Neural Network Recognition of Frequency Disturbance Recorder Signals

Authors Name/s per 1st Affiliation *(Author) – Do this later after figure out whose names go where*

*Abstract*—The ability to determine the location a digital audio recording was made through frequency imprints from a nearby electrical grid is important to forensics. This application requires both a reference grid frequency database and the ability to recognize the local source of such effects. While the former requirement is in the process of being satisfied with the Frequency Monitoring Network (FNET), this paper attempts to solve the latter with Neural Network (NN). Local factor measured by frequency disturbance recorders (FDRs) are first preprocessed using Fast Fourier Transform (FFT) for feature extraction. Principle Component Analysis (PCA) is then optionally used to reduce dimensionality of inputs. NNs demonstrate decent accuracy within a small window of time from their training period but their accuracy quickly degrades after roughly a day. PCA is found to cause negligible difference in accuracy while reducing training time. The geographic location of sensors and NN accuracy are investigated and are found to be unrelated.

Keywords—Frequency Disturbance Recorder (FDR); Frequency Monitoring Network (FNET); Neural Network (NN); Principle Component Analysis (PCA); Fast Fourier Transform (FFT)

# Introduction

Previous work has demonstrated the application of the Frequency Monitoring Network (FNET) data toward digital audio authentication by comparing frequencies captured by an audio recording device connected to the grid via outlet with a frequency database such as that provided by FNET. Misalignment between the recording and database frequencies suggest that the recording has been tampered with such as the insertion or deletion of a sound clip. The frequency caused by the grid is extracted by using a bandpass filter to remove frequencies outside of a neighborhood around 60 Hz and then using a Fast Fourier Transform (FFT) to approximate the frequency over time [1]. This study explores the possibility of retrieving more information from this extracted frequency, specifically with approximating the location a recording was taken by comparing local differences observed by different frequency disturbance recorders (FDRs). A recording affected by the grid should contain local factors that a nearby FDR would also be affected by. Direct comparison between FNET data and the recorded signal is computationally expensive, especially with a larger amount of sensors, and highly sensitive to differences between the extracted frequency from the audio sample and the best matching FDR due to the closeness of FDR signals. This motivates the need for a more efficient method of recognition that is able to recognize key features of a signal without being distracted by noise.

The problem is difficult due to the synchronization and regulation of frequency across the grid, making differences between different FDR’s small. For four units taken from the Eastern Interconnect (EI), mean square errors for durations of both a second up to 24 hours are of order 10-5 Hz [1]. Additionally, the concept’s current applicability is limited due to the partial coverage of the grid due to the relative newness of the frequency network. Practical application would require higher density of measurement, since the proposed method approximates location by matching with the nearest FDR. With the level of power system measurement growing with time, the concept will be useful in the near future.

The study addresses the problem with neural networks (NNs). NNs are supervised machine learning tools useful for function approximation. They can be used for pattern recognition and classification given an appropriate choice of transfer functions, error function, and training algorithm. They exhibit good decision making ability when classifying noisy data and do not require that the trainer understand the mechanics behind boundaries between classes. Their hidden layers of customizable size enable them to be trained to approximate nonlinear functions. Applications of NNs towards the power grid include early warning of grid failure [3] and load forecasting [4] based off of grid and environment measurements. The application of NNs for classification of signals is less prevalent in power systems, though they are frequently used in other fields such as the classification of EEG signals as belonging to thinking or relaxing patients [5] or the classification of sound samples as originating from voices, machinery, or vehicles [6].

The time needed to train a NN is directly related to the size of its inputs due to the required number of input layer weights being proportional to the size of inputs. This motivates the use of a dimensional reduction technique such as principle component analysis (PCA). The transformation reduces the dimensionality of input vectors by producing an ordering of principle components such that the latter components contribute negligibly to variation in the transformed space.

The methodology used by this paper is similar to the feature extraction method via Fast Fourier Transform (FFT) used by [6]. Section II provides an in depth description of the preprocessing steps involved in treating the data for the NN as well as a brief discussion on hidden layer size. Section III presents a case study on the accuracy of the NN under different amounts or location of training time relative to test data and accuracy with reduced component number, as well as a discussion on the relationship between classification ability and geographic location of sensors. Conclusions are presented in Section IV.

# Methodology



(a)



(b)

Fig. 2. Example of median subtraction for 5 units for a typical input vector of 512 samples. (a) Raw data and median. (b) Same data with median subtracted.

The NN requires several stages of preprocessing. The architecture of the NN used is also discussed. A flowchart showing the ordering of steps taken to evaluate outputs is given in Fig 1. Classification requires preprocessing and NN evaluation, which are indicated as major elements of the flowchart. Each of the three components of preprocessing is described below in the following subsections, as well as the inner mechanics of the NN.

## Time Matching and Median Subtraction

Since the local factors affecting the signal are of interest instead of the synchronized state of the grid, relevant information is extracted from the signals by subtracting the median of all signals in the interconnection for each point in sampling time. The concept of average grid frequency is explored in [2] as a weighted average of the frequencies of the interconnected generators based off of each generator’s inertia. The median of several units is chosen to approximate grid frequency due to its robustness against events where measurements may not reflect the state of the grid such as islanding. The need for the median to serve as an accurate representation of synchronized interconnection frequency is one reason why a large interconnection with several frequency recorders is needed, making WECC and EI potential targets. An example is provided in Fig. 2.

## Fast Fourier Transform (FFT)

The FFT performed over a moving window for each unit in the interconnection under study is used to vectorize the median subtracted data. A window length of 512 samples was chosen because the Fast Fourier Transform performs better with samples of power 2 and this number of samples is nearly a minute of real time data. There was a 256 sample overlap between successive windows to inflate the number of vectors. The output of the FFT is a complex 512 point spectrum of the energy and phase of frequency bins with the th bin referring to a frequency of Hz. The absolute value of this vector is taken, and as the latter 256 points are the complex conjugates of the former, they are discarded. The first point, which encodes the DC component is also dropped, leaving a 255 point vector to be fed to the neural network.



Fig. 1. Flow chart indicating the steps of computation leading from inputs in the form of FDR signals to outputs of FDR Unit IDs as well as latitude, longitude data. Principle Component Analysis is optional and dotted in the chart.

If no data is available for a unit at a certain time, the vector(s) belonging to that unit at that time are dropped and not considered for neural network training or testing. Interpolation as done in [1] to deal with missing data is not applicable when the median of the data set cannot be used as a reference. Since this component is subtracted off as part of preprocessing, doing so in this case is equivalent to inserting a 0 into that vector.

## Principle Component Analysis (PCA)

PCA is an optional preprocessing step that was found to reduce the training time of the neural network. Typically used to reduce the number of variables in a relation for purposes of making data able to be displayed in 3 or fewer dimensions, it is used here to simplify network inputs. By transforming a data set onto its principle components, components responsible for less variation in the data set can be dropped to reduce the dimensionality of the data set. In applying this to the neural network, the principle components of the training data set were discovered and the cumulative proportion of variation provided by each of the 255 components is shown in Fig. 3. The first 100 of these principle components was found to account for 88% of variation, so the loadings of these components were kept as a 100x255 coefficient matrix. The neural network is then trained with a data set transformed by this matrix such that it takes in an input of length 100 rather than 255. This decreases the number of weights in the input layer of the neural network by a proportional amount, reducing its size and training time. After a network has been trained with this setup, future inputs of 255 elements must first be multiplied by this matrix before being fed into the network. As the first action of the neural network is to multiply its 100 length input by its input weight matrix, this is equivalent to applying the product of the input weight and PCA matrices to the original 255 length input. Since the neural network without PCA had a 100x255 element input matrix, using this product does not change the efficiency of classification.

## Network Architecture and Training

A multi-layer perceptron (MLP) was implemented in Matlab and is briefly described here. The NN architecture is composed of two layers. Inputs are preprocessed within the NN by a Matlab function mapminmax, which maps the values of each input variable to [1,-1] during training (the same mappings are applied to later inputs for each variable). The mapped inputs are then multiplied by an input weight matrix and added to an input bias vector associated with the network’s hidden layer. The dimensions of the weight matrix and bias vector are determined by the size of input vectors and the NN’s hidden layer’s node count. The hidden layer’s node count is the responsibility of the network designer. A higher node count increases the number of weights in the neural network which requires a longer training time, but increases the ability of the neural network to partition data sets into separate classes. Having too high of a node count also decreases the generalizability of the neural network making it less effective at classifying inputs outside of its testing set. Fig. 4. shows the fraction of error a NN had on adjacent test sets with varying hidden layer sizes, which is minimized around 100 nodes. The postprocessing function used for the hidden layer is the hyperbolic tangent sigmoid transfer function (tansig), which is implemented in Matlab as



Fig. 4. Error with hidden layer size



Fig. 3. Cumulative percent of variance explained in one day’s data

|  |  |
| --- | --- |
|  |  |

Hidden layer outputs are then passed to the output layer. The output layer also has a weight matrix and bias vector, though the dimensions of these are determined by the hidden layer size and the number of outputs of the NN. The transfer function used by the output layer is the soft max (softmax) function,

|  |  |
| --- | --- |
|  |  |

for a set of inputs of size and an index such that . This function maps inputs into range (0,1] while preserving their ordering. This better matches the format of target vectors that are members of the set , the standard basis vectors of where each index is associated with a FDR unit id and location. Each output of the neural network is matched to the closest target vector, which is equivalent to choosing the output index of maximum value.

A training set of input and target vectors must be provided to fit the NN to the problem. An interval of training time is decided for which all the raw data provided from FDRs for the interconnection under study during this interval are preprocessed according to the steps described above, and these input vectors are provided alongside target vectors indicating their FDR source to the NN for training. Training vectors are randomly assigned to training, validation, or test sets. The NN is trained according to the scaled conjugate gradient backpropagation algorithm (traincsg) with cross-entropy as the performance indicator. Network training was allowed to continue until a maximum number (6) of continuous increases in validation set error were allowed to occur to prevent overgeneralization.

# Case Study

Test sets are produced to evaluate the accuracy of a NN. Test sets contain vectors preprocessed to be evaluated by the NN as described in Section II as well as their corresponding target vectors. Similarly to the training set, a test set contains vectors covering all FDRs in an interconnection for a given time interval. The time that a test set is provided from is given by a date and an optional time. A date without a time indicates the test set takes vectors from that day from 12:00:00 AM to 12:00:00 AM the next day in UTC.

This paper uses two measures to judge the effectiveness of a NN for any particular test set. The general accuracy of the NN is measured by percentage of test input vectors correctly matched to corresponding target vectors, while more detail is provided in the test set’s confusion matrix. The confusion matrix details the number of vectors originating from one source FDR that were classified as another FDR for all combinations of two FDRs within the interconnection. The former metric is used to evaluate network performance under a variety of training and testing conditions. The latter, which measures the rate of misclassification between units, is of interest in establishing the similarity of units from the network’s perspective, and is compared with geographic data.

## Network Accuracy over Time

The accuracy of the NNs in terms of percentage correct over time is presented in Table 1 for EI. The time interval of data used for training each NN is all of a single day ranging from June 2nd to June 6th, 2013 (Sunday through Thursday), with the number of available units in EI at this time being 72. The percentage associated with the same day as the training set is the accuracy of the network on the testing subset of the vectors provided during network training. Because this test set is made up of a random sample of vectors from within the training interval, it is called adjacent. The adjacent test set is then the first measurement in every row. Each of the other accuracy percentages is reflective of that network’s performance on that entire day. Finally, the network is tested on data from June 1st and 2nd from 2014 to demonstrate network degradation one year into the future. June 1st is chosen as well as June 2nd since June 2nd of 2013 and June 1st of 2014 are both Sundays.

The network displays good accuracy (around 90%) for adjacent test sets but performs poorly on test sets at later times. A network’s accuracy on a test set taken a day after training is typically higher than accuracy on later days, which tend to have accuracy slightly above 60 percent. Exceptions to these trends exist, especially for networks trained and tested near the weekends. An example would be the network trained on June 2nd, which had higher performance on June 8th and 9th compared to earlier days. Of note is that June 2nd, 8th, and 9th all fall on weekends, which suggests that higher accuracy over a longer time period might be attainable if training time is tailored to the test set.

## Network Accuracy with Limited Training

It is shown that frequency behavior of the grid varies according to time based off of hour of day, day of week, and time of year [7]. This motivates the use of more specific intervals of training time in order to potentially yield higher accuracy for later test sets. Because the NN quickly becomes unable to differentiate between FDRs in a short amount of time, the possibility was studied under a small time scale of a 1 week difference between training and test set. Table 2 shows the accuracy of NNs trained with intervals of training data from June 2, 2013 varying from one hour to two days. Accuracy of the NN is provided for both adjacent data and data from a week later taken on June 9th during the same time interval.

The results shown in the table indicate small differences in adjacent sets and larger losses for the weeklong difference for short training times. The difference might be explained by the larger amount of training data provided for training and test sets with larger intervals.

## Accuracy with PCA

Reducing input dimension with PCA also reduces the total information provided to the NN, so a decrease in accuracy is expected in exchange for shorter network training time, both dependent on the number of principle components dropped. The accuracy difference due to PCA is demonstrated in Table 3, in which the performance two NNs, one with and one without PCA preprocessing, are compared for different test sets. The results show negligible loss in accuracy with PCA, meaning it is safe for use in reducing training time.

## Confusion and Geographic Location

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Day of Training** | **Day of Testing - 2013** | | | | | | | |  | **2014** | |
| ***June 2*** | ***June 3*** | ***June 4*** | ***June 5*** | ***June 6*** | ***June 7*** | ***June 8*** | ***June 9*** |  | ***June 1*** | ***June 2*** |
| June 2 | 92.07 | 84.96 | 66.49 | 64.31 | 62.35 | 64.31 | 73.92 | 72.93 |  | 44.30 | 42.86 |
| June 3 | - | 91.26 | 67.39 | 64.64 | 65.10 | 64.18 | 72.20 | 67.32 |  | 44.74 | 43.25 |
| June 4 | - | - | 91.92 | 74.09 | 73.31 | 66.12 | 63.25 | 57.48 |  | 36.51 | 41.56 |
| June 5 | - | - | - | 89.10 | 78.35 | 71.69 | 67.71 | 60.34 |  | 37.22 | 41.55 |
| June 6 | - | - | - | - | 92.07 | 73.94 | 69.05 | 62.20 |  | 39.84 | 44.32 |

Table I. Training Accuracy Over Time (%)

|  |  |  |  |
| --- | --- | --- | --- |
| **Neural Network** | **Day of Testing** | | |
| ***Adjacent*** | ***Next Day*** | ***1 Week*** |
| No PCA | 92.07 | 84.96 | 72.93 |
| With PCA | 92.96 | 84.39 | 72.03 |

Table III. Training Accuracy with PCA (%)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Set** | **Training Duration (hours)** | | | | |
| ***1*** | ***5*** | ***10*** | ***24*** | ***48*** |
| Adjacent | 94.78 | 92.25 | 92.60 | 92.07 | 89.21 |
| 1 Week | 60.65 | 68.88 | 69.04 | 72.93 | 74.14 |

Table II. Training Accuracy with Limited Training (%)

A qualitative analysis is performed by plotting the locations of FDRs on a US map and forming edges between units that are misclassified as each other by a NN. The confusion matrix produced by Matlab after NN testing is a natural adjacency matrix of such a graph after some processing. If is the confusion matrix of a test set, the edge weight between different units and is defined to be

|  |  |
| --- | --- |
|  |  |

which is the number of inputs originating from classified as over the number of inputs correctly classified as . The addition of 1 prevents division by zero in case no vectors were correctly classified. Note this definition makes the graph directed. Equation (3) is preferable to directly using the entries of because it accounts for an unequal distribution of inputs among the units.

Each unit is also assigned a value measuring its relative ability to be recognized by the NN. For unit this value is

|  |  |
| --- | --- |
|  |  |

Because EI had roughly 70 units active during the time period under study and small amounts of misclassification between units is likely, the graph produced is dense. To improve visibility, edge are colored and made transparent based on their relative weight magnitude. Large weights correspond to opaque red coloring while small weights have transparent blue colorings. The directed nature of edges is indicated by gradient coloring, with the coloring of an edge closer to a unit referring to the weight directed from that unit to the other unit on that edge. For a user defined edge threshold and rate parameter

|  |  |
| --- | --- |
|  |  |

defines an intensity function that continuously maps edge weights on the positive real line to an interval within and, for sufficiently large thresholds, close to (0,1). This intensity is used to scale color and transparency because it has the desirable traits that values near the threshold change steadily while values far from the threshold approach upper and lower bounds. Values of greater than a half are more red while values lower than a half are more blue, with complete opaqueness occurring at 1 and transparency at 0. Likewise, the size and color of unit nodes in the graph are scaled by this function with replacing with lower node values being smaller and green while high values are larger and red. The graph for a network trained on June 2 and tested on some of the sets used in Table I is shown in Fig. 5.

An alternative undirected edge weighting formula is

|  |  |
| --- | --- |
|  |  |

which produces lower weights for previously one-sided edges compared to edges with similar confusion in both directions. This makes such edges become transparent under the coloring scheme, as seen in Fig. 5. The appearance of these edges in Fig. 5 (a) and their disappearance in (b) indicate that a large amount of misclassifications made by the NN are of the kind where one FDR is mistaken as another. These misclassifications are one-sided and do not seem related to location.

# Conclusions and Future Work

The potential application of location authentication of digital audio samples makes the problem of recognizing FDR signals worthwhile. This paper attempts to address the problem with training NNs to recognize these signals. These networks are able to recognize small sets of units within a short time interval, but struggle with measurement networks large enough to measure the interconnections and have low accuracy further in time from the training interval. Accuracy is affected by the relationship and duration of training and testing time intervals in a nonobvious way. The networks have slightly better performance with longer training time, though this drastically increases the amount of time needed to train the network. This can be mitigated somewhat with dimensional reduction techniques such as PCA with little change in accuracy. Lastly, the NNs tend to have certain units that they are consistently unable to classify correctly, and as such are often misclassified as other units. These misclassifications go against the expectation that FDRs that are geographically near each other have the highest misclassification, and is disadvantageous for location finding.

Much work is needed to make this approach feasible. Future studies should explore what other information besides the frequency spectrum of a FDR signal is necessary for recognition. Optimum relationships between training and testing intervals should also be found. Alternative methods of classification besides NN should also be explored.

# Acknowledgment



(a)



(b)

Fig. 5. Graphs of NN confusion trained on June 2nd and tested on June 3rd. (a) Directed weighting with .. (b) Undirected weighting with same thresholds and parameters

This work was supported primarily by the Engineering Research Center Program of the National Science Foundation and the Department of Energy under NSF Award Number EEC-1041877 and the CURENT Industry Partnership Program.

References

[1] Yuming Liu; Chenguo Yao; Caixin Sun; Yilu Liu, "The Authentication of Digital Audio Recordings Using Power System Frequency," Potentials, IEEE , vol.33, no.2, pp.39,42, March-April 2014

[2] Chan, M.L.; Dunlop, R.D.; Schweppe, F., "Dynamic Equivalents for Average System Frequency Behavior Following Major Distribances," Power Apparatus and Systems, IEEE Transactions on , vol.PAS-91, no.4, pp.1637,1642, July 1972

[3] Jinying Li; Yuzhi Zhao; Jinchao Li, "Power Grid Safety Evaluation Based on Rough Set Neural Network," Risk Management & Engineering Management, 2008. ICRMEM '08. International Conference on , vol., no., pp.245,249, 4-6 Nov. 2008

[4] Hao-Tian Zhang; Fang-Yuan Xu; Long Zhou, "Artificial neural network for load forecasting in smart grid," Machine Learning and Cybernetics (ICMLC), 2010 International Conference on , vol.6, no., pp.3200,3205, 11-14 July 2010

[5] Gope, C.; Kehtarnavaz, N.; Nair, D., "Neural network classification of EEG signals using time-frequency representation," Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on , vol.4, no., pp.2502,2507 vol. 4, July 31 2005-Aug. 4 2005

[6] Stoeckle, S.; Pah, N.; Kumar, D.K.; McLachlan, N., "Environmental sound sources classification using neural networks," Intelligent Information Systems Conference, The Seventh Australian and New Zealand 2001 , vol., no., pp.399,403, 18-21 Nov. 200

[7] Markham, P.N.; Yilu Liu; Bilke, T.; Bertagnolli, D., "Analysis of frequency extrema in the eastern and western interconnections," Power and Energy Society General Meeting, 2011 IEEE , vol., no., pp.1,8, 24-29 July 2011