Performance Analysis of Machine Learning based Spectrum Sensing Methods for Cognitive Radio

Arjun Satheesh, Gautam Govindaraju, Prajakta Gurav, Priyanshu Bhagat and Ranjan Bala Jain

Department of Electronics and Telecommunication Engineering, Vivekanand Education Society's Institute of Technology, University of Mumbai, India.

2019arjun.satheesh@ves.ac.in, 2019gautam.govindaraju@ves.ac.in, 2019prajakta.gurav@ves.ac.in, 2019priyanshu.bhagat@ves.ac.in, ranjanbala.jain@ves.ac.in

Abstract— **Cognitive radio is treated as a favorable solution to address the frequency shortage issue in 5G. It consists of spectrum sensing unit that can effectively identify which communication channels are currently occupied and which ones are available for further usage through intelligent means. Spectrum sensing is the process of periodically monitoring frequency bands. Traditional methods of Spectrum sensing are complex, require some prior knowledge about the primary user signal and perform poorly under low signal-to-noise ratio (SNR), Hence, there is a need for simple and accurate method of spectrum sensing. Reliability of spectrum sensing method under all SNR conditions is a research issue. In this work, supervised machine learning algorithms, namely Decision Tree, Support Vector Machine and K- Nearest Neighbour are employed for spectrum sensing with a large data set for various modulation schemes. ML based solution is introduced since it improves the probability of determining the availability of channel better as compared to the traditional methods. The performance of all these algorithms is measured in terms of accuracy, probability of detection and probability of false alarm for large range of SNR and based on these, most suitable algorithm for spectrum sensing is determined for a particular usage.**

Keywords— Machine learning, Spectrum sensing, K-Nearest Neighbour, Support Vector Machine, Probability of detection, Probability of false alarm, Primary user, Secondary user, Cognitive Radio, 5G networks

I. INTRODUCTION

In recent years, technological advances and the emergence of wireless communication devices like GPS, Wi-Fi, mobile phones, and cordless telephones have increased the demand for radio spectrum. [1]. There is a lack of uniformity in the use of the frequency spectrum. The usage of the frequency spectrum depends on some factors like, time and geographical location. There may be some frequencies partially occupied at a given time while others are overcrowded. Frequencies that have been left unused by users are called white spaces or spectrum holes [1, 2]. RF spectrum is a scarce resource, so to make efficient use of it at a specific location or time, the concept of Cognitive Radio (CR) has been proposed [3]. The concept of CR refers to an intelligent device that has the ability to detect unused portions of the spectrum, commonly known as spectrum holes, and subsequently allocate these available resources to users in need. [1]. IN CR, Primary users (PU) are those users who use licensed spectrum, and Secondary users (SU) are those who are not licensed users but want to access the spectrum when they are in need for a specific period of time when the PU is absent. This allocation of spectrum to SU can increase the spectrum utilization without getting it wasted.

Fig. 1 illustrates the flow of CR cycle. In CR user's task involves monitoring and capturing spectrum band information to identify available spectrum opportunities, also known as spectrum holes, which are then estimated using Spectrum sensing (SS). The appropriate frequency band is chosen based on the user requirements. After determining the operating spectrum band, communication can be carried out over this band.

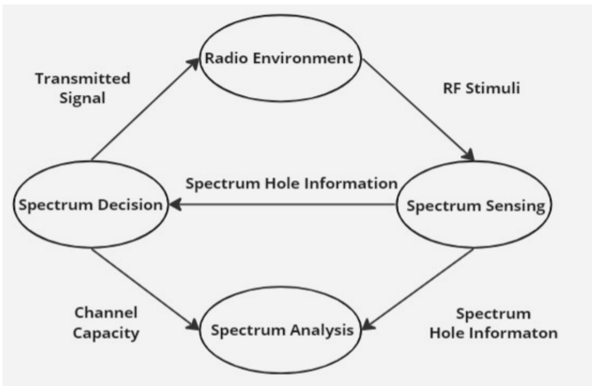

Fig. 1: CR cycle

CR's primary function is spectrum sensing, the objective of which is to identify the band state, i.e., whether a spectrum is available or occupied. There are many SS techniques available, each having its unique ways to sense the spectrum. Some SS techniques work without using ML as others use ML algorithms. Some of the Spectrum sensing methods which do not use ML algorithms are Energy Detection, Matched Filter and Cyclostationary feature detection [4, 5, 6 and 7]. The energy detection technique for spectrum sensing involves measuring energy or power of the received signal and comparing it to a pre-determined threshold to determine the presence or absence of a signal of interest in the spectrum [4].

The matched filter technique for SS involves correlating an incoming signal with a known reference signal to detect the presence of the reference signal in the received signal [5]. The cyclo-stationary feature detection technique for spectrum sensing involves analysing the cyclic properties of the received signal to identify unique patterns or features that indicate the presence of a signal of interest in the spectrum [6]. It is observed that Energy detection is a widely used technique for SS due to its low complexity and simplicity. However, it is found out that it is not an accurate method of SS, as it requires some prior knowledge about the PU signal and the sensing time is relatively high. Both Matched filter and Cyclostationary feature detection are popular SS methods. However, they are complex and sensitive to signal variations. To solve these problems, ML algorithms are adopted as they are more efficient as compared to non-ML spectrum sensing techniques. Various machine learning algorithms are cited in literature for SS. They are Support Vector Machine (SVM), Decision Tree (DT) and K- Nearest Neighbours (KNN). These algorithms have emerged as successful algorithms for PU detection because of their high discriminative approach and pattern recognition capabilities.

In [1] all these algorithms are employed for spectrum sensing with only 1600 samples which are very less for ML models. In [8], data set is generated using Arduino Uno and 433 MHz wireless transmitter with only ASK and FSK modulation techniques. Here SVM and KNN algorithms are proposed to be better in terms of better training accuracy for spectrum sensing. In [9], SVM based classifier is evaluated for spectrum sensing with various types of Kernels and it is found that non-linear kernels perform better with small samples. In [10], novel co-operative spectrum sensing algorithms like K-means clustering, Gaussian mixture model, SVM and weighted KNN are implemented for Co-operative Spectrum Sensing using energy as feature vector. The performance is evaluated in terms of average training time, sample classification delay and receiver operating characteristic (ROC) curve. In [11], random forest, SVM with different kernels, decision tree, Naïve Bayes, K-nearest neighbours, and logistic regression algorithms are employed for spectrum sensing and random forest model is proposed to be the best among all. In [12], the authors proposed a machine learning-based reliable spectrum sensing algorithm in which the FC uses a weight-based decision combination rule. This scheme uses the quantized information, where the local decisions of all CR users are combined to make a global decision, taking into consideration the reliability of each CR user. However, in all these related works, which algorithm will be suitable under low Signal to Noise Ratio (SNR) condition is not clearly described.

This work aims to investigate the use of ML methods for SS in cognitive radio on basis of large range of SNR conditions. SNR which stands for signal-to-noise ratio is the measure of actual signals strength with respect to the undesired signal. SS is carried out using three well-known ML algorithms namely DT, SVM and KNN. Accuracy of each algorithm is evaluated for certain parameters, such as depth of tree for DT, type of Kernel for SVM and value of K for KNN algorithms. The values of parameters are chosen for maximum accuracy. The performance of each algorithm is evaluated in terms of Accuracy, Probability of Detection (P_d), and Probability of False Alarm (P_{fa}) for various SNRs. As per our knowledge performance analysis of various ML models for SS is not available in the literature under various SNR conditions with optimum parameters, large data sets and various modulation schemes. As wireless channels are notorious in nature, the objective is to identify the most suitable algorithm that can provide accurate and reliable SS in CR networks under all SNR conditions.

II. SYSTEM MODEL

SS based on ML involves data acquisition, data cleaning, training the models and prediction based on test data as illustrated in Fig. 2. The ML models employed for spectrum sensing are - KNN, SVM, and DT for CR networks.

For analysis and training the ML models, appropriate dataset is needed. Data acquisition and data cleaning are essential steps in ensuring the quality of data for training the models. For this proposed work, dataset is obtained from [13]. There is a challenge in determining the presence or absence of signals as it is not mentioned in the dataset. To address this issue, SNR and amplitude threshold levels are set for different modulation schemes to distinguish between signal and noise among the data.

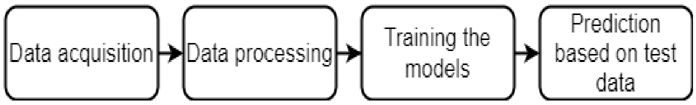

Fig. 2: Spectrum sensing using ML methods for CR

A. *Dataset Acquisition*

For the purpose of training the model, the dataset HISARMOD: A new challenging modulated signals is acquired from [13]. It consists of three attributes of data containing the I/Q signal information, Signal-to-Noise Ratio (SNR) and modulation type of all signals. It includes 26 modulation types from 5 different modulation families, namely analog, FSK, PSK, QAM and PAM. There are 20 different SNR levels ranging from -20dB to 18dB. In this dataset, it is mentioned that while generating signals for dataset, a raised cosine pulse shaping filter is employed with roll-off factor of 0.35 and oversampling rate of 2.

B. *Data Processing*

In the dataset, information about the presence or absence of the primary user is not mentioned, but from the I/Q information, the amplitude of the signal is calculated as follows,

$$A = \sqrt{I^2 + Q^2}$$

where,

A is the amplitude of the signal,
I is the In-phase parameter of the signal,
Q is the Quadrature phase parameter of the signal.

For each of the 26 modulation types, different threshold values of SNR and amplitude have been considered to classify the channels whether a user is present or not. Table 1 shows the various threshold levels for SNR and amplitude for different modulation types that is set to determine presence or absence of signal[14, 15].

C. Training ML models

There are many ML algorithms which could be addressed for SS. In order to predict the presence of user in a channel, three different ML methods have been employed to identify vacant spectrum. They are DT, KNN and SVM. In the following subsections each ML method is described in detail.

i) Decision Tree

It is a supervised machine learning technique that is one of the most popular ML algorithms and it may be used for both classification and regression. It is a structured classifier where at each node, based on certain conditions, splits are being done for the purpose of classification as illustrated in Fig.3. It has two types of nodes, decision node and leaf node. Decision nodes are used to make decisions based on the specified conditions while leaf nodes are the last nodes of the decision tree that shows the output. DTs are easy to understand as they make decisions similar to human psychology.

Table 1: SNR and amplitude values for various modulation types

Modulation Type	SNR	Amplitude
BPSK	3.8dB	-
QPSK	6.6dB	-
8PSK	9.4dB	-
16PSK	12.1dB	-
32PSK	14.9dB	-
64PSK	17.7dB	-
4QAM	6.0dB	0.89
8QAM	8.4dB	0.71
16QAM	11.1dB	0.56
32QAM	13.9dB	0.44
64QAM	16.7dB	0.35
128QAM	19.5dB	0.28
256QAM	22.3dB	0.22
2FSK	2.9dB	-
4FSK	5.3dB	-
8FSK	8.1dB	-
16FSK	10.8dB	-
4PAM	3.3dB	0.97
8PAM	6.0dB	0.89
16PAM	8.8dB	0.71
AM-DSB	-0.4dB	1.41
AM-DSB-SC	3.3dB	0.97
AM-USB	-2.2dB	1.78
AM-LSB	-2.2dB	1.78
FM	14.9dB	-
PM	12.7dB	-

The main elements of DT are,

- **Root Node:** It is the start node of DT. Here, the entire dataset is split into two or more sets.
- **Splitting:** The process of dividing the nodes into sub-nodes based on conditions is known as splitting.
- **Branch:** It is a tree formed after splitting.
- **Pruning:** The process of removing unwanted branches from the tree is known as pruning.
- **Parent and Child Node:** Root node is a parent node and all other nodes are child nodes.
- **Leaf Node:** They are the last nodes of DT that shows the final output.

For selecting the best attribute in a decision node, different Attribute Selection Measures (ASMs) are available. They are, Information Gain and Gini index, which are defined in following subsections,

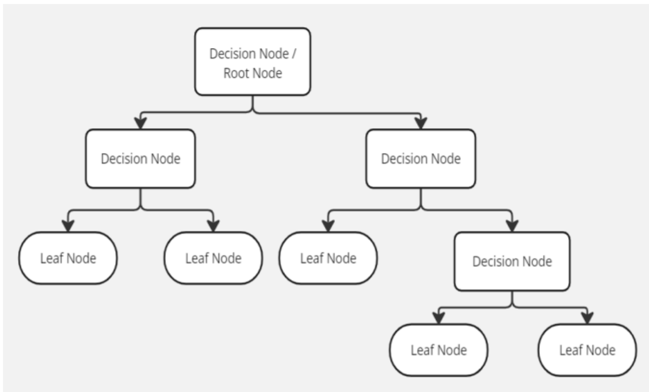

Fig. 3: Various nodes of Decision Tree

Information Gain: It shows how much information an attribute gives. It is measured as the change in entropy after splitting the node. The Information Gain (G) is given as,

$$G = S - [W_{avg} \times S_f]$$

where,

S is Entropy of the parent node,

W_{avg} is Weighted sum of entropy of each child node,

S_f is Entropy of each child node.

Entropy is used to measure the impurity of a given attribute. It is given as,

$$T = -P_{yes} \times \log_2(P_{yes}) - P_{no} \times \log_2(P_{no})$$

where,

T is Total number of the samples,

P_{yes} is Probability of yes at a particular decision node

P_{no} is Probability of no at a particular decision node.

Gini Index: It measures the impurity of an attribute while creating a split in a node. An attribute with low Gini index should be preferred above high one. If the Gini index is 0.5, it means that the elements are split equally among the nodes. Gini index is calculated as follows,

$$\mathrm{Gini} = 1 - \sum_{i=1}^{n} (p_i)^2$$

where, p_i is the probability of an object belonging to a particular class.

Thus, it can be said that this algorithm is very simple to understand and can be useful for solving decision related problems. Moreover, less data cleaning is required compared to other algorithms. However, presence of lot of layers makes it complex. Furthermore, for more class labels, there is an increase in computational complexity of DT.

ii) K-Nearest Neighbour

One of the simplest ML algorithms is KNN, which is based on supervised machine learning techniques. The technique places the new case/new data in a class that is closest and most identical to the existent classes based on the assumption that the new case/new data are identical to the cases that are already available. All data is stored by the KNN algorithm, which also ranks new data points according to similarity. This means that using KNN, new data can be easily categorised into the class which is most identical to it.

Consider the example illustrated in Fig.4,

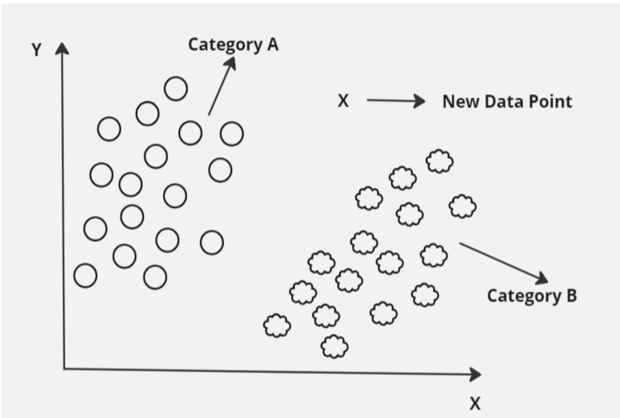

Fig. 4: KNN sample showing Clusters

In the given scenario, the objective is to classify new set of data points 'X' into appropriate category. Firstly, the number of neighbours ('K') are to be determined. There is no particular way to determine the best value for 'K', so we need to try some of the values to find out the best of them. A very low value for 'K' such as K=1 or K=2, can be noisy and can lead to the over-fitting problem. Large values for 'K' are good, but too large value of 'K' can lead to under-fitting problem. It could make the model inefficient. One of the ways to set the value of 'K' can be found using, $K = \sqrt{n}$, where, n is the total number of samples. It helps in finding the optimal value of 'K' for a given number of training samples [16].

Following this, the next step involves computing the Euclidean distance between the new data point and the existing data points. The Euclidean distance, which we have previously learned in geometry, represents the distance between two points and can be calculated using the following formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2},$$

where, d is the Euclidean distance,
(x_1, y_1) is the coordinate of the first point,
(x_2, y_2) is the coordinate of the second point.

By calculating the Euclidean distance, the nearest neighbours are determined. New data will be considered in the category with the highest nearest neighbours. Thus, with the help of KNN, the category or class of a particular dataset can be easily identified. In conclusion, the KNN algorithm is a versatile and widely used method in machine learning for classification tasks. Its simplicity and intuitive concept of finding the 'K' closest neighbours make it applicable in various domains such as spam detection, image recognition, and predicting values for classification problems, making it a valuable tool in many real-world applications.

iii) Support Vector Machine

SVM is one of the most well-liked ML algorithms which is mainly used to solve classification and regression problems. However, it is primarily employed in ML Classification issues. SVM generates a decision boundary which divides N-dimensional space into classes, allowing it to quickly classify new data points in the future. The SVM algorithm selects the most extreme vectors and points to aid in constructing the hyperplane. These extreme cases are represented by support vectors and serve as the foundation for the SVM algorithm. The two types of SVM are:

Linear SVM: A Linear SVM classifier is employed when dealing with data that can be distinguished into two classes utilizing a single straight line, thus qualifying it as linearly separable. This form of SVM is aptly named as Linear SVM.

Non-linear SVM: Non-linear SVM is utilized when dealing with non-linearly separable data. If a dataset cannot be distinguished by means of a straight line, it is deemed non-linear data. To address this scenario, Non-linear SVM classifier is implemented.

The main elements of SVM are,

- **Hyperplane:** The hyperplane is the best decision boundary that helps in classifying data points in n-dimensional space. There may be multiple lines or decision boundaries for segregating classes, but the goal is to find the best boundary.
- **Support vectors:** Support vectors refer to specific data points or vectors that have the closest proximity to the hyperplane and, as a result, play a critical role in determining its position. These vectors act as pillars to support the hyperplane and are consequently known as support vectors.
- **Margin:** The margin is the distance between the hyperplane and the support vectors. The SVM algorithm determines the line from each class that is closest to the other, with these points being referred to as support vectors.

The main objective of SVM is to maximize the margin. The hyperplane that possesses the largest margin is considered the optimal hyperplane. In summary, Support

Vector Machine (SVM) is a powerful and widely used machine learning algorithm known for their ability to handle both linear and nonlinear classification tasks. With their ability to find optimal decision boundaries and handle high-dimensional data, SVMs have found applications in areas such as image recognition and text classification.

The dataset HISARMOD: A new challenging modulated signals is acquired from [13] has 3,00,000 samples in it. Out of these 2,00,000 samples are used for training the models. The models are trained on the basis of three attributes, which are namely amplitude, SNR and modulation type.

Training the DT model: In order to train DT model, classifier DT is used. In this, the root node is chosen as SNR because the SNR attribute has the least Gini index value and the decision node is chosen which has a low value of Gini index.

Training the KNN model: In KNN model, finding out the appropriate value of K is an integral part. This is because if we choose a lower value of K, then the model will be over-fitted while on the other hand, if its value is high then the model may be under-fitted. In our model we choose the value of 'K' as 447 as per formula $K = \sqrt{n}$, [16].

Training the SVM model: In this model, three kernels, namely linear, Radial Basis Function (RBF) and polynomial kernels are employed. Among these, RBF and polynomial kernels have low efficiency of detection. As a result, the linear kernel is finalized to be used as the kernel in SVM.
After training the models, their performance is evaluated by testing them on the remaining 100,000 samples using different evaluation parameters,

- **Accuracy(A):** It is the ratio of number of times primary user is correctly detected to the total number of samples.
- **Probability of detection (P_d):** It measures how many samples are predicted as present out of all samples that are actually present.
- **Probability of false alarm (P_{fa}):** It measures how many samples are predicted as absent out of all the samples that are actually present.

III. RESULTS AND DISCUSSIONS

In this section, we present the performance of three ML based models employed for spectrum sensing in CR.
For DT model, the depth of the tree is varied to observe the effect on accuracy of the model. Fig. 5 illustrates the variation of testing accuracy versus depth of the tree for DT model. Maximum accuracy is achieved at a depth of 20 and remains constant after that and as the depth is decreased accuracy diminishes.

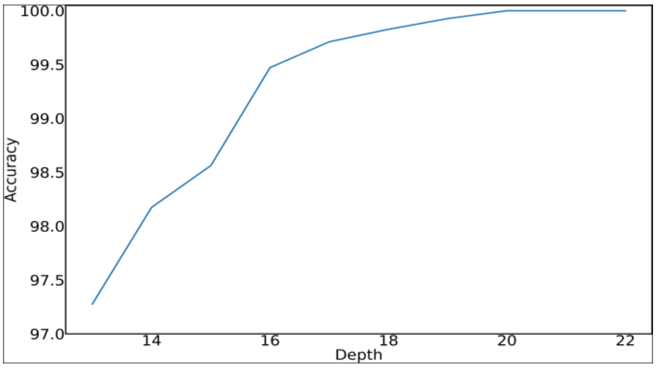

Fig. 5: Accuracy vs Depth of tree for DT

For SVM model, both linear and non-linear kernels were used to get best accuracy results. Table 2 shows the accuracy for different kernels in SVM. As linear kernel has better accuracy compared to Non-linear kernels, Linear kernel is chosen for training the SVM model.

Table 2: Accuracies for different SVM kernels

Kernels	Accuracy(%)
Linear	74.22
RBF	73.94
Polynomial	72.08

In KNN model, different values of number of neighbours (K) are chosen to observe the change in accuracy. Fig. 6 illustrates the plot for number of neighbours (K) vs Accuracy.

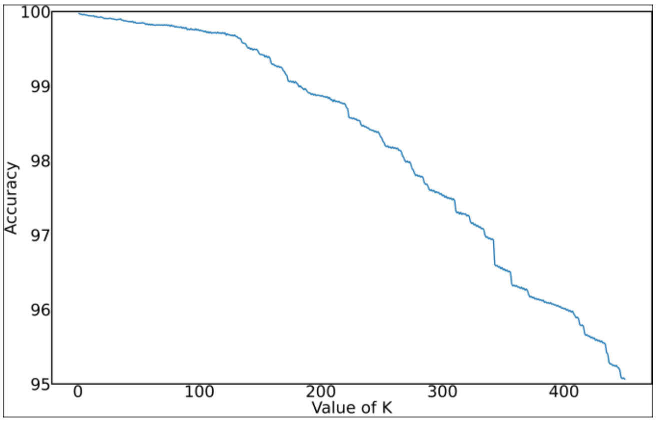

Fig. 6: Accuracy vs Number of neighbours (K) for KNN

It is observed that as the value of K increases up to a certain extent (around K=130-140), the training accuracy decreases linearly. After that value when K is increased further, the accuracy drops drastically due to under-fitting of the model. After determining the accuracy of all three ML models the depth of tree as 15 in DT, linear Kernel in SVM and K= 447 in KNN is chosen and the presence of spectrum is detected with each model.

Fig. 7 illustrates the plot of probability of detection vs SNR for various models. It is observed that in case of SVM for low values of SNR, the probability of detection is very low and detection is difficult with this algorithm. However, DT and KNN perform better for lower values of SNRs. It can also be observed that for SNR values greater than 10 dB, all the algorithms perform similarly. Therefore, any algorithm can be chosen for higher SNR values.

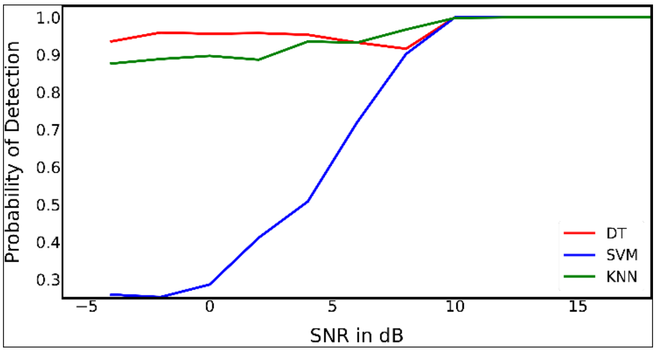

Fig. 7: P_d vs SNR for various ML models in SS

Fig. 8 shows the plot of Accuracy vs SNR for various models. It is observed that for SVM the accuracy decreases at first when SNR is increased as it is not able to distinguish between signal and noise at low values of SNR and SVM is a complex model and it undergoes over-fitting which includes the underlying features which has information about signal and noise and also fits random noise present in the data. The accuracy increases after a certain value of SNR as the signal becomes prominent and it is easier for the model to predict presence of user. Even for KNN, accuracy decreases at certain SNR values due to over-fitting problem. Over-fitting occurs for lower values of K in KNN. Thus, for higher SNR values, all models give high accuracy but for lower SNR values, DT is the best choice.

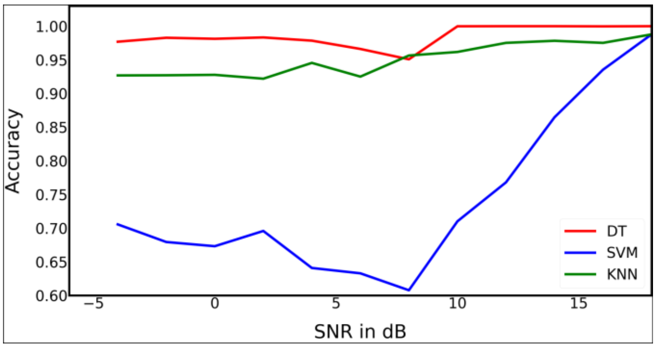

Fig. 8: Accuracy vs SNR for various ML models in SS

Table 3 illustrates the comparison of accuracy, P_d and P_{fa} for different models DT, SVM and KNN. Out of the three models, DT is found to perform better based on the evaluation parameters as this algorithm is very simple to understand and it has good computational efficiency.

Table 3. Comparison of Evaluation Parameters for various ML models

Models	**Accuracy (%)**	**Probability of Detection**	**Probability of False Alarm**
Decision Tree	**98.49**	**0.97**	**0.0025**
Support Vector Machine	74.22	0.81	0.35
K Nearest Neighbour	95.08	0.96	0.07

IV. CONCLUSION

The conventional methods for SS like, energy detection, matched filter and cyclo-stationary are time consuming and require prior knowledge of the surrounding environment. Moreover, SS using matched filter is complex as it requires threshold optimization. Machine Learning based model is a viable solution for spectrum sensing in cognitive radio that provides simplicity, high accuracy and does not require any thresholding. One of the main criteria for using ML algorithms in spectrum sensing is the ability to utilize them in the absence of knowledge of the radio in real time. In this work, it is observed that a larger comprehensive data size gives better training accuracy as compared to [6]. Decision tree performs better due to higher computational efficiency. Probability of Detection is high for higher values of SNR.

At low SNR also all models show higher probability of detection compared to [8]. SNR is used to detect the desired signal strength in analog and digital communications which compares two levels in ratio and determines the impact of noise on desired signal. Probability of False alarm obtained for all the three models are almost negligible and are much better than in [8]. Machine Learning helps in improving the determination probability of the model and also better utilization of spectrum. The results of this study may help in the creation of effective and dependable SS methods that will enhance the functionality and use of wireless networks under all SNR conditions for 5G.

REFERENCES

[1] M. Saber, A. E. Rharras, R. Saadane, A. H. Kharraz, and A. Chehri, “An Optimized Spectrum Sensing Implementation Based on SVM, KNN and TREE Algorithms,” *Signal-Image Technology and Internet-Based Systems*, Nov. 2019, doi: 10.1109/sitis.2019.00068.

[2] A. Parvathy and G. Nsarayanan, “Comparative Study of Energy Detection and Matched Filter Based Spectrum Sensing Techniques,” *International Conference on Computational Intelligence and Communication Networks*, Sep. 2020, doi: 10.1109/cicn49253.2020.9242609.

[3] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," in *IEEE Personal Communications*, vol. 6, no. 4, pp. 13-18, Aug. 1999, doi: 10.1109/98.788210.

[4] Ashish Ranjan, Anurag and Balwinder singh, “Design and Analysis of Spectrum Sensing in Cognitive Radio based on Energy Detection ”, *IEEE explore”, September 2016.*

[5] Xinzhi Zhang, Rong Chai and Feifei Gao, ”Matched Filter Based Spectrum Sensing and Power Level Detection for Cognitive Radio Network”, “*Signal Processing for Cognitive Radios and Networks*”, *978-1-4799-7088-9/14/$31.00 ©2014 IEEE* GlobalSIP 2014.

[6] Khadeeja Sherbin.M and Ms .Sindhu.V,” Cyclostationary Feature Detection for Spectrum Sensing in Cognitive Radio Network” , ”*International Conference on Intelligent Computing and Control*

Systems ",IEEE Xplore Part Number: CFP19K34-ART; *ISBN: 978-1-5386-8113-8*, (ICICCS 2019).

[7] E. E. A. Medina and S. E. Barbin, "Performance of Spectrum Sensing Based on Energy Detection for Cognitive Radios," *Topical Conference on Antennas and Propagation in Wireless Communications*, Sep. 2018, doi: 10.1109/apwc.2018.8503791.

[8] N. Usha, K. S. Reddy, and N. N. Nagendra, "Dynamic Spectrum Sensing in Cognitive Radio Networks using ML Model," *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Aug. 2020, doi: 10.1109/icssit48917.2020.9214146.

[9] S. U. Jan, H. Vu-Van, and I. Koo, "Performance Analysis of Support Vector Machine-Based Classifier for Spectrum Sensing in Cognitive Radio Networks," *Cyber-Enabled Distributed Computing and Knowledge Discovery*, Oct. 2018, doi: 10.1109/cyberc.2018.00075.

[10] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, "Machine Learning Techniques for Cooperative Spectrum Sensing in Cognitive Radio Networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 11, pp. 2209–2221, Oct. 2013, doi: 10.1109/jsac.2013.131120.

[11] Y. Arjoune and N. Kaabouch, "On Spectrum Sensing, a Machine Learning Method for Cognitive Radio Systems," *Electro/Information Technology*, May 2019, doi: 10.1109/eit.2019.8834099

[12] H. A. Shah and I. Koo, "Reliable Machine Learning Based Spectrum Sensing in Cognitive Radio Networks," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–17, Sep. 2018, doi: 10.1155/2018/5906097

[13] Kürşat Tekbıyık, Cihat Keçeci, Ali Rıza Ekti, Ali Görçin, Güneş Karabulut Kurt, October 27, 2019, "HisarMod: A new challenging modulated signals dataset", IEEE Dataport, doi: https://dx.doi.org/10.21227/8k12-2g70.

[14] M. Simon, J. Omura, R. Scholtz, and B. Levitt, *Spread Spectrum Communications Handbook*, Electronic Edition. 2001.

[15] A. M. Wyglinski and D. Pu, *Digital Communication Systems Engineering with Software-Defined Radio*. 2013.

[16] A. C. Müller and S. Guido, *Introduction to Machine Learning with Python*. 2016.