**Predicting Real-Time Spectrum Interference Using Support Vector Machine**

|  |  |  |
| --- | --- | --- |
| *Hubert Kyeremateng-Boateng*  *Department of Computer Science*  *Bowie State University*  *Bowie, MD USA*  *kyerematengboh0404@students.bowiestate.edu* | *Marvin Conn*  *Department of Computer Science*  *Bowie State University*  *Bowie, MD USA*  *marvinconn@gmail.com* | *Darsana Josyula, Ph.D.*  *Department of Computer Science*  *Bowie State University*  *Bowie, MD USA*  *darsana@cs.umd.edu* |

***Abstract -*** As more people and organization use wireless networking for communication and business, spectrum availability will become a scarce resource commodity that will not readily available for both licensed and unlicensed users. The ability to detect or predict when a spectrum is available or not available for use is of great importance and that is the purpose of this research. In this work, we validate a previous research conducted by Jacob A. Kovarskiya et al in the paper *“Predictive Energy Detection for Inferring Radio Frequency Activity”* using Linear Support Vector Regression (LinearSVR), and then we improve upon it. Radio frequency (RF) energy level is used to detect whether the spectrum is busy or idle for transmission. We calculate the energy level of the simulated RF signal using two methods: average of the RF or the Cumulant (first-order). As part of this research we compare both methods to determine which one trains and predicts faster and accurately.

**Keywords - Radio Frequency, Cumulants, Signal interference, Support Vector Regression, Linear SVR**

# Introduction

As more people continue to use wireless electronic devices, there will come a time where radio frequency will be a scarce resource. Both licensed and non licensed user’s will be scrambling for radio frequency bandwidth. The ability to detect and predict accuractly Cognitive Radio (CR) is a system which senses its electromagnetic environment and dynamically adjusts its radio parameters to improve performance [5]. With CR and radio frequency detection, network providers can detect or predict which radio frequency bandwidth is currently in use and which bandwidth is not in use. Network providers can also prevent congestion in bandwidth by detection if a specific bandwidth is in use before another user attempts to connect to the same bandwidth.

In order to prevent spectrum bandwidth inference and congestion, there is the need to accurately predict when a bandwidth is available or not for use. This research work seeks to validate the one-step look ahead method proposed by Jacob et al [1] by using Python’s Sklearn Support Vector Regression (SVR) module.

This paper first provides an overview of previous work by Jacob A. Kovarskiya et al and then discuss our research. At the end of the paper, we compare the results of using Python Sklearn and MATLAB SVM. We also compare the error rate of predicting the presence or absence of a radio frequency using the average power level or Cumulant (first-order). We center our comparison along the following dimensions: training speed, prediction speed and prediction accuracy. We then discuss how these two machine learning algorithms and methods of generating the energy levels compare with the original research.

# Overview

Kovarskiya’s original research work calculates the average energy level of the spectrum in order to predict the idle and busy state of the spectrum bandwidth [1]. The time series of energy levels of the training data is used to train the Matlab SVR model and the trained model is then used to predict future spectrum occupancy state.

Kovarskiya’s work uses a one-step look ahead methodology to predict the RF signal down (idle) and up (busy) state. One-step look ahead prediction uses past predicted data to predict the next state of the spectrum. To use the one-step methodology, the entire test data is given as input to the previously trained Matlab SVR model. Once the Matlab SVR model has predicted the initial prediction using the test data, the original test data less the first sample is concatenated with the new prediction to create the new test data which is then used to get a new prediction from the SVR model. This method has shown to provide better results than the initial predicted results using the original test data.

To be able to make comparison between the Python and MATLAB codes, this work converts the Jacob’s et al code (MATLAB) to Python. At the last section, we compare the accuracy performance of both MATLAB and Python code if similar results were achieved with the Python cade as in the MATLAB code.

# SIMULATING RF DATA

## Generating Data

Radio frequencyignals at constant intervals were generated using a fixed sample size. The idle and busy state of the spectrum was simulated using probability values ranging between 0 and 1 for each interval [1].

Addictive white Gaussian noise was added to the generated signal to simulate RF signals. A noise floor variance of -100 dBm and a power level of -40 dBm.

Below is a sample of a generated RF signal with a sample size of 500(Ns)

A screenshot of a cell phone

Description automatically generated

Figure 1 – Moving Average Input signal

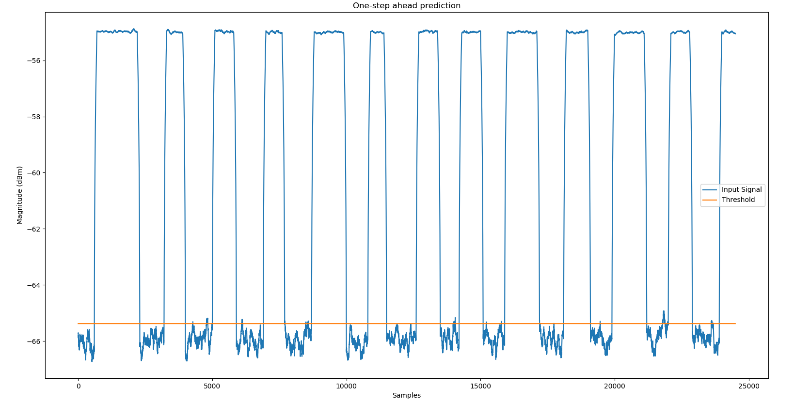


Figure 2 – Cumulants (First-order) Input signal

## Methodology – Energy Detection

In detecting whether a signal is in its idle/busy state, energy detector is used to determine the current state of a signal. Energy detection is one of the most popular sensing methods, it detects the primary user’s(PU) activity based on the energy signal received.

Energy detection computes the energy of the received N samples as the squared magnitude of the Fast Fourier Transform (FFT) of these samples averaged over N samples using the following formula [13]:

Where is the observed RF signal .

The energy is compared with a computed threshold to obtain the sensing decision [1] as follows:

The detection performance of the algorithm can be evaluated through the probability of detection and the probability of false alarm . The probability of detection refers to the numbers of correct detections (PU is present) over the total number of sensing trials (test data) while the probability of false alarm refers to the number of times that the PU is falsely detected over the total number of trials (test data) [13]. The probabilities are as follows:

Where represents the presence of a user or busy signal state and represents the absence of the primary user or idle state of the signal.

The desired probability of false alarm determines the decision threshold which does not require signal power [1].

## MOMENTS AND CUMULANTS

Another methodology used in this research was instead of using the average total power from the generated radio frequencies, the cumulants to the first other of the generated radio frequencies.

Cumulants is defined as the formal relation between the coefficients in the Taylor expansion of one function M(ξ) with M(0) = 1, and the coefficients in the Taylor expansion of log M(ξ)[4].

The moment of a 1-dimensional (d=1) of a random variable X is defined as M1 = (x), M2 = (x2),.,.,.,Mn = (xn) and in a multi-dimensional setting ( d > 1) the moments becomes a tensor[5].

The rth moment of a real-world value random variable X with density f(x) is

for integer r= 0, 1,.... The value is assumed to be finite[4].

Cumulants are expressed in terms of the moments by equating Taylor expansion coefficients. The cumulants κr are the coefficients in the Taylor expansion of the cumulant generating function about the origin [4].

The cumulants can be expressed in terms of the moments by equating Taylor expansion coefficients [5]. For d = 1, we find:

|  |  |
| --- | --- |
|  |  |
|  | ’ |
|  |  |
|  |  |

[5]

# Support Vector Regression

[1]

Support vector machine (SVM) analysis is a popular machine learning tool for classification and regression, first identified by Vladimir Vapnik and his colleagues in 1992. SVM regression is considered a nonparametric technique because it relies on kernel functions [11].

Support Vector Regression (SVR) machine learning algorithm is used to develop a model which used to predict the radio frequency up and down time. SVR which is an extension of Support Vector Machine is a supervised learning model that construct an optimal hyperplane in an N-dimensional space through margin maximization [1].

Skearn library’s Linear SVR implementing the *liblinear* kernel is used in this research because is it scalable compared to the SVR which implements *libsvm* kernel [2]. Also Linear SVR is more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples [2].

The linear function that describe the hyperplane in SVR is given by:

where *w* is the training data vector and *x* is the weight and *b* is the bias term [1].

In margin maximization, we refer to finding the optimal margin for which the closet data point and the hyperplane is at its maximum [1].

# RESULTS

We complete this paper by comparing the accuracy performance of both MATLAB and Python code. The parameters are used in our testing with the exception of the sample size. In our testing we use a sample of 1000 (N*s*)

Apart from comparing the results of our Python code with the original work, we also performs tests with Cumulants(first-order). This is to provide other solutions to effectively predicting and detection RF signals. Below are the test results that was from

### TOTAL AVERAGE POWER

A picture containing sky

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

# REFERENCES

[1] Jacob A. Kovarskiy, Anthony F. Martone, Kyle A. Gallagher, Kelly D. Sherbondy, Ram M. Narayanan, "*Predictive energy detection for inferring radio frequency activity,*" Proc. SPIE 10633, Radar Sensor Technology XXII, 1063318 (4 May 2018);

[2] Scikit-learn.org. (2019). *1.4. Support Vector Machines — scikit-learn 0.21.2 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/svm.html#svm-regression [Accessed 1 Jul. 2019].

[3] Scikit-learn.org. (2019). ‘sklearn.preprocessing.StandardScaler — scikit-learn 0.21.2 documentation’. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html [Accessed 12 Jun. 2019].

[4] A. Geron, *Hands-On Machine Learning with Scikit-Learn & Tensorflow*, 1st ed. O'Reilly, 2017, pp. 147,156-166.

[5] Zitouni, R., George, L. and Abouda, Y. (2005). *A Dynamic Spectrum Access on SDR for IEEE 802.15.4 networks*.

[6] "NVIDIA cuDNN", *NVIDIA Developer*, 2019. [Online]. Available: https://developer.nvidia.com/cudnn. [Accessed: 11- Jul- 2019].

[7] "Fit a support vector machine regression model - MATLAB fitrsvm", *Mathworks.com*. [Online]. Available: https://www.mathworks.com/help/stats/fitrsvm.html. [Accessed: 07- Aug- 2019].

[8] "Understanding Support Vector Machine Regression- MATLAB & Simulink", *Mathworks.com*. [Online]. Available: https://www.mathworks.com/help/stats/understanding-support-vector-machine-regression.html. [Accessed: 07- Aug- 2019].

[9] Sanaullah, M. (2013). *A Review of Higher Order Statistics and Spectra in Communication System*. Global Journal of Science Frontier Research.

[10] Mendel, Jerry. (1991). Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications. Proceedings of the IEEE. 49. 278 - 305. 10.1109/5.75086.

[11] ZHOU. (2010). Signal Classification Method Based on Support Vector Machine and High-Order Cumulants. Wireless Sensor Network. 2. 10.4236/wsn.2010.21007.

[12] Agarwal, Anirudh & Dubey, Shivangi & Asif Khan, Mohammad & Gangopadhyay, Ranjan & Debnath, Soumitra. (2016). Learning based primary user activity prediction in cognitive radio networks for efficient dynamic spectrum access. 1-5. 10.1109/SPCOM.2016.7746632.

[13] Y. Arjoune, Z. E. Mrabet, H. E. Ghazi and A. Tamtaoui, "Spectrum sensing: Enhanced energy detection technique based on noise measurement," *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, 2018, pp. 828-834. 10.1109/CCWC.2018.8301619.