Coloured Noise Signal Identification using Supervised Learning Algorithm

Mahsa Nourbagheri
Department of Electrical and Computer Engineering
Oakland University
Rochester, MI
Email: Nourbagheri.mahsa [AT] gmail.com

Mohamed Zohdy
Department of Electrical and Computer Engineering
Oakland University
Rochester, MI

Abstract— **This paper presents a supervised learning algorithm that investigates how a computer program would discern the similarities and differences between six kind of colored noise sound signals. By preprocessing a set of known input noise signals and extracting the features of each signal in time-domain and frequency domain, the algorithm classifies the known input dataset into corresponding known classes. Different methods are used to verify the stability and accuracy of the predictions. In the testing stage, new unknown dataset is given to the classifier to predict the corresponding class of the input noise signal.**

Keywords- sound; colored noise; machine learning; supervised learning; classification; feature extraction

I. INTRODUCTION

Supervised learning method can be used to predict and classify between noise signals. The learning process of the supervised learning method is dependent on the presence of a teacher. One of the use case of supervised learning method is, when a set of input and corresponding output responses are known. Many pattern recognition systems could partitioned into several steps such as sensing, segmentation, feature extraction, classification and post-processing [1]. By using the knowledge of the input-output samples or so called training dataset, we can make more reasonable predictions about the original data set and the resulting classifier is then used to label the unknown instances [3]. The purpose of the learning process is not to get optimal prediction and classification of the data, but to provide insight into the nature of the target concept [2], [4]. First step in supervised learning is to prepare sample data. All supervised learning methods start with an input data matrix. Each column and row represents variables and observation respectively. The response data would be a column vector where each row contains the output of the corresponding observation in the input data. Supervised machine learning classification methods have been used in different areas such as online reviews [6], accelerometers [7], Gaussian processes [8]. To analyse and train the supervised learning model, different classification algorithms such as Decision tree, Discriminant analysis (Linear and Quadratic), Nearest Neighbour were applied.

Sound is a wave of positive and negative pressure disturbances. When the molecules of the air vibrate, the ear perceives these vibrations in the pressure as sound and converts it to mechanical energy and then into nerve impulses which the brain can interpret through middle ear and inner ear and that is how human discern sound [9]. But how computers would recognize the difference between sounds, is what this paper will investigate. Noise signal is unwanted sound that is intense and therefore it has the same features as sound. Any kind of filtered noise signal is called 'coloured noise' that implies to impure white noise. Noise colours are characterised by colour type because of the particular frequency of that noise. For example, the sound wave of pink noise has the same frequency of light waves that create pink colour. By extracting the features from sounds and using a machine learning algorithm method, we can predict the colour of the noise. In [2] T.Bryant and M.Zohdy present a self-organizing feature map to display discriminate clustering between 6 types of coloured noise sound tracks. Self-organizing feature map is an unsupervised machine learning method that is used in many different applications such as malware detection [10], intelligent hearing assistance [11], engine health diagnostics [12] and GPS [13].

Coloured noise signals have been generated using MATLAB Simulink. In this paper, the key features extracted from time domain and frequency domain for each generated noise signal and then a supervised learning algorithm has been developed to classify between the different sample colours of noise sound tracks to generate a model (learning phase). Later the model is used to predict the colour of each input sound track from the unseen data set (application phase).

II. MACHINE LEARNING METHOD

A. *Feature Extraction*

In the pre-processing stage, an algorithm has been developed to extract features from both training dataset and new dataset in time domain and frequency domain. We have considered eight key features from time and frequency domain. In time domain the mean and variance of the noise signal were calculated using the equations (1) and (2):

$$\mu = \frac{1}{B}\sum_{b=1}^{B} A_b \qquad (1)$$

$$V = \frac{1}{B-1}\sum_{b=1}^{B} |A_b - \mu|^2 \qquad (2)$$

Where A is the vector of noise signal, B is the scalar observation of noise signal (samples), μ is the mean value and V is the variance for the input noise signal. We have also considered the maximum and minimum values as the key features of the noise signal in time domain.

In the frequency domain the estimate of the power spectral density (PSD) of the input noise signal is obtained. Estimate of PSD is the Fourier Transform of the biased estimate of the autocorrelation sequence. For the input signal, sampled at $f_s = 44{,}100\ Hz$ the estimate of PSD of input signal for the frequencies in radian/sample is defined as:

$$\hat{P}(f) = \frac{T_s}{B}\left|\sum_{b=0}^{B-1} A_n e^{-j2\pi f b}\right|^2 \quad -\frac{1}{2T_s} < f < \frac{1}{2T_s} \qquad (3)$$

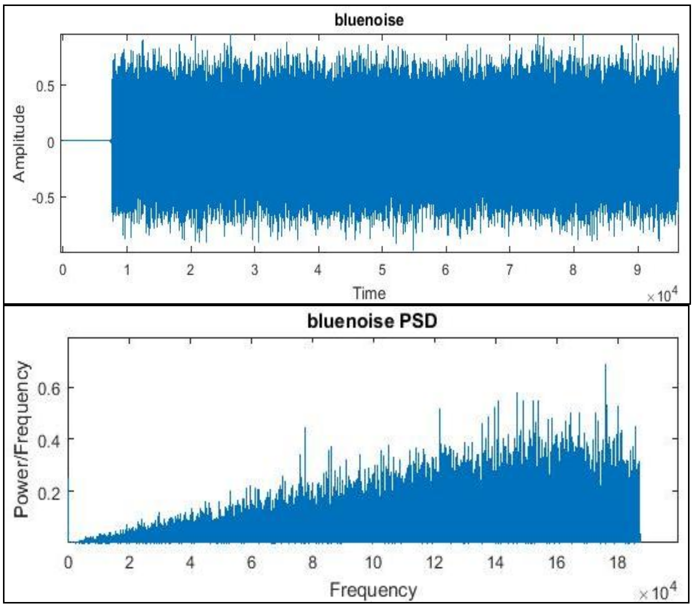

Figure 1. A sample blue noise signal in time and frequency domain

The PSD estimate is a real-valued signal with the frequency range of $0 < f < f_s/2$ and contains N samples. T_s is the sampling time of the input signal. The most important features of the noise signal that considered in frequency domain are, magnitude, median frequency, bandwidth and the power of the spectrum. The power of a signal is the sum of the absolute squares of its time-domain samples divided by the signal length, or in the other word, the sum of squares of its RMS level. The occupied bandwidth is the difference in frequency between the points in PSD estimate, where the integrated power crosses 0.5% and 99.5% of the total power in the spectrum.

B. Classification Algorithm

Machine learning systems typically searches for similarities between the features and specified classes to classify the training set during learning process. To analyse the data set classification, four classification learning algorithms were applied such as Decision Tree, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) and Nearest Neighbour.

Discriminant Analysis Classification Methods can be derived from probabilistic models that assume different classes generate data based on different Gaussian distributions. In the training phase the discriminant classification algorithm estimates the parameters of a Gaussian distribution for each class (prior probabilities $\hat{P}(k)$, means $(\hat{\mu}_k)$ and the covariance matrices$(\hat{\Sigma})$).This method constructs weighted classifiers and estimates the class mean and variance for weighted and unweighted data set. The estimate of the class mean for unweighted data ($\hat{\mu}_k$) and mean for weighted data ($\hat{\mu}_{k_{weighted}}$) are:

$$\hat{\mu}_k = \frac{\sum_{n=1}^{N} M_{nk} x_n}{\sum_{n=1}^{N} M_{nk}} \qquad (4)$$

$$\hat{\mu}_{k_{weighted}} = \frac{\sum_{n=1}^{N} M_{nk} w_n x_n}{\sum_{n=1}^{N} M_{nk}\, w_n} \qquad (5)$$

Where M_{nk} is an element of the N-by-K membership matrix (M), x_n is the observation n of training set and w_n is the positive weight of data. The unbiased estimate of the covariance matrix $(\hat{\Sigma})$ and estimate of weighted covariance matrix $(\hat{\Sigma}_{weighted})$ with assumed sum of weights equals one, are as below:

$$\hat{\Sigma} = \frac{\sum_{n=1}^{N}\sum_{k=1}^{K} M_{nk}(x_n - \hat{\mu}_k)(x_n - \hat{\mu}_k)^T}{N-K} \qquad (6)$$

$$\hat{\Sigma}_{weighted} = \frac{\sum_{n=1}^{N}\sum_{k=1}^{K} M_{nk} w_n (x_n - \hat{\mu}_k)(x_n - \hat{\mu}_k)^T}{1 - \sum_{k=1}^{K} \frac{W_k^{(2)}}{W_k}} \qquad (7)$$

Where W_k is the sum of the weights for class k and W_k is the sum of squared weights per class (k). In the process of fitting the training data, classifier do not use the posterior probability or cost matrix. For both LDA and QDA the class conditional distribution of data $P(x|k)$ is modelled as a multivariate Gaussian distribution with density:

$$P(x|k) = \frac{1}{(2\pi)^n |\Sigma_k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T \sum\nolimits_k^{-1} (x-\mu_k)\right) \qquad (8)$$

The posterior probability $P(k|x)$ estimate defines the probability of an observation x of class k:

$$\hat{P}(k|x) = \frac{P(x|k)P(k)}{P(x)} = \frac{P(x|k)P(k)}{\sum_l P(x|l).P(l)} \qquad (9)$$

Where $P(k)$ is the prior probability and $p(x)$ is the normalization constant which is the sum over class l of $P(x|l).P(l)$. In the LDA, the Gaussian for each class are assumed to have the same covariance matrix and only the mean vary, but in QDA there are no assumption on the covariance matrix and therefore, both mean and covariance of each class vary. Predictions are obtained using Bayes rule as below to minimize the expected classification cost:

$$\hat{y} = arg \min_{y=1,\dots,K} \sum_{k=1}^{K} \hat{P}(k|x)C(y|k) \qquad (10)$$

Where $C(y|k)$ is the cost of classifying an observation as y while the true class is k and it is a K-by-K matrix and each element is either 0 (if observation is in the true class) or 1 (if the observation is not classified in the true class).

Decision Tree Classification Algorithm Method, computes the weighted impurity of each node (t) and estimates the probability that an observation is in node [5] t using equation (11):

$$P(T) = \sum_{j \in T} w_j \qquad (11)$$

w_j is the weight of observation j, and T is the set of all observation indices in node t. Each predictor (feature) that is sorted in ascending order by the Decision Tree classifier is a splitting candidate or cut point. The classifier determines the best way to split node t using predictors by maximizing the impurity gain (ΔI) over all splitting candidates. It splits the observations in node t into left and right child nodes, t_L and t_R respectively. For a particular splitting candidate, if the left and right child nodes contain observation indices in the sets of T_L and T_R respectively, then the impurity gain (ΔI) for that splitting candidate is calculated with equation (12):

$$\Delta I = P(T)i_t - P(T_L)i_{T_L} - P(T_R)i_{T_R} \qquad (12)$$

Nearest Neighbour Classification Algorithm, predicts the class of an input data by finding the number of points and response values(Y) in the training set that are nearest to the new input point and assigns the classification label ($Y_{predict}$) that has the largest posterior probability among the values in known response set. Many different distance matrices can be used to find the distance between a set of data and query points such as Correlation distance, Euclidean distance, Mahalanobis distance, Minkowski metric and etc.

III. Implementation

Data Analysis was carried out using MATLAB software environment for data pre-process such as feature extraction and also algorithm development for classification of data and the application phase which is prediction of new data given to the classifier.

The sample noise signals have been generated using MATLAB Simulink noise signal generator. We considered 6 colours as shown in figure 2, and each class has different spectrum. White noise and pink noise which are used more as reference signals have very close spectrums. We did not consider black noise as it is the total silence and human ear cannot discern it. Red noise, blue noise, violet noise and also grey noise are used.

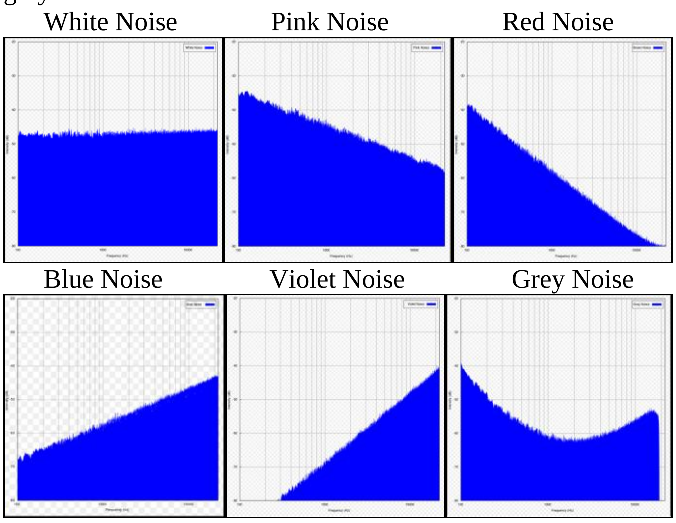

Figure 2. Sample colored noise signal spectrum

This program can read all types of sound files. These files were later pre-processed in MATLAB for feature extraction. Figure 3 shows the extracted features in time and frequency domain. The features extracted in frequency domain are magnitude, median frequency, band width and band power of the power spectral density (PSD) of each signal and features of the time domain are, minimum, maximum, mean and variance of each sample noise signal. The input data to the classifier includes these features with 300 observations.

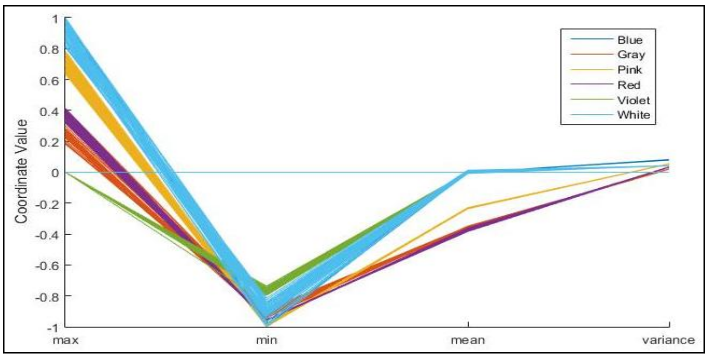

Figure 3. Sample input data features in time domain

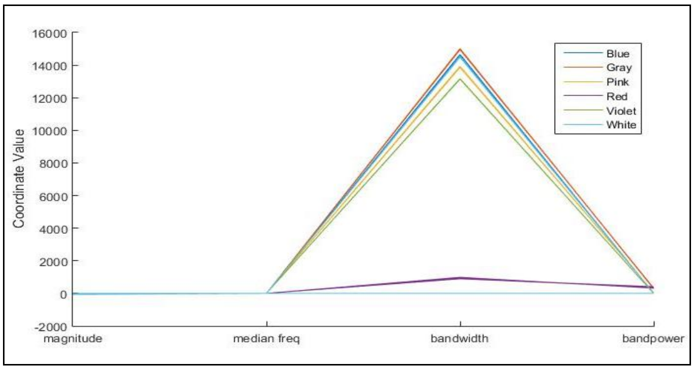

Figure 4. Sample input data features in frequency domain

A. Learning Phase

After extracting the features of each signal and constructing the input training set and corresponding output responses, we used these data sets to train a classification model. At the core, choosing a classification algorithm is a balancing act and the trickiest part of the problem. There is an inherent trade-off between the complexity of the algorithm that can be used and how restrictive the assumptions are about the data. At one extreme we have a simple easy and understanding algorithm (like decision tree, LDA and QDA) which if the data doesn't match with the assumptions algorithm won't give accurate results. At the other extreme complexity we have Bagged Decision tree that supports the vector machines. These models will operate on the wide variety of data sets. Unfortunately the resulting models can be extremely complex. Therefore, simpler is usually better because they are easier to interpret and explain to others. Table 1 shows the characteristics of 4 simple classification models that are used in this research.

TABLE I. CLASSIFICATION METHOD CHARACTERISTICS

Classifier	**Characteristics**			
	Multi class	***Prediction speed***	***Interpretability***	***Memory usage***
Decision Tree	Yes	Fast	Easy	Small
LDA	Yes	Fast	Easy	Small
QDA	Yes	Fast	Easy	Large
Nearest Neighbour	Yes	Medium	Hard	Medium

The classifier has two inputs. One the sample input matrix (X) which is created by extracting 8 features (4 in time domain and 4 in frequency domain). Therefore the training set is a 300-by-8 matrix which the columns represent the number of features and the rows represents the observation of each sample signal. The other input is the known response matrix with 300 rows. Inputs are given to the classifier has the time domain and frequency domain features for 6 different classes as Blue, Grey, Pink, Red, Violet, White.

During the classifying (fitting) process, the sample mean and covariance of each class from the observation computed and the empirical covariance matrix of each class was taken. Figure 4 shows the Q-Q plot that examines the Gaussian mixture assumption.

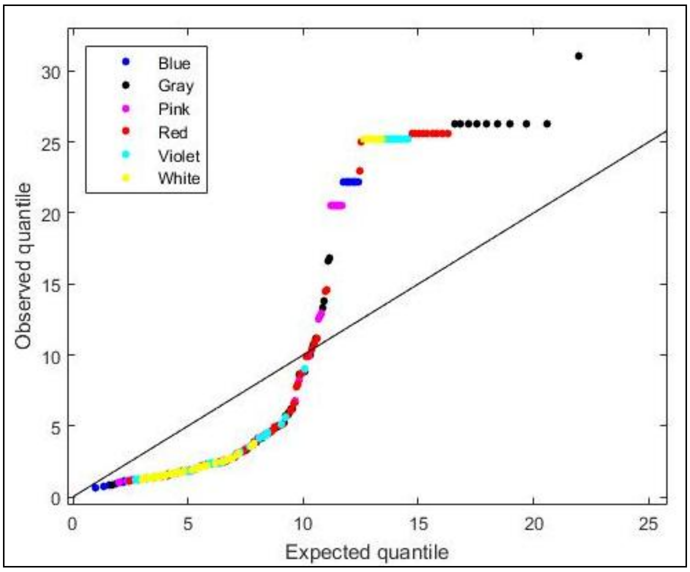

Figure 5. Q_Q plot

Deviation in the plot shows the data has tails heavier than a normal distribution and therefore the data does not match a single covariance matrix that we assumed for LDA. The 45 degree line represents the full agreement between the expected and observed quantiles.

B. Verification of Classifier

After training the classifier with different classifiers, in order to validate the classification, the accuracy of the classifier is tested with different functions such as Misclassification Error, Cross-Validation Error,

Confusion Matrix and Receiver Operating Characteristic ROC curve of MATLAB Statistics Toolbox. Misclassification Error is the difference between the response training data which are the true predictions of the known training set and the predictions that classifier makes of the response based on the input training data. Misclassification Error is often an estimate of the predictive error on the unknown new data.

Confusion matrix is a simple way of visualizing predicted class versus true class that shows how many times misclassification occurs during learning phase. Diagonal of Confusion Matrix represents the correctly classified inputs and anything off the diagonal is misclassified. The table below, shows the Misclassification Error and the Confusion Matrix for four different classifiers used to classify 300 sound signals for 6 classes of colour noise (50 from each class that is given sequentially to the classifier in learning phase). For example, the confusion matrix of decision tree classifier shows that this classification method have misclassified 9 signals from each class to green class.

Normally computational models run into problem because of overfeeding. Cross-validation and feature selection methods are used to reduce size of the model and minimize the set of features that has the same predictive power as the original model. The stability of the classifiers should be verified as the number of predictors is much less than observations. The Prediction Accuracy of the classifiers was evaluated using 10-fold Cross Validation and the results as shown in Table 2.

TABLE II. CLASSIFICATION VERIFICATION WITH CONFUSION MATRIX

Classifier	**Classification Verification**								
	MC	***CV***	***Classes***	***B***	***G***	***P***	***R***	***V***	***W***
Decision Tree	0.15	0.1833	Blue	41	0	0	0	9	0
			Gray	0	41	0	0	9	0
			Pink	0	0	41	0	9	0
			Red	0	0	0	41	9	0
			Violet	0	0	0	0	50	0
			White	0	0	0	0	9	41
Nearest Neighbor	0.15	0.15	Blue	50	0	0	0	0	0
			Gray	9	41	0	0	0	0
			Pink	9	0	41	0	0	0
			Red	9	0	0	41	0	0
			Violet	9	0	0	0	41	0
			White	9	0	0	0	0	41
Linear Discriminant Analysis	0.15	0.1800	Blue	41	0	9	0	0	0
			Gray	0	41	9	0	0	0
			Pink	0	0	50	0	0	0
			Red	0	0	9	41	0	0
			Violet	0	0	9	0	41	0
			White	0	0	9	0	0	41
Quadratic Discriminant Analysis	0.15	0.173	Blue	50	0	0	0	0	0
			Gray	9	41	0	0	0	0
			Pink	9	0	41	0	0	0
			Red	9	0	0	41	0	0
			Violet	9	0	0	0	41	0
			White	9	0	0	0	0	41

In order to investigate the performance of the classifier, we used Receiver Operating Characteristic (ROC) curve, which shows the trade-off between the true positive rate and false positive rate and equivalently sensitivity versus specificity for different threshold of the classifier output. ROC curves are also used to find the threshold that maximizes the classification accuracy (regions of high sensitivity and high specificity). Figures 6, 7 and 8 show the ROC curve for different classifiers. It shows how these classifiers predict the input data for blue class. The red circle shows the optimal operating point of the ROC curve.

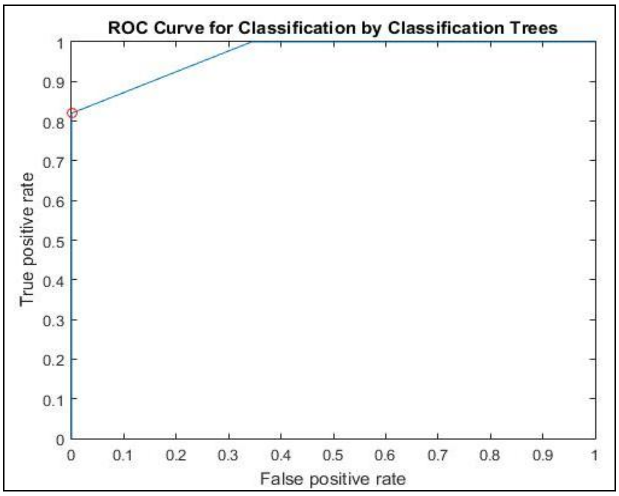

Figure 6. ROC curve by classification tree classifier

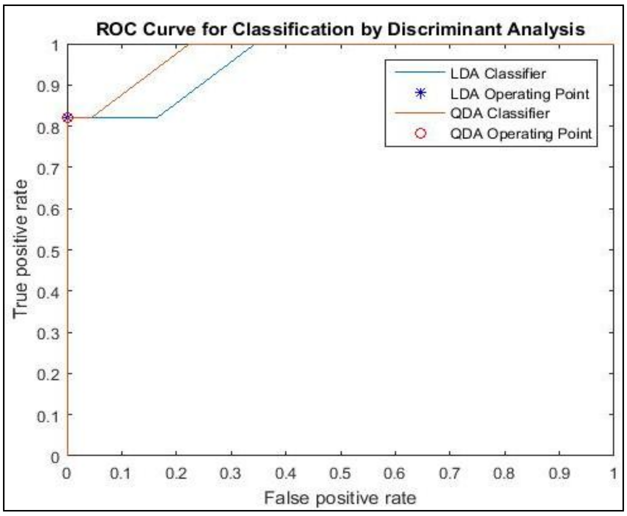

Figure 7. ROC curve by discriminant analysis classifier

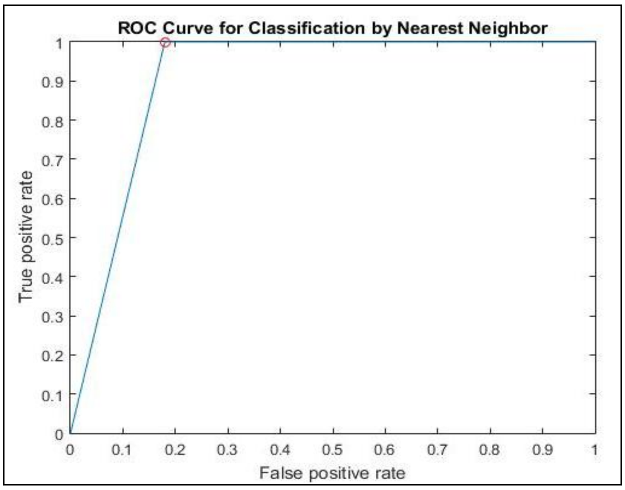

Figure 8. ROC curve by nearest neighbour classifier

IV. Simulation Results and Discussion

Once we trained our classifiers, we can then use it to make predictions about unknown new dataset that contains features of time-domain and frequency domain of different noise signals. In order to test the classification models, eight new unknown datasets were given to the classifiers with different number of observations. Table 3, shows the prediction errors for each classifier while the new input sets has different number. The third column shows the number of truly predicted points from the all given data set. For each classifier we gave a new set of input data of 20, 50, 80,120, 160, 200, 250 and 300 observations. As classifier was trained, the prediction error of the new data set is less than the misclassification in the learning phase.

TABLE III. Testing Results

Classifier	**Stability Verification**			
	Number of inputs	***Prediction Error (%)***	***Number of true predictions***	***Number of false predictions***
Decision Tree	20	0.15	17	3
	50	0.12	44	6
	80	0.137	69	11
	120	0.133	104	16
	160	0.1125	142	18
	200	0.125	175	25
	250	0.112	222	28
	300	0.123	263	37
Nearest Neighbour	20	0.15	17	3
	50	0.12	44	6
	80	0.125	70	10
	120	0.133	104	16
	160	0.1125	142	18
	200	0.135	173	27
	250	0.112	222	28
	300	0.126	262	38
Linear Discriminant Analysis	20	0.1	18	2
	50	0.08	46	4
	80	0.1	72	8
	120	0.1	108	12
	160	0.1	144	16
	200	0.115	177	23
	250	0.1	225	25
	300	0.113	266	34
Quadratic Discriminant Analysis	20	0.2	16	4
	50	0.18	41	9
	80	0.15	68	12
	120	0.15	102	18
	160	0.131	139	21
	200	0.15	270	30
	250	0.144	214	36
	300	0.136	259	41

In order to measure which classifier gives a better prediction of input data sets of colored noise with different number of observation, the accuracy of each classifier is calculated by the number of truly estimated signals to the number of given signals into the classifier. Figure 9 shows that the Linear Discriminant Analysis model with the average accuracy of 90% is more accurate than the other classification methods used in this paper for classifying colored noise signals.

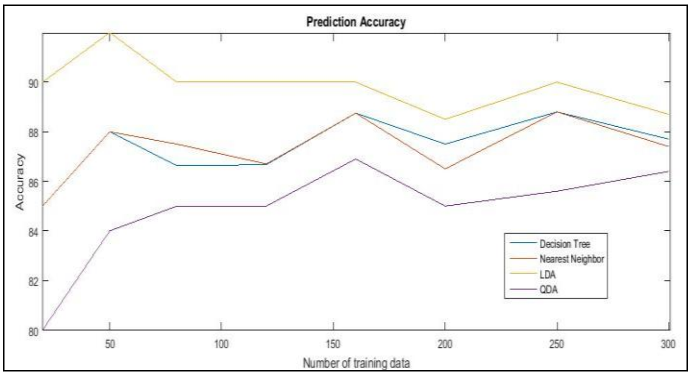

Figure 9. Prediction Accuracy

Conclusion

In this paper, different classification methods (Decision Tree, Linear Discriminant Analysis, Quadratic Discriminant Analysis and Nearest Neighbour) have been examined to classify a set of sample input noise signals with extracted features from time domain and frequency domain with 300 observation (50 observation from six different classes of coloured noise) and later on these trained models were used to predict the color class for new data sets of different coloured noise signals with different numbers of observations. Toghether, the results from testing of the classifiers shows that noise sound signals could be distinguished by the computers with using supervised machine learning methods that could be used in electronics field like signal processing [14] and image processing [15-17]. These simple classification models used in this paper had the accuracy of around 90% for Linear Discriminant Analysis, 83% Quadratic Discriminant Analysis and 87% for Decision Tree and Nearest Neighbor classification methods.

Acknowledgment

This research work was conducted at Oakland University and the authors would like to thank you OU's School of Engineering and Computer Science (SECS) for the research opportunity. Also, Thanks to Wail Hedow for his contribution in the class project level of this research in Estimation and Control Theory class at OU.

References

[1] Richard O. Duda, Peter E. Hart, David G. Stork. Pattern Classification. John Wiley & Sons, Edition 2, 2012

[2] T. Bryant, M. Zohdy, "Noise signal identification by modified self-organizing maps", International Journal of Computer and Information Technology(ISSN: 2279-0764), Vol. 2, Jan. 1, 2014

[3] S.B. Kotsiantis. Supervised Machine Learning: A Review of Classification Techniques. Emerging Artificial Intelligence Applications in Computer Engineering, IOS Press,2007

[4] M. Garg, K. Joshi, "Machine learning approach for feature classification using supervised learning algorithms", National Journal on Advances in Computing and Management, Vol. 2, No.1, April 2011

[5] Coppersmith, D., S. J. Hong, and J. R. M. Hosking. Partitioning Nominal Attributes in Decision Trees. Data Mining and Knowledge Discovery, Vol. 3, 1999, pp. 197–217.

[6] Qiang Ye, Ziqiong Zhang , Rob Law. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. Expert Systems with Applications 36 (2009) 6527–6535

[7] Richard Williamson and Brian J. Andrews. Gait Event Detection for FES Using Accelerometers and Supervised Machine Learning. IEEE TRANSACTIONS ON REHABILITATION ENGINEERING, September 2000

[8] Carl Edward Rasmussen. Gaussian Processes in Machine Learning. Citeseerx, 2006

[9] Hass, Jeffrey. "What is sound?" An Acoustics Primer, chapter 10. N.p., n.d. Web. 2 2013.< www.indiana.edu/~emusic/acoustics/sound.htm > .

[10] 1Richard R. Yang, 2Victor Kang, 3 Sami Albouq and 3Mohamed A. Zohdy. Application of Hybrid Machine Learning to Detect and Remove Malware. Transactions on Machine Learning and Artificial Intelligence. August 2015

[11] Michael Hodges, Mohamed Zohdy. Intelligent Hearing Assistance Using Self-Organizing Feature Maps. . Transactions on Machine Learning and Artificial Intelligence. December 2014

[12] Bryant, T., Hodges, M., & Zohdy, M. Self-organizing maps applied to engine health diagnostics. International Journal of Computer and Information Technology, 2014 3(2): p. 205-211.

[13] Nsour, A., & Zohdy, M., Self-organized learning applied to global positioning system (GPS) data. Proceedings of the 6th WSEAS International Conference on Signal, Speech, and Image Processing, Lisbon, Portugal. 2006.

[14] Abdel-Aty-Zohdy, H. S., & Zohdy, M. A., Neural networks for pattern discovery and optimization in signal processing and applications. , 1995. 1: p. 202-206 vol.1.

[15] RM Khoshki, S Oweis, S Wang, G Pappas, S Ganesan, FPGA Hardware Based Implementation of an Image Watermarking System, International Journal of Advanced Research in Computer and Communication Engineering, Volume3, Issue5, May 2014.

[16] Rohollah Mazrae Khoshki, S Ganesan, Improved ALPR system based on Smart LicensePlate Character Detection Algorithm, International Journal of Advanced Research in Computer and Communication Engineering, Volume 4, Issue 12, December 2015.

[17] Rohollah Mazrae Khoshki, S Ganesan, Improved Automatic License Plate Recognition(ALPR) system based on single pass ConnectedComponent Labeling (CCL)and reign property function, 2015 IEEE International Conference on Electro/Information Technology (EIT), Dekalb, IL, Pages: 426 – 431, May 2015.