ell + \delta]), \quad (2)$$

where  $\delta$  defines the width of each bin. The non-parametric function  $p(f_\ell)$  is called the base-frequency histogram of the ENF.

Multi-harmonic histogram is a simple extension of the base-frequency histogram. Suppose there are  $K$  harmonics in the spectrogram, we define the  $k$ -th harmonic histogram as

$$p_{\text{freq}}^{(k)}(f_\ell) = \sum_{n=1}^N \mathbb{I}(f_{\text{ENF}}^{(k)}(t_n) \in [f_\ell - \delta, f_\ell + \delta]), \quad k = 1, \dots, K \quad (3)$$

where  $f_{\text{ENF}}^{(k)}(t)$  is the ENF extracted around the  $k$ -th harmonic. For power recordings, we observe that only odd harmonics (i.e.,  $k = 1, 3, 5, 7$ ) are significant.

The frequency histogram  $p(f_\ell)$  is a summary statistics of *where* the peak frequencies are located. This alone is inadequate for the classification because it does not tell *how good* these peak frequencies are. To increase the set of useful statistics, we consider three additional histograms, namely the signal histogram, the noise histogram, and the signal-to-noise histogram.

At every time stamp  $t$ , we define  $v^{(k)}(t)$  as the signal strength of the  $k$ -th harmonic:

$$v^{(k)}(t) \stackrel{\text{def}}{=} \sum_{\ell \in \Omega_v} |X(f_\ell, t)|, \quad (4)$$

where  $\Omega_v = [(kf_0 - \Delta), (kf_0 + \Delta)]n_{\text{FFT}}/F_s$ , with  $f_0$  being the base-frequency. Similarly, we define the noise strength as

$$\eta^{(k)}(t) \stackrel{\text{def}}{=} \sum_{\ell \in \Omega_\eta \setminus \Omega_v} |X(f_\ell, t)|, \quad (5)$$

where  $\Omega_\eta = [(kf_0 - \Delta'), (kf_0 + \Delta')]n_{\text{FFT}}/F_s$ , for  $\Delta' > \Delta$ . Intuitively,  $v^{(k)}(t)$  and  $\eta^{(k)}(t)$  can be thought of as the sum of the spectral values in two different regions:  $\Omega_v$  is the set of frequencies that contains significant “signal”;  $\Omega_\eta \setminus \Omega_v$  is the set of frequencies that contains only “noise”. With  $v^{(k)}(t)$  and  $\eta^{(k)}(t)$  we can also define the signal to noise ratio at the  $k$ -th harmonic as

$$r^{(k)}(t) \stackrel{\text{def}}{=} \frac{v^{(k)}(t)}{\eta^{(k)}(t)}. \quad (6)$$

Consequently, we can define the signal histogram  $p_{\text{signal}}^{(k)}(v)$ , the noise histogram  $p_{\text{noise}}^{(k)}(\eta)$ , and the signal-to-noise histogram  $p_{\text{SNR}}^{(k)}(r)$ . Figure 2 illustrates the histograms of the signal-to-noise ratio. It is clear that for different harmonics, the histograms of each grid behave drastically different.

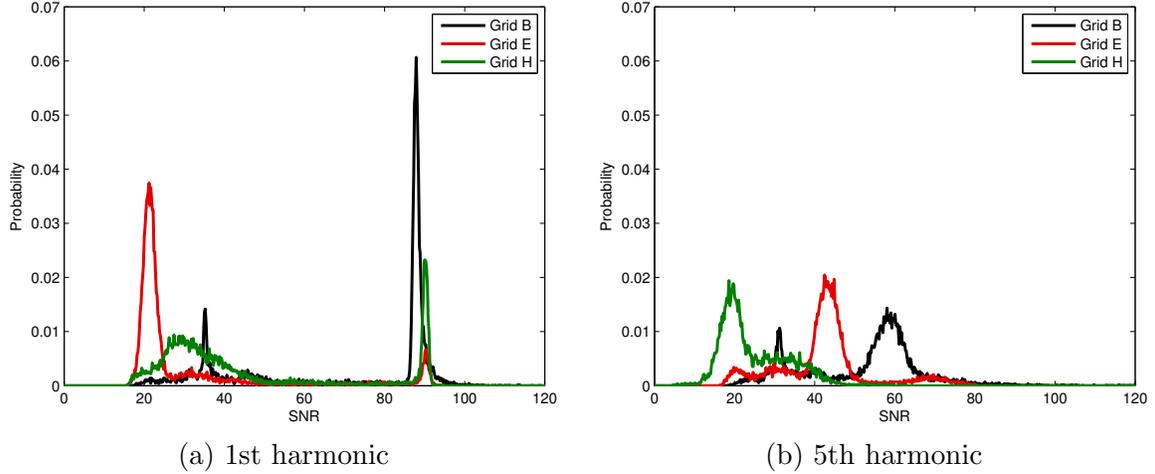


Figure 2: Histograms of the signal-to-noise ratio for power recordings B, E, H.

**Remark:** The four histograms  $p_{\text{freq}}^{(k)}(f_\ell)$ ,  $p_{\text{signal}}^{(k)}(v)$ ,  $p_{\text{noise}}^{(k)}(\eta)$  and  $p_{\text{SNR}}^{(k)}(r)$  all have different number of bins and ranges. Specific values can be seen in the MATLAB attachment.

### 3 Location Classification / Identification System

In this section we present our classification system. Our classification system consists of three major steps: (1) Preliminary classification to determine whether the signal is power or audio, 50Hz or 60Hz; (2) Multi-harmonic histogram comparison to return scores for each test; (3) Decision making to make the final decision.

#### 3.1 Power/Audio, 50Hz/60Hz

Classification between power and audio is based on the observation that power signals usually have significantly stronger signal to noise ratio than audio signals, especially around the harmonics. Therefore, our classification method is to compute the signal strength and noise strength at a set of designated frequency locations  $K = \{50, 60, 100, 120, 150, 180, 200, 240, 250, 300, 350, 360, 400, 420, 450\}$  Hz. We define the signal and noise power as

$$\text{signal power} = \sum_{k \in K} \sum_{n=1}^N v^{(k)}(t_n), \quad \text{noise power} = \sum_{k \in K} \sum_{n=1}^N \eta^{(k)}(t_n). \quad (7)$$

If the signal power is significantly larger than the noise power, we return the decision that the grid is power. Otherwise it is audio. This can be summarized as

$$\text{signal power} \geq_{\text{audio}}^{\text{power}} \text{noise power}.$$

Classification between 50Hz versus 60Hz is done similarly. The idea is to compare the signal to noise ratio at harmonics of 50Hz and harmonics at 60Hz. Letting  $K_{50} = \{50k \mid k = 1, 2, 3, 4\}$  and  $K_{60} = \{60k \mid k = 1, 2, 3, 4\}$ , we compute

$$r_{60} = \sum_{k \in K_{60}} \sum_{n=1}^N r^{(k)}(t_n), \quad r_{50} = \sum_{k \in K_{50}} \sum_{n=1}^N r^{(k)}(t_n), \quad (8)$$

and make the decision based on

$$r_{60} \geq_{50\text{Hz}}^{60\text{Hz}} r_{50}.$$

**Remark:** In practice, the summations in (7) and (8) are approximated using a single FFT on the entire recording data. A single FFT is computationally more efficient than short time Fourier transform.

### 3.2 Multi-Harmonic Histogram Comparison

The main classification is based on a list of scores measuring the distances between the trained histograms and the testing histograms. Given a training histogram  $p(x)$  and a testing histogram  $q(x)$ , we use the  $\mathcal{L}^1$ -norm as the distance metric, defined as

$$d(p, q) = \|p - q\|_{\mathcal{L}^1} \stackrel{\text{def}}{=} \sum_{\ell=1}^L |p(x_\ell) - q(x_\ell)|,$$

where  $L$  specifies the number of bins in the histograms  $p(x)$  and  $q(x)$ . We have also experimented with other distance measures such as the KL divergence. But the difference is insignificant.

In our classification system, we have four histograms: (i) frequency location; (ii) signal strength; (iii) noise strength; (iv) signal to noise ratio. Each of these are repeatedly calculated for all harmonic locations. Specifically, for every power grid  $i = 1, \dots, 9$  (stands for Grid A to Grid I), and every harmonic  $k = 1, \dots, K$ , we define the following four distance scores

$$\begin{aligned} D_{\text{freq}}(k, i) &= d\left(p_{\text{freq},i}^{(k)}, q_{\text{freq}}^{(k)}\right), & D_{\text{signal}}(k, i) &= d\left(p_{\text{signal},i}^{(k)}, q_{\text{signal}}^{(k)}\right), \\ D_{\text{noise}}(k, i) &= d\left(p_{\text{noise},i}^{(k)}, q_{\text{noise}}^{(k)}\right), & D_{\text{SNR}}(k, i) &= d\left(p_{\text{SNR},i}^{(k)}, q_{\text{SNR}}^{(k)}\right). \end{aligned}$$

To aggregate the  $K$  harmonics into one single score, we compute the maximum over all harmonics:

$$\begin{aligned} \tilde{D}_{\text{freq}}(i) &= \max_k D_{\text{freq}}(k, i), & \tilde{D}_{\text{signal}}(i) &= \max_k D_{\text{signal}}(k, i), \\ \tilde{D}_{\text{noise}}(i) &= \max_k D_{\text{noise}}(k, i), & \tilde{D}_{\text{SNR}}(i) &= \max_k D_{\text{SNR}}(k, i). \end{aligned}$$

Clearly,  $\tilde{D}_{\text{freq}}(i)$ ,  $\tilde{D}_{\text{signal}}(i)$ ,  $\tilde{D}_{\text{noise}}(i)$  and  $\tilde{D}_{\text{SNR}}(i)$  are scores of the  $i$ -th grid ( $i = 1, \dots, 9$ ).

### 3.3 Final Decision

The final decision is based on the following combined score:

$$\widehat{D}(i) = 3\widetilde{D}_{\text{freq}}(i) + \widetilde{D}_{\text{signal}}(i) + \widetilde{D}_{\text{noise}}(i) + \widetilde{D}_{\text{SNR}}(i). \quad (9)$$

Here, the factor 3 in front of  $\widetilde{D}_{\text{freq}}(i)$  is used to emphasize the equal importance of the frequency histogram as compared to the other three histograms.

To incorporate the case of “None of the above”, we set a simple threshold on the scores:

$$\widehat{D}(i) = \begin{cases} \infty, & \widetilde{D}_{\text{signal}}(i) > \alpha, \text{ or } \widetilde{D}_{\text{noise}}(i) > \alpha, \text{ or } \widetilde{D}_{\text{SNR}}(i) > \alpha \\ \widehat{D}(i), & \text{otherwise} \end{cases} \quad (10)$$

for some threshold  $\alpha$  (We choose  $\alpha = 0.995$ ). The final decision is

$$i^* = \underset{i}{\operatorname{argmin}} \widehat{D}(i). \quad (11)$$

If  $\widehat{D}(i) = \infty$  for all  $i = 1, \dots, 9$ , we return “N” as none of the above.

### 3.4 Modification for Audio

For audio recordings, we observe that it is more robust to locate the first  $K$  peaks instead of a fixed set of  $K$  harmonics. Moreover, the score is computed using the mean-squared summation instead of the maximum

$$\widetilde{D}_{\text{freq}}(i) = \left( \sum_k D_{\text{freq}}(k, i)^2 \right)^{1/2}. \quad (12)$$

The corresponding  $\alpha$  is tuned as  $\alpha = 0.8$ .

### 3.5 Testing Results

On the practice dataset, our algorithm returns the following sequence

AHCNF, BGIND, AFBDC, IINAE, HBBAD, NGNGB, DDNHG, EAIHI, EHECF, FNGEI

This yields 46/50, i.e., 92% correctness. The wrongly classified grids are:

- 4: (power) Ours: F; True: N
- 17: (power) Ours: I; True: N
- 26: (audio) Ours: N; True: C
- 33: (audio) Ours: N; True: C

Based on this result, we observe that the proposed algorithm is less reliable when encountering “None of the above”.

On the actual competition dataset, our algorithm returns the following sequence

NDDND, NNDAF, ANGBG, BFCEH, GHHNG, HFDAI, DNFHI, IECBD, EIIBE, FGNAG  
NIIIG, HAEFC, CNFDG, CECGI, EICDN, BDBHA, DINNG, AABIH, NNDBA, GBFB

## 4 Circuit Design and Data Analysis for ENF Acquisition

In this section we discuss the hardware implementation for the sensing circuit.

### 4.1 Circuit Design

The ENF data was acquired using a circuit that consists of a digital to analog (A/D) converter provided by the Arduino Uno, a clock provided by Raspberry Pi, and a few discrete components, as shown in Figure 3 and Table 1. Within a closed box we have the transformer in series with a fuse connected through a cord to the wall outlet. The transformer steps down the outlet voltage to about 20V peak to peak ( $V_{pp}$ ). This then connects to a circuit constructed on a prototyping shield for the Arduino.

Component	Function
1k $\Omega$ Resistor (R1)	Current limiting resistor for the status LED
3.9k $\Omega$ Resistor (R2)	Pull down resistor for the push button
10k $\Omega$ Resistor (R3)	Pull down resistor for the clocking signal
1k $\Omega$ Resistor (R4)	Current limiting resistor for the power LED
10k $\Omega$ Resistor (R5)	Resistors R5 and R6 are used to further step down the voltage to 5V peak to peak
33k $\Omega$ Resistor (R6)	Refer to R5
First 470 $\Omega$ Resistor (R7)	Resistors R7 and R8 are used to add a 2.5V bias to the signal so that the signal oscillates between 0 and 5V
Second 470 $\Omega$ Resistor (R8)	Refer to R7
.01 $\mu$ F Capacitor (C)	Decoupling capacitor for the Arduino input
Push Button (SW)	Push button to start and stop data capture
Status LED (LED <sub>r</sub> )	LED to signal when the circuit is capturing data
Power LED (LED <sub>g</sub> )	LED to signal when the Arduino shield has power
Transformer (T)	Transformer to step down the voltage to about 20V peak to peak
.5A Fuse (F)	The fuse was added to insure that no part of the circuit would be destroyed from a short circuit

Table 1: Discrete components

The Arduino shield has a collection of resistors and a capacitor, which work to insure the signal is readable by the Arduino. Two resistors (R5 and R6) are implemented to step down the transformer output voltage from 20V<sub>pp</sub> to 5V<sub>pp</sub> centered at 0V. Resistors R7 and R8 are then used to provide a 2.5V bias to the circuit so that signal being fed to the Arduino oscillates between 0 and 5V. The .01 $\mu$ F was added as a decoupling capacitor for a proper reading from the Arduino. The shield also has a push-button switch to activate the measurement code on the Arduino. One red LED tells the user data is being captured, and a green LED shows that there is power to the shield.

### 4.2 Data Analysis

We collected power recordings in West Lafayette, Indiana on Jan 5, 2016. A snapshot of 20 minutes of the data is shown in Figure 4. The data sampling frequency for the device we use is 1093 Hz, which can be improved to match 1000 Hz if we use a dedicated clock instead of the Raspberry Pi. In Figure 4, we observe that our collected data is clean and produces a high quality spectrogram.



any of these claims. One possibility to cause the ambiguity is that although “C” is “US East” (based on our guess), it may be recorded in a different part of the US East than our data (which is collected in central Indiana). Some local behavior of the grid may cause some unexpected results to the grid classification problem.

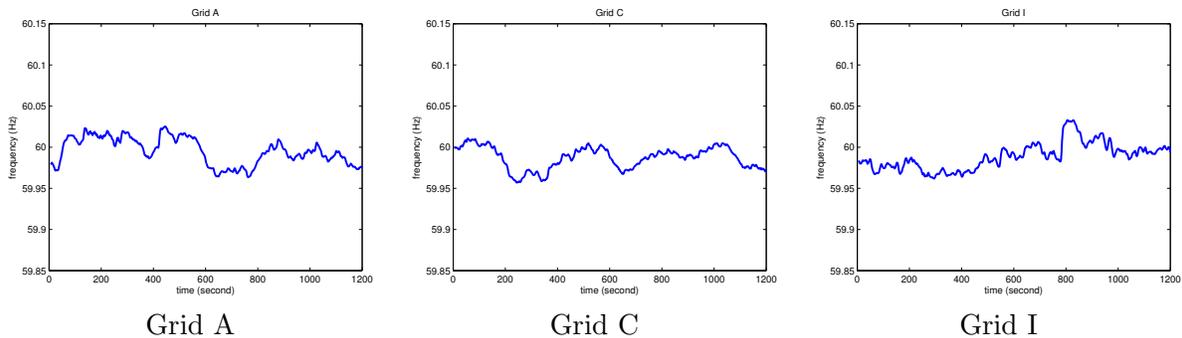


Figure 5: Extracted base ENF of Grid A, Grid C and Grid I.

## 5 Conclusion

We presented a multi-harmonic histogram method for classifying 9 power grids from its power and audio recordings. The multi-harmonic histogram method demonstrated effective and robust classification performance on the practice dataset. While it is unclear about the algorithm’s performance on the actual competition data, the failure cases on the practice data show that the algorithm is less reliable on classifying “C” and “N”.

On the hardware design, we built a sensing circuit and measured power recordings in West Lafayette, Indiana. Our circuit consists of a transform, an Arduino Uno for A/D conversion, a Raspberry Pi for clock, and a number of discrete components. Compared to the training datasets, we believe that our collected data best matches with Grid C, although our algorithm returns “N”.

## Appendix: MATLAB Code Overview

The MATLAB consists of two major parts: (1) Training; (2) Testing. The suggested procedure of using the MATLAB code is to first run

```
>> trainAudio; trainPower;
```

In both files, the folder name of the training data should be changed accordingly. We assume that the training data follows the naming structure as the one posted online. When running `trainAudio.m` and `trainPower.m`, a folder `./hist_data/` storing the trained histograms will be created in the current directory.

To test the algorithm, we suggest running the script `main.m`. Running `main.m` will display the estimated output on the screen.

The two testing functions `testPower` and `testAudio` require trained histograms (which should have been stored in `./hist_data/`). The output argument of the two functions include various scores, which are only used for debugging.

## References

- [1] C. Grigoras, “Applications of ENF analysis in forensic authentication of digital audio and video recordings,” *J. Audio Eng. Soc.*, vol. 57, no. 9, pp. 643–661, 2009.
- [2] R. Garg, A. L. Varna, A. Hajj-Ahmad, and M. Wu, “‘seeing’ ENF: Power-signature-based timestamp for digital multimedia via optical sensing and signal processing,” *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 9, pp. 1417–1432, Sep. 2013.
- [3] A. Hajj-Ahmad, R. Garg, and M. Wu, “Instantaneous frequency estimation and localization for ENF signals,” in *Asia-Pacific Signal Inf. Process. Association Annual Summit and Conf. (APSIPA-ASC’12)*, Dec. 2012, pp. 1–10.
- [4] H. Su, R. Garg, A. Hajj-Ahmad, and M. Wu, “ENF analysis on recaptured audio recordings,” in *IEEE Intl. Conf. Acoustics, Speech and Signal Process. (ICASSP’13)*, May 2013, pp. 3018–3022.
- [5] A. Hajj-Ahmad, R. Garg, and M. Wu, “ENF-based region-of-recording identification for media signals,” *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1125–1136, Jun. 2015.
- [6] A. Hajj-Ahmad, R. Garg, and M. Wu, “Spectrum combining for ENF signal estimation,” *IEEE Signal Process. Lett.*, vol. 20, no. 9, pp. 885–888, Sep. 2013.

## Team Biography

**Prof Stanley H. Chan** is currently an assistant professor of Electrical and Computer Engineering at Purdue University. He supervises the team and provides guidelines to the members.

**Benjamin Vondersaar** is a master student of Electrical Engineering at Purdue University. He assists Prof Chan in supervising the team.

**Christopher Chow** is a junior Computer Engineering student at Purdue University. He is responsible for designing filters to reject noise and extract signals for the team. Christopher is currently studying abroad at Nanyang Technological University Singapore in Spring 2016.

**Gregory Dykes** is a junior Electrical Engineering student at Purdue University. He is a circuit engineer of the team who designs, implements and tests the sensing circuit.

**Alexandria Moore** is a junior Electrical Engineering student at Purdue University. She is the original author of the multi-harmonic histogram method and is involved in the development and analysis of the method. Alexandria is currently studying abroad at National University of Singapore in Spring 2016.

**Xiangyu Qu** is a junior Electrical Engineering student at Purdue University. He helps the team by studying the spectral combining techniques, analyzing the audio datasets, and implements the SVM approach.

**Samuel Sowell** is a junior Electrical Engineering student at Purdue University. He is responsible for programming Arduino and Raspberry Pi to control the A/D converter and the I/O of the circuit.

**Siqing Wei** is a junior Electrical Engineering student at Purdue University. He helps the team by analyzing the performance of the SVM and the multi-harmonic histogram method.

**Fangjia Zhu** is a junior Electrical Engineering student at Purdue University. He studies the original SVM method as suggested by the project guideline, and develops a number of decision ideas for the multi-harmonic histogram method.