FEATURE SELECTION FOR CYCLOSTATIONARY-BASED SIGNAL CLASSIFICATION

Garrett Vanhoy, Noel Teku
Faculty Advisor: Dr. Tamal Bose
University of Arizona
Tucson, AZ, 85721
[gvanhoy, nteku1, tbose] @email.arizona.edu.

ABSTRACT

Cognitive radio (CR) is a concept that imagines a radio (wireless transceiver) that contains an embedded intelligent agent that can adapt to its spectral environment. Using a software defined radio (SDR), a radio can detect the presence of other users in the spectrum and adapt accordingly, but it is important in many applications to discern between individual transmitters and this can be done using signal classification. The use of cyclostationary features have been shown to be robust to many common channel conditions. One such cyclostationary feature, the spectral correlation density (SCD), has seen limited use in signal classification until now because it is a computationally intensive process. This work demonstrates how feature selection techniques can be used to enable real-time classification. The proposed technique is validated using 8 common modulation formats that are generated and collected over the air.

INTRODUCTION

Cognitive radio (CR) and software defined radio (SDR) are a pair of new technologies enabling rapid growth in the field of wireless communications. The concept of radio that can adapt to its environment was formalized by Joseph Mitola and was a revolutionary idea at a time where most radios could only perform in one way. This concept of Cognitive Radio has, at times, been simplified to applications in spectrum sharing where a radio can act as a secondary user to already licensed spectrum, but its applications go far beyond this. With the advent of SDR, which allows a radio to change fundamental transmission parameters such as center frequency and bandwidth, the concept of a fully Cognitive Radio has come closer to reality. SDR technology has developed over the last 20 years and has become a technology that is very necessary in many common devices such as cellular phones. The ultimate aim of CR is to be able to adapt a radio's operation to its environment, hence techniques that enable the understanding of its current operating environment must be developed.

Within an increasingly busy spectrum, a radio must be able to understand how the spectrum is currently being used. There are many existing techniques that allow a radio to determine whether a

particular frequency is in use through measuring the amount of energy present at that frequency and comparing it to the amount of energy present when it is not in use. This technique enables spectrum sharing, but enabling more complex applications requires not only determining if a frequency is occupied, but *who* or *what* is occupying it. This can be accomplished through signal classification techniques. More specifically, the kind of signal classification that must be done here is both **blind** in the sense that no *a-priori* information is present, and **real-time**.

The task of determining the modulation of an incoming signal has been named modulation classification (MC), recognition, or identification [1]. Much of the existing work in this area has been on MC applications involving non-cooperative communications. Non-cooperative communications is when the signal of interest is coming from a transmitter that does not intend for its data to be interpreted by the observing radio. This is opposed to the normal wireless system that is designed to relay the modulation through the use of bits in the beginning of a packet that are of a known modulation. Non-cooperative communications is in general a more challenging scheme than cooperative communications as it is possible to exploit other known properties of the incoming signal.

A. Likelihood-Based Modulation Classification

Modulation classification methods can be largely separated into two types: likelihood-based and feature-based. Likelihood-based methods can be considered optimal in the sense that they achieve the least probability of misclassification [2]. However, this comes at the cost of a computational complexity that eludes real-time implementation in many cases [1]. Likelihood-based methods come in three categories including the Average Likelihood Ratio Test (ALRT), the Generalized Likelihood Ratio Test (GLRT), and the Hybrid Likelihood Ratio Test (HLRT).

The ALRT is named as such because it attempts to look at the averaged distribution of unknown parameters such as noise power and carrier phase offset with known probability density functions (PDF) conditioned on each modulation. This is the most accurate of the likelihood-based approaches and is often used as a theoretical upper bound for the probability of correct classification for other methods, but it is also the most computationally complex. Other types of likelihood-based approaches include the Generalized Likelihood-Ratio Test (GLRT) and the Hybrid Likelihood-Ratio Test (HLRT). The GLRT is less computationally complex than the ALRT, but is less accurate than the ALRT and suffers from being unable to differentiate nested constellations such as 16-QAM and 64-QAM entirely. These two algorithms can be combined to create the HLRT. The GLRT and HLRT computationally more tractable than the ALRT by design, but are still too computationally complex to use for many real-time applications.

B. Feature-Based Modulation Classification

A feature-based method extracts a set of descriptive values from the signal that differentiates each signal from each other. This approach is sub-optimal in terms of probability of correct classification, but substantially reduces the computational complexity. These features can include cumulants, time and frequency domain statistics, wavelet transform coefficients, or a combination of them [3]. Finding the best set of features to accurately identify modulation schemes from each other has been the subject of the bulk of research in the area of MC. The classification stage em-

ploys a variety of machine learning techniques to various degrees of success. The most prominent of these techniques include artificial neural networks, support vector machines, and decision trees.

C. Contributions

Likelihood-based MC establishes a framework capable of performing blind MC, but because of its computational complexity, it ultimately fails to lead to a real-time implementation. Feature-based MC significantly simplifies the signal classification process without sacrificing too much performance and thus enables real-time implementations. However, much of the work in feature-based MC makes prohibitive assumptions about the state of the received signal such as the need for system synchronization. This work proposes an MC design that does not require prohibitive assumptions and thus is both truly blind and enables real-time implementations.

In Section D., the spectral correlation density (SCD) is introduced as a feature by which several signal modulations can be classified is introduced. The method by which the computational complexity of classifying with SCD is significantly reduced is proposed is then proposed in F.. Classifiers that will be used in this work are detailed in Section G.. The results of using feature selection techniques using multiple classifiers as compared to not using feature selection techniques is presented in Sections H. and I..

BACKGROUND

D. Classification of Cyclostationary Signals

D..1 Signal Model

Signal models used in MC vary in their level of detail, but a model which incorporates all of the important parameters that are discussed in this work are presented here. The noiseless complex baseband signal is commonly represented by (1) from [1]

$$r(t) = \sum_{n=1}^{N} e^{j2\pi\Delta f_n t} e^{j\theta_n} \alpha_n s_k g(t-(k-1)T-\epsilon_n T) \qquad 0 \le t \le KT \tag{1}$$

with Δf_n, θ_n, and ϵ_n, and α being the carrier frequency offset, the time-invariant carrier phase, and timing offset of the n^{th} signal with respect to the receiver's reference clock, and amplitude of the n^{th} signal respectively. Each of the equi-probable complex data symbols s_k have symbol period T for the modulation of order P. The channel response to the transmitted pulse shape p_{TX} is the convolution between $h(t)$ and p_{TX} and is denoted by $g(t)$. The nature of channel impulse response $h(t)$ and the p_{TX} is assumed to be such that frequency-flat, slow Rayleigh fading takes place.

Attempting to classify the modulation of the n^{th} signal can be broken down into three stages that each have its own challenges. First, a signal goes through a pre-processing stage in which common signal parameters such as the center frequency offset Δf_n or symbol rate can be estimated, channel effects such as multi-path fading can be compensated through equalization, and multiple signals are isolated using filtering techniques. This is done to maximize the effectiveness of the second

stage, the feature extraction stage, in which a set of features are extracted from the signal using a variety of algorithms. Choosing a set of features that will most effectively describe the incoming signal has been the primary subject of research in MC. Lastly the classification stage implements a decision structure to best differentiate the incoming signals based on the extracted features. A few common decision structures include a decision tree, an artificial neural network, or a support vector machine. These three stages are not always treated as disjoint, but details about the pre-processing stage often omitted or lacking in detail in work in this area. For this reason, much of the work in MC can be summarized while focusing on the feature extraction and classification stages.

Many signals and systems have been modelled as wide-sense stationary stochastic processes where second-order statistics of the signal remain constant with time, but whose autocorrelation is independent of time. However, many man-made signals exhibit a periodic or an almost-periodic autorrelation function because they contain various periodic structures in time. These signals are called *cyclostationary*. Over the years, cyclostationarity has been studied rigorously in continuous and discrete, real and complex, and stochastic and non-stochastic contexts [4]. The cyclic spectrum at cycle frequency α of the process $x(t)$ can be written

$$S_x^\alpha(f) = \int_{-\infty}^{\infty} R_x^\alpha(\tau) e^{-j2\pi f\tau} d\tau, \tag{2}$$

which can interpreted as the time-averaged statistical correlation of two spectral components separated by cycle frequency α as the bandwidth of each spectral component approaches zero. For this reason, the cyclic spectrum can also be called the spectral correlation density (SCD). According to this definition $S^0(f)$ is actually the traditional power spectral density (PSD) of the process $x(t)$. The SCD has been used for analysis and classification of signals in many areas and this primarily stems from its ability to detect and characterize the presence of cyclic features such as cyclic prefix length, symbol period, or carrier frequency even in the presence of noise and other channel effects. Peaks in the SCD describe correlation in time between each pair of frequencies in the PSD. Naturally, analysis of the SCD permits the identification and characterization of underlying periodicities of the signal. The SCD is also granted a level of noise immunity because there are no inherent periodicities in noise. Thus, one might think of noise as spreading energy across all possible periodicities evenly.

E. Estimation of the SCD

The SCD is readily derivable in closed form for many continuous forms of communications signals and a substantial efforts has been made decades ago to estimate this quantity for a finite-duration digital signal. For a digital signal, estimating the SCD is commonly estimated using the FFT Accumulation Method (FAM) and is calculated as follows:

$$X_{N'}(n,k) = \sum_{r=-N'/2}^{r=N'/2} a[r]x[n-r]e^{-j2\pi k(n-r)T_s} \tag{3}$$

$$S_x^\alpha(n,k) = \frac{1}{N}\sum_{n=0}^{N-1} \frac{1}{N'} X_{N'}\left(n, k+\frac{\alpha}{2}\right) X_{N'}^*\left(n, k-\frac{\alpha}{2}\right) \tag{4}$$

where $N^{'}$ and N together determine a resolution in both the time and frequency domains and $a[n]$ is an arbitrary windowing function. Equation (3) is the sliding window discrete Fourier transform (DFT) with window $a[n]$. For this work $N^{'} = 8$, $N = 512$, and $a[n]$ is a hamming window of length $N^{'}$.

The SCD itself contains enough information to determine particular properties of each signal, however it contains far too many points to be used directly for classification and the number of points can be significantly reduced using feature selection techniques talked about in Section F.. A common way of reducing the SCD to a more reasonable number of points for classification is to take the maximum along the α axis to create an $\alpha-$profile and is thus used as a basis of comparison for some of the techniques presented in this work.

F. Feature Selection Algorithms

The objective of feature selection algorithms are to reduce a larger set of features to a smaller subset by removing features that are determined to be either redundant or irrelevant. There are three major types of feature selection algorithms: wrappers, embedded methods, and filters. Wrappers use the performance of classifiers to score the importance of features in a feature space. Embedded methods are similar to wrapper methods in that they both utilize the classifier when ranking features, but they "incorporate[s] variable selection as part of the training process" [5]. Filter methods function differently than embedded and wrapper methods by determining the importance of a feature using a classifier-independent scoring function [5]. For this project, filter and wrapper methods are used.

F..1 Score-based Feature Selection

Score-based feature selection techniques are an example of filter feature selection algorithms. These algorithms reduce an initial feature space of M features to a set of K features (where $K < M$) by passing all M features to an appropriate scoring function and selecting the K best-scoring features. This scoring function is independent of the classifier being used on the features which makes this technique a filtering algorithm. There are various scoring functions that can be used, but for this project, the analysis of variance (ANOVA) and mutual information feature selection techniques were implemented. The ANOVA scoring function determines the importance of each feature by comparing 1) the variance of a set of features $\boldsymbol{X}$ within samples of a common label (group) and 2) the variance of the set of features $\boldsymbol{Y}$ across all groups. The ANOVA test takes the ratio of these two values to produce a score known as the F-value as shown below [6]:

$$F_{rat} = \frac{MS_{bn}}{MS_{wn}} = \frac{\sum(\boldsymbol{X} - \bar{\boldsymbol{X}})^2}{\sum(\boldsymbol{Y} - \bar{\boldsymbol{Y}})^2} \tag{5}$$

If F_{rat} is closer to 1 this means that the variances are similar implying that the groups are similar looking only at one feature. The farther from 1, the more likely that this feature can be used to diff For a more detailed discussion on such experiments and one-way ANOVA, the reader is encouraged to refer to [6].

Another scoring function that can be used is based on mutual information. Mutual information is

another statistical based on entropy that attempts to describe how much information is gained about one variable when another is known. The mutual information between several variables requires knowing joint probability density functions (PDFs) for each pair of variables and thus requires a substantial amount of samples. Instead, the joint PDF can be estimated using the provided samples with the k-nearest neighbors density estimation technique. An expression for this implementation of mutual information is provided in equation 6 [7]:

$$I(X,Y) \approx \phi(N) - \phi(m) + \log(V_{m;y}) - \phi(N_x) + \phi(k) - \log(V_{k;y|x}), \tag{6}$$

where X and Y are random variables, V is the volume, $\phi(n)$ is the digamma function, and k and m denote different neighbors.

F..2 Recursive Feature Elimination

Recursive feature elimination (RFE) is an example of a wrapper feature selection algorithm. RFE determines a feature's measure of importance by observing how the incorporation of certain features affects the classifier's performance. At each iteration of RFE, a certain number of features are removed. This new space is then used to train and test the classifier. Based on the classifier's performance on the testing data, a new set of features are removed from the feature space, and the procedure repeats until the desired number of features is obtained [8]. The advantage of this algorithm is that it weights the features based on performance rather than a statistical measure of redundancy. However, this algorithm has a large computational cost, as it requires a substantial amount of time to be completed for high-dimensional data sets.

G. Classification Algorithms

SVM: Support vector machines (SVM) are generally used to classify between two classes that are linearly separable. If classes are linearly separable, then a hyper-plane can be chosen such that points that have a negative distance to the hyper-plane are of one class and those that have a positive distance are the other [9]. Finding such an optimal hyper-plane involves solving an optimization problem using quadratic programming of which there are many tools available. Most data-sets are not linearly separable and often contain noise. So to counteract this, SVMs use a linear transformation with a chosen kernel to transform low-dimensional problem into a higher dimensional space where data is more likely separable.

Boosting: Boosting is an algorithm for training an ensemble of classifiers. Ensemble classifiers are used in widely in machine learning and is based on the principle that it is easier to generate weakly performing classifiers than creating one strong classifier. A boosting algorithm can be implemented in several ways, but in general, an ensemble is first trained on an initial data-set. Then, samples that are classified incorrectly are given heavier penalties to the training algorithm in hopes that the next time it is trained, these samples will be correctly classified. This algorithm is repeated iteratively until a chosen condition (perhaps performance related) is met. [10]

Bagging: Bagging stands for **b**ootstrap **agg**regat**ing**. Bagging is also an algorithm for training ensemble classifiers. It is known that an ensemble classifier performs well when the constituent classfiers are diverse in terms of their classifications. Bagging aims to create a diverse set of classifiers by training the constituent classifiers on different subsets of the original training set.[10]

Multi-layer Perception: The multi-layer perceptron (MLP) is essentially a feed-forward neural network. The MLP is a machine learning technique that attempts to model the human brain to make classification decisions. Its structure consists of connections between neurons, with each connection having an assigned weight. A standard neural network contains an input, output, and hidden layer. When training the MLP, the objective is to update the weights of each connection after calculating the error. A common method of training the MLP is called the back-propagation algorithm, which provides a method of updating the weights by estimating the local gradient of a neuron, which could be accomplished using stochastic gradient descent (SGD).

METHODOLOGY

The SCD calculation presented in section E. were implemented in Python and two sets of features were produced. The first set of features was the α-profile and the second was the 'flattened' featured. The 'flattened' features were produced by estimating the SCD and flattening the result (a matrix) into a one-dimensional vector. This could also be called the full SCD. Eight different modulations were generated on a Universal Software Radio Peripheral (USRP) N210 using GNU Radio, transmitted over the air and captured using GNU Radio on another USRP. The noise level in this process was kept as low as possible because the purpose was to emulate a signal being received by a non-cooperative receiver. Hence, the signal being received was not synchronized in frequency or time. The 8 modulations were BPSK, QPSK, 16-QAM, CPFSK, GMSK, OFDM, FM, and AM. Complex circularly symmetric additive white Gaussian noise was then added to each signal to simulate an AWGN channel with signal to noise ratios (SNRs) varying from -5 to 15 dB. This SNR was calculated by taking the ratio of the power in the signal to the power in the noise on a decibel scale. Note that this SNR is not equivalent to the SNR seen by a cooperative receiver because the non-cooperative receiver receives twice the necessary bandwidth to receive the signal. Hence, a cooperative receiver would see half as much noise as the non-cooperative receiver which is effectively 6 dB higher in SNR than the non-cooperative receiver. Receiving twice as much bandwidth as necessary from a classification standpoint will unlikely have an effect as the SCD measures the correlation of a pair of frequencies.

One-thousand independent samples were taken from each of the 8 modulations to be used for classification. Half of these samples are to be used for training classifiers and the other half are to be used for testing the classifiers. The α-profle and full SCD sets were separately reduced using feature selection algorithms discussed in F.. Implementations of these feature selection algorithms were obtained from the scikit-learn library, which hosts a wide array of machine learning algorithms programmed in python. The *SelectKBest* function was used as an implementation of the score-based feature selection. The ANOVA and mutual information scoring functions were implemented by calling *f_classif* and *mutual_info_classif* respectively, with each function finding their aforementioned score between features and labels. Each feature selection algorithm was used to reduce the set of features used for classification to 105 features based on the training set.

Once the training phase was complete, the classifiers were then given the testing data, and the percent errors were recorded for each configuration (i.e. feature selection algorithm and classi-

fier). This procedure was repeated 10 times with the average error calculated over all 10 iterations. Scikit-learn's implementation of the four classifiers described in section G. were used. (**Note:** Scikit-learn's RFE implementation did not support Neural Networks and Bagging. Results obtained from RFE are only provided from using SVMs and Bagging) Their own specific set of parameters were used are as follows [8]:

SVM: The one vs one algorithm was selected for multi-class classification. A linear kernel was also chosen for all experiments.

Bagging: One hundred decision trees were chosen as the base classifiers. *max_features* was set to 0.5.

Boosting: One hundred decision trees were similarly chosen as the base classifiers. The learning rate was set to 0.5.

Multi-layer Perceptron: This consisted of 2 hidden layers (each with 100 neurons), using a rectified linear unit function as the activation function and a stochastic gradient-based optimizer to estimate the local gradient.

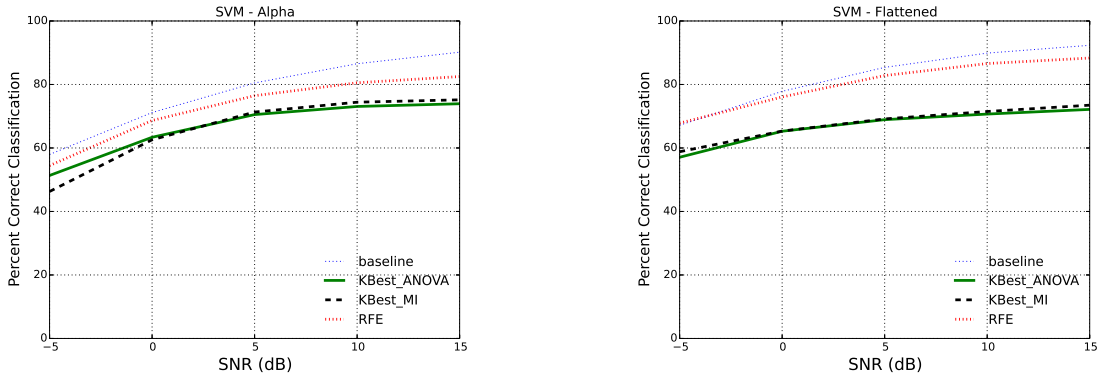
NUMERICAL RESULTS

H. Results by Feature Selection Algorithm

The results for percent correction classification for using an SVM while using different feature selection algorithms for both the flattened set of features and the α-profile are shown in 1. Each of the three algorithms shown in this Figure 1 reduced the feature set to 105 features. The 'baseline' method does not use any feature selection method and serves as a basis of comparison for the feature selection methods. Looking at both Figure 1b and Figure 1a, the 'baseline' algorithm generally performs the best as it uses all available features to classify each of the modulations. The 'RFE' method of feature selection has the best performance among the three feature selection methods and performs close to the baseline performance except for higher SNRs. At 15 dB SNR, the 'RFE' has a decrease of 7% percent correct classification. This is a meager reduction in terms of correct classification considering that less than 10% of the original features remain when 105 features are kept in the case of the α-profile and less than 3% of the features for the flattened features. This supports the idea that proposed approach to classification can be made real-time by reducing the feature set significantly.

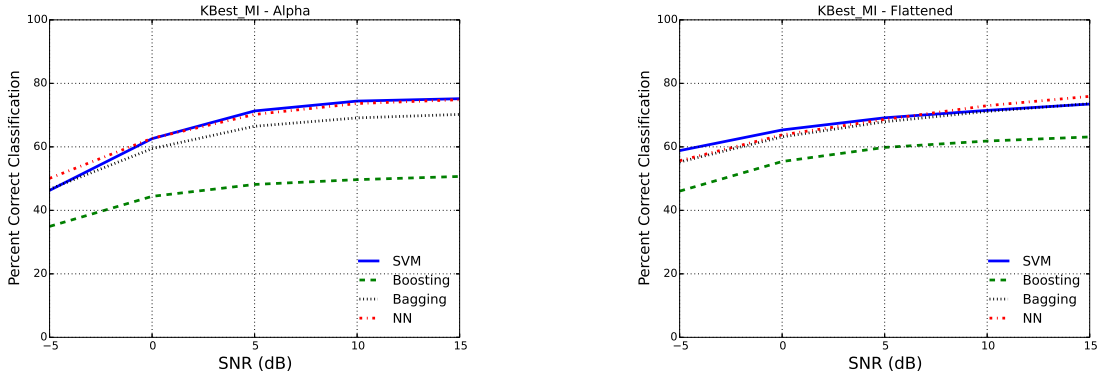
I. Results by Classifier

The results for percent correction classification for using the mutual information based feature selection and different classifiers for both the flattened set of features and the α-profile are shown in 2. The number of features in Figure 2 are the same as the previous section 105 features. All classifiers appear to preform roughly the same using this feature selection algorithm except for Boosting. We suspect this is because Boosting used a decision tree classifier as its base classifier which is subject to over-fitting. What is interesting to note here is that the flattened features generally perform better than the α-profile for low SNRs At -5 dB SNR, the SVM classifier has an accuracy of 45% for the α-profile and 60% for the flattened features. This suggests that the flattened features are a better set of features to consider for future work than just the α-profile.



(a) (b)

Figure 1: Percent Correct Classification using SVM and α-profile **1a**) and flattened features **1b**)



(a) (b)

Figure 2: Percent Correct Classification using mutual information based feature selection for α-profile **2a**) and flattened features **2b**)

CONCLUSIONS

In Cognitive Radio, signal classification is an important task in enabling higher levels of awareness. Modulation classification in non-cooperative scenarios requires that no *a-priori* information is used in the process of capturing a signal and classifying it from others. Until now, many modulation classification techniques have used features that require system synchronization to classify and hence assume some level of *a-priori* knowledge. The spectral correlation density (SCD) is a feature that can be used to classify signals without the need for system synchronization, but it generally too computationally complex for real-time use. Using feature selection techniques that have developed thoroughly in the machine learning community, only a small portion of the SCD needed to be calculated to properly classify 8 different modulations captured over the air in real time. Results show this technique is a promising baseline technique for classifying signals in non-cooperative scenarios with SNR ranges from -5 dB to 15 dB.

ACKNOWLEDGEMENTS

This project was partially supported by the Broadband Wireless Access and Applications Center (BWAC); NSF Award No. 1265960.

REFERENCES

[1] O. a. Dobre and R. Inkol, "Blind signal identification: Achievements, trends, and challenges," *2012 9th International Conference on Communications (COMM)*, pp. 349–352, jun 2012.

[2] F. Hameed, "On the Likelihood-Based Approach to Modulation Classification," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 5884–5892, 2009.

[3] A. Hazza and M. Shoaib, "An Overview of Feature-Based Methods for Digital Modulation Classification," *Conference on Communications, Signal Processing, and their Applications*, vol. 1, no. 08, 2013.

[4] W. A. Gardner, "Cyclostationarity: Half a century of research," *Signal Processing*, vol. 86, no. 4, pp. 639–697, 2006.

[5] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research (JMLR)*, vol. 3, no. 3, pp. 1157–1182, 2003.

[6] G. Heiman, *Basic Statistics for the Behavioral Sciences*. Cengage Learning, 2010.

[7] B. C. Ross, "Mutual information between discrete and continuous data sets," *PLoS ONE*, vol. 9, no. 2, 2014.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[9] E. Alpaydin, *Introduction to Machine Learning*. The MIT Press, 2nd ed., 2010.

[10] R. Polikar, "Ensemble based systems in decision making," *Circuits and Systems Magazine, IEEE*, vol. 6, no. 3, pp. 21–45, 2006.