EURASIP Journal on Wireless Communications and Networking

RESEARCH **Open Access**

Efficient improvement of energy detection technique in cognitive radio networks using K-nearest neighbour (KNN) algorithm

Aneesh Sarjit S. Musuvathi[1], Jofin F. Archbald[1], T. Velmurugan[1*], D. Sumathi[2], S. Renuga Devi[1] and K. S. Preetha[1]

*Correspondence:
tvelmurugan@vit.ac.in

[1] School of Electronics Engineering, Vellore Institute of Technology, Tamil Nadu, Vellore 632014, India
[2] School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Tamil Nadu, Vellore 632014, India

Abstract
With the birth of the IoT era, it is evident that the existing number of devices is going to rise exponentially. Any two devices will communicate with each other using the same frequency band with limited availability. Therefore, it is of vital importance that this frequency band used for communication be used efficiently to accommodate the maximum number of devices with the available radio resources. Cognitive radio (CR) technology serves this exact purpose. The stated one is an intelligent radio that is made to automatically identify the optimal wireless channel in the available wireless spectrum at a given instant. An important functionality of CR is spectrum sensing. Energy detection is a very popular algorithm used for spectrum sensing in CR technology for efficient allocation of radio resources to the devices intended to communicate with each other. Energy detection detects the presence of a primary user (PU) signal by continuously monitoring a selected frequency bandwidth. The conventional energy detection technique is known to perform poorly in lower SNR ranges. This paper works towards the improvement of the energy detection algorithm with the help of machine learning (ML). The ML model uses the general properties of the signal as training data and classifies between a PU signal and noise at very low SNR ranges (−25 to −10 dB). In this research, a K-nearest neighbours (KNN) model is selected for its versatility and simplicity. Upon testing the model with an out-of-sample dataset, the KNN model produced a detection accuracy of 94.5%.

Keywords: Cognitive radio (CR), Energy detection (ED), Detection accuracy, Machine learning, K-nearest neighbour

1 Introduction
With the advancement of wireless communication technology, more and more devices are finding usage among people. Hence, the demand for the frequency spectrum too increases rapidly. However, the radio resources are not available in an unlimited way. These resources being scarce have to be used efficiently in order to accommodate more and more devices. But for various reasons, these resources, mainly the frequency spectrum, are not entirely used or are not being used efficiently as suggested by many studies. To address this issue, cognitive radio (CR) technology has been developed. CR is a dynamic paradigm that allocates the available radio resources efficiently to the devices

looking to communicate [4]. This technology caters to the efficient use of the limitedly available frequency spectrum. An important functionality in CR technology is spectrum sensing.

Spectrum sensing is an important functionality of CR technology. This process continuously monitors a selected bandwidth or channel in the frequency spectrum to check for vacancy. These vacancies present in the channel are called spectrum holes. By leveraging the use of these spectrum holes, we improve the utilization of the spectrum [2]. Hence, it is important that the spectrum sensing technique performs effectively so that it finds these spectrum holes and this can be used by the CR technology to accommodate the new devices looking to communicate. CR technology assumes two kinds of bands: a licensed band and an unlicensed band [6]. Users who own a band in the spectrum become licensed users of the band and are entitled to use the band at any time, and hence, these users are called primary (PU) users as shown in Fig. 1. Other users who look to use any available band are referred to as secondary users. Since the primary users do not always use their licensed band, these bands can be used by the secondary users to communicate [3]. These gaps in these licensed bands can be identified by spectrum sensing algorithms. The effectiveness of the spectrum sensing algorithm becomes very important so as to find the gaps effectively and maximize the use of the unused channels when not in use by the primary users. Spectrum is one of the functionalities of CR technology. It has three other main functions—spectrum mobility, spectrum management and spectrum sharing. There are several spectrum sensing methods such as matched filter detection, cyclo-stationary detection, waveform detection algorithm and many more. Among all of the available spectrum sensing algorithms. The most popular is the energy detection algorithm (ED). The ED algorithm has gained popularity because of its less complexity compared to the other algorithms [15]. Research is carried out on this algorithm in different ways to improve its performance in different conditions, as the

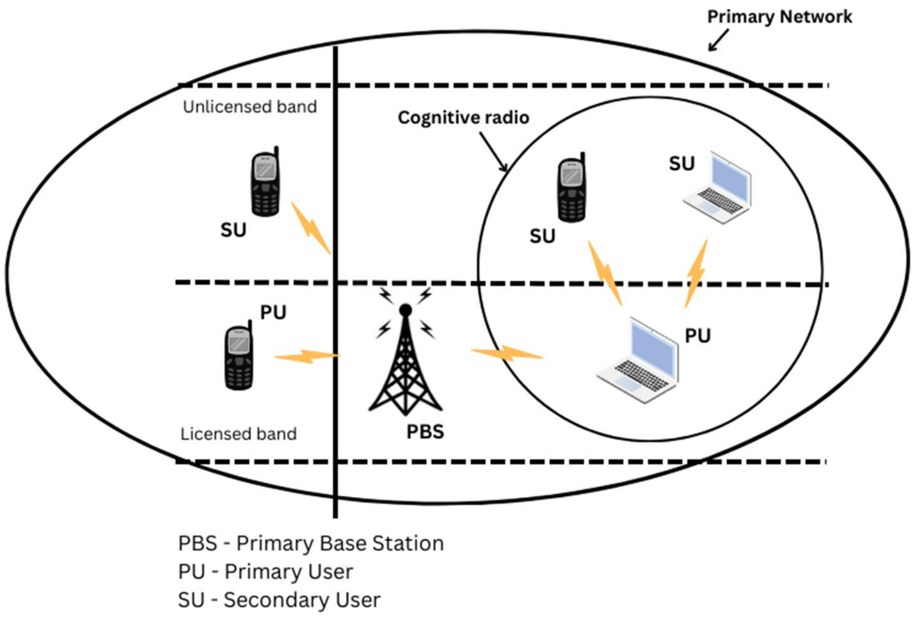


Fig. 1 Illustration of cognitive radio network

algorithm is not performing well under all the cases, especially in the lower SNR regions, that is when the power of the noise is as significant as the PU signal. However, studies have helped develop improved versions of the algorithm that perform much better than the conventional ED (CED) system. Yet, there is still scope to take this system ahead in terms of its performance.

In today's world, the need to solve a variety of complex real-world problems more efficiently and accurately than their conventional counterparts is important. On this account, machine learning has been nearing its pinnacle for over a decade in the field of research and development. Also, with the rise of heavy-duty computational chips embedded within the smallest of electronic devices, the computational power required for edge machine learning (ML) has only been getting more reliable for incorporation. This scenario also applies to wireless communication technologies. In the vast field of communication technologies, there are countless research projects going on for the incorporation of ML into the older and more conventional methods to improve performance and efficiency so as to meet real-world market demands. Therefore, we have used ML to improve the performance of the energy detection method in the lower ranges of SNR value used in CR networks [1]. This paper discusses a solution that uses K-nearest neighbours (KNN) classification algorithm to classify PU signals from noise. Using MATLAB software, signals were simulated and their properties were extracted based on the detection. These properties are the features used for training the KNN model. After training and hyperparameter tuning, the model was evaluated and tested with an unknown dataset. The complete ML process was conducted using Python programming language.

2 Literature review

Analysis of energy detection techniques was studied and analysed in many research cases. Energy detection technique though being one of the simplest and well-performing spectrum sensing techniques, required further attention to improve its performance under low signal-to-noise ratios [2]. There was further discussion of how fading, shadowing, and concealed terminal issues affected detection performance [4]. Despite a thorough analysis of energy detection methods, this study was unable to pick up PU signals at low SNR levels due to high levels of noise density. Further studies were carried out with this motive and several techniques had been introduced to improve the performance with their own merits and demerits. In [6], to sense spectrum pivoted on detecting power in the ambiance of Rayleigh fading was narrated. The mathematical equation in closed form for AWGN as well as Rayleigh channels was derived together with the respective detection probability as well as false alarm probability related to SNR. The outcomes of theoretical computations and simulations were compared. This study's comparison revealed that Rayleigh channels had a lesser likelihood of picking up a primary signal than AWGN channels. The findings of this investigation demonstrate that the detection probability increased along with the SNR value. Also, it was clear that a rise in the likelihood of a false alert improved detection efficiency. [2] had discussed the effects of introducing entropy techniques in energy detection algorithms. Here by taking a large number of samples from the received signal at low SNR values the distribution curve could be assumed to be a Gaussian curve. This enabled the author to choose one among the many methods of entropy. For example, entropies related to Gaussian distribution or uniform distribution. All the test statistics and other parameters were estimated as Gaussian distribution and the conclusions were made. This did a slight improvement in the lower SNR regions. The recent advancements in this field led forward with the help of ML. In the study conducted in [1], training of models of ML was done utilizing the characteristics of power of the system's primary client as well as the rest of the clients available inside the system. The effort to improve was performed by incorporating ML algorithms to discover the network's operational characteristics. This system served to be a discriminating one to distinguish among SU, PU as well as an interfering one of potential type in the study. Hence, such a study was satisfying the interferences minimization effort while maintaining the CED's sensitivity. However, the study does not usage of characteristics other than power to determine presence of PU signal. In recent times, KNN algorithms were utilized in major areas in the field of electronics and communication for their simplicity and adaptability. This was shown in [7], where authors had improved the diagnostic accuracy of power transformer faults using the KNN classification algorithm through optimization. An equally important advantage of the KNN was the ease of hyperparameter tuning as they had only two major parameters to tune. The aforementioned study and [8] incorporated hyperparameter tuning extensively to improve the detection accuracies of the model in various other fields of electronics and communication. These researches have galvanized to work with the KNN model in the present research in the hopes of improving the detection capability of the ED system.

The research gap that was found in the following literature review is that a majority of the research focusses only on a tight bound of SNR ranges. This could improve research focus; however, the main conundrum is detecting the PU signal at extremely low SNR values. The conventional noise cancellation does not work at very low SNR values as there is lots of signal interference resulting in lots true negative values. Also, the previous researches have also not considered on how to incorporate machine learning in the practical applications as they are too complex to carry out. In our approach, it is much easier to extract signal properties in the practical standpoint, since they can be made as modules, and with the advent of transfer learning, we can improve the speed of training algorithms by using pre-trained algorithms. In overall sense, our research shows a much greater promise for practical applicability.

3 Contribution of this paper

To sense the spectrum is a function of vital concern in the networks of radio of cognitive type. Among the schemes that are different that are available for spectrum sensing, energy detection has been the more recent and more sophisticated method for various reasons [13, 14]. However, the conventional and its improved versions (entropy method) have performed satisfactorily well under higher SNR ranges, but the performance of the same methods is very poor for low ranges of SNR. The purpose of this research is to develop an effective energy detection system by incorporating machine learning. The capability of machine learning to find complex relations between multiple independent variables, especially supervised learning, has brought great performance improvement in various domains. Many research was conducted in bringing machine learning to energy detection in past years, where the signals are taken directly training and predictions for detection of

PU. However, none of them approach in a discrete like in our research, wherein we extract properties of the signal and those become the independent variables for the machine leaning algorithm. After many trials with many models, we have chosen the K-nearest neighbour algorithm, that performs with high accuracy (detecting for the presence or absence of primary user signal in the channel) consistently at all ranges of SNR. The proposed system in this paper improves spectrum sensing in CR networks and thereby helps in the effective usage of the limitedly available frequency bandwidth used for communication.

4 Methodology

4.1 System model

The method for detecting power, a popular method to sense the spectrum, is employed in this study for sensing the frequency band. Following is the block diagram showing how the ED model works in Fig. 2 [22].

The signals that went via BPF (utilized for variance of noises normalizing as well as for noises energy minimizing) with W as bandwidth, initially get squared utilizing a device of square-law [9, 10], followed by a sum (integration is done for obtaining a continuous signal) across the observed period T to determine the received signal energy. The received energy of the signal also known as the test statistics is given as:

$$E(s) = \frac{1}{N}\sum |s(n)|^2 \tag{1}$$

n order to establish whether a licensed user is present, the summation's output (or integration) comparison is done with a threshold predetermined λ [25].

The energy detection model concludes between two hypotheses, as shown below:

$$s(n) = \begin{cases} H0 : g(n) \\ H1 : \mathrm{pu}(n) + g(n) \end{cases} \tag{2}$$

where $g(n)$ is the AWGN noise signal, $\mathrm{pu}(n)$ is the primary user signal, and $s(n)$ is the received signal. Here, hypothesis $H1$ is an indication of the existing primary user signal and $H0$ indicates its absence [11].

4.2 Entropy method (ED + EN)

Among the different kinds of methods for entropy, namely uniform distribution entropy and Gaussian distribution entropy, the latter has been chosen. This is because on observing the signal at high sample rates, the received signal tends to become a Gaussian curve. This occurrence can be clarified by the Central Limit Theorem in statistics, which asserts that the distribution of the sum or average of a large number of independent, identically distributed random variables converges to a normal (Gaussian) distribution, irrespective of the original distribution of the variables [14]. When a signal is sampled at a high rate, each

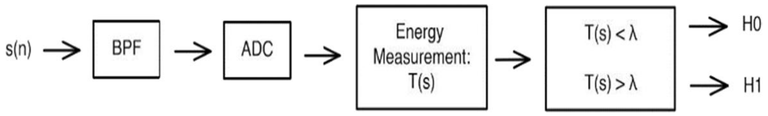


Fig. 2 Block diagram of CED

sample can be seen as an independent random variable. With an increase in the number of samples, the distribution of the overall signal tends to mimic a Gaussian curve [26]. As shown in the flowchart in Fig. 3, the Gaussian PDF is computed for test statistic $T(s)$. The result of this PDF is later fed into the discrete entropy method, $H(x)$, and the resulting value is compared against the threshold for detection.

This addition of the entropy technique to the ED system has resulted in the reduction of Pf, therefore improving the detection accuracy at low SNR ranges [24].

On studying the energy detector, it becomes evident that the system is incapable of detecting the primary user (PU) signal existence in a highly noisy environment, i.e. during small SNR ranges [16, 17]. In the instant of such range, while the power of the PU signal is similar to that of the noise, the CED system cannot differentiate between the two signals. Moreover, the uncertainties of the noise characteristics further degrade the performance of the CED [20]. Therefore, to improve the CED's performance, we propose a novel system using KNN.

4.3 Proposed model: usage of K-nearest neighbour algorithm

The k-nearest neighbours (KNN) algorithm, a cornerstone in supervised machine learning, operates on the principle of proximity-based prediction. In the training phase, KNN memorizes the entire dataset, and during the prediction phase, it identifies the k-nearest neighbours to a given data point in the feature space. The algorithm's strength lies in its simplicity and interpretability, making it particularly well-suited for scenarios where decision boundaries are irregular. However, its efficiency is effective on careful consideration of parameters such as the number of neighbours (k) and the choice of distance metric. KNN has demonstrated success in various real-world applications, including

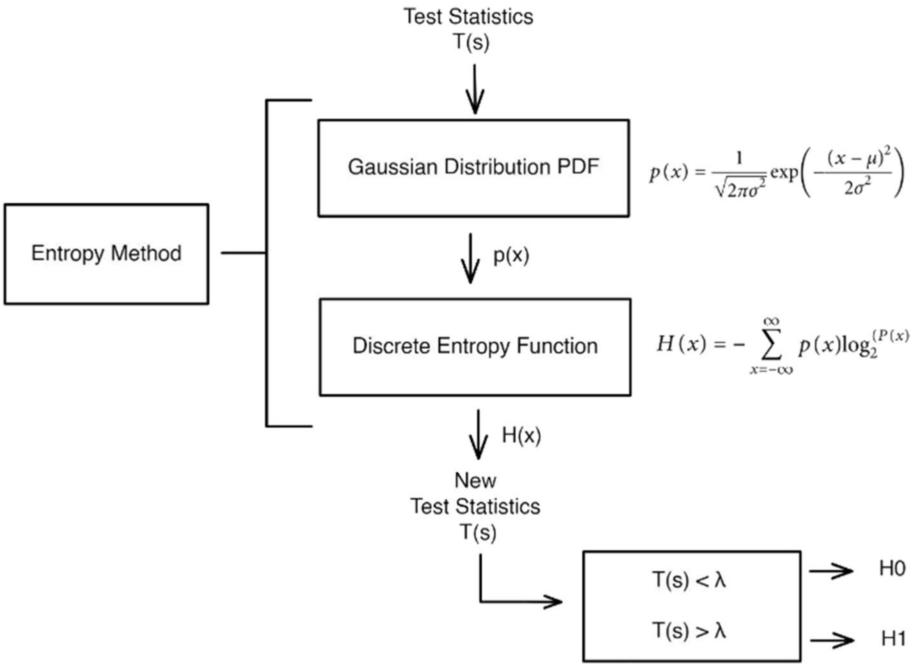

Test Statistics
T(s)
Gaussian Distribution PDF
$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
Entropy Method
p(x)
Discrete Entropy Function
$H(x) = -\sum_{x=-\infty}^{\infty} p(x)\log_2^{(P(x)}$
H(x)
New
Test Statistics
T(s)
T(s) < λ
T(s) > λ
H0
H1

Fig. 3 Block Diagram explaining the process of EN in ED + EN

image recognition and healthcare. The proposed technique is aimed to improve accuracy in the detection of PU signals using the KNN machine learning algorithm.

First, the data for model training are collected in a scenario-based approach. It is assumed that the noise in a particular environment is generally static and consistent.

$$D = \{(x_1, y_1), \ldots, (x_{64000}, y_{64000})\} \subseteq R_{14} \times C_2 \tag{3}$$

Equation (3) shows the training data D used in the research, and Eq. (4) shows the representation of each sample of the data of size 64,000. It consists of 15 fields, out of which 14 fields have properties of the signal and the 15th column contains the Boolean value of whether the PU signal was present. The (x, y) are the pairs of inputs such that $x_i = \left(x_i^1, x_i^2, \ldots, x_i^{14}\right)$ as they are 14 properties that were extracted from the signal and $y_i = \{0, 1\}$ is the target variable that represents the detection of PU signal. When the target variable results in "0", it depicts the H0 hypothesis, meaning that the signal is composed of only the general AWGN signal, $g(n)$. The target variable "1" depicts the $H1$ hypothesis, which is an additive result of the primary user, pu(n) signal and $g(n)$.

$$(x, y) \in D = \begin{cases} \overbrace{\{\mathcal{P}(\mathrm{pu}(n) + g(n)), 1\}}^{32000 \text{ samples}} \\ \underbrace{\{\mathcal{P}(g(n)), 0\}}_{32000 \text{ samples}} \end{cases} \tag{4}$$

where $\mathcal{P}()$ is a custom-built function that extracts various properties of the signal and these properties are the features that are used to train the machine learning model. Equation (4) represents how the dataset was built for supervised learning. The first case would be a signal with noise and that be 1 (TRUE) value, the second case would be to have only noise that would be 0 (FALSE) value.

Figure 4 represents the methodology of the proposed model implemented using MATLAB and Python programming language. The dataset was constructed using MATLAB. The primary user signal generated for the dataset consisted of 1000 samples of varying amplitude within a constrained range. To these primary user signals, random AWGN signals were added in a varying SNR range of − 25 to − 10 dB. This is to replicate the different SNR values in the environment. For each SNR value, 1000 Monte Carlo simulations were generated [19]. The properties of all these signals were extracted and stacked

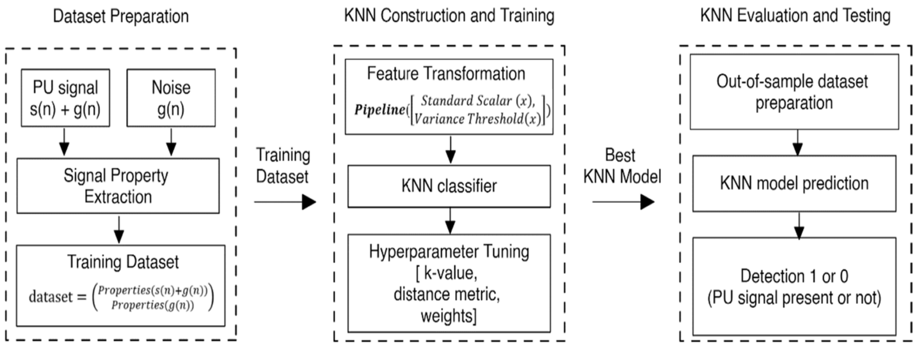

Dataset Preparation
PU signal s(n) + g(n)
Noise g(n)
Signal Property Extraction
Training Dataset
dataset = (Properties(s(n)+g(n)) Properties(g(n)))
Training Dataset
KNN Construction and Training
Feature Transformation
Pipeline([Standard Scalar (x), Variance Threshold(x)])
KNN classifier
Hyperparameter Tuning [k-value, distance metric, weights]
Best KNN Model
KNN Evaluation and Testing
Out-of-sample dataset preparation
KNN model prediction
Detection 1 or 0 (PU signal present or not)

Fig. 4 Block diagram showing the methodology of the proposed system

into arrays thereby producing a dataset of 64,000 samples. Before feeding the dataset into the KNN model for training it has to be transformed. The values of all the features are scaled and standardized evenly.

$$T(x) = \left\{ \frac{(x_i - \mu)}{\sigma} \middle| \frac{\sum_{i=1}^{n}(x_i - \mu)}{n-1} \right\} \tag{5}$$

Equation (5) shows the pipeline $T(x)$ that consists of standard scaling and variance thresholding. Standard scaling scales all the features x_i and standardizes them evenly using the mean μ and standard deviation σ. The variance threshold was built to automatically drop the low-variance features that do not contribute to the decision-making process of the model [21]. The second segment of Eq. (5) shows the variance formula. This is a particularly important pipeline as the KNN algorithm is known to perform poorly in the presence of irrelevant features or improperly scaled features.

The dataset was generated in such a way as to satisfy various values of low ranges of SNR to improve the model's performance.

K-nearest neighbour is a supervised learning algorithm that uses distances about a single data point to make predictions. k is the number of neighbouring points whose distances from the test point are considered for training and prediction. The value is solely based on the dataset, and a higher value is useful for data with high noise and outliers.

$$N_x = \left\{ \left(x'', y''\right) : \left(x'', y''\right) \in D, \mathrm{dist_c}(x, x\prime) \geq \max\left(\mathrm{dist_c}\left(x, x''\right)\right) \right\} \tag{6}$$

such that $\left(x\prime, y\prime\right) \in D - N_x$

Equation (6) is about the overall general process of K-nearest neighbour algorithm during the training process. N_x, which is the set of k neighbouring points from dataset D. It is built on the condition that the points that are in D and not in N_x have a distance greater than the point in N_x that has the largest distance from x. Here $|N_x| = k$, and for our research, values of k were tried from 11 to 300 (randomly). $\mathrm{dist_c}()$ is the distance function and $c = \{\mathrm{Euclidean, Manhattan, Minkowski}\}$. In our research, we tried these 3-distance metrics to find the best results.

Equation (7) shows Euclidean distance. This distance shows the shortest path between the two points in space. d denotes the dimension of the vectors x and $x\prime$ such that $d < 14$ due to the dropping of features during variance thresholding.

$$\mathrm{dist_{euclidean}}(x, x\prime) = \sqrt{\sum_{i=1}^{d} (x_i - x_i\prime)^2} \tag{7}$$

Equation (8) shows the Manhattan distance. This distance measures the absolute difference between the two points in space. It is used as a distance metric in KNN as it produces a grid-like paths which can shape the decision boundary in a unique way.

$$\mathrm{dist_{manhattan}}(x, x\prime) = \sum_{i=1}^{d} |x_i - x_i\prime| \tag{8}$$

Equation (9) shows Minkowski distance. This distance is a generalized version of distance metrics which has a power parameter p.

$$\begin{aligned} \mathrm{dist}_{\mathrm{minkowski}}\left(x,x',p\right) &= \sqrt[p]{\sum_{i=1}^{d}\left(x_i - x_i'\right)^p}, \quad p \geq 1 \\ \mathrm{dist}_{\mathrm{minkowski}}\left(x,x',1\right) &= \mathrm{dist}_{\mathrm{manhattan}} \\ \mathrm{dist}_{\mathrm{minkowski}}\left(x,x',2\right) &= \mathrm{dist}_{\mathrm{Euclidean}} \end{aligned} \tag{9}$$

The KNN classifier $\mathcal{K}(x)$ is defined as the most frequent class found in the N_x.

$$\mathcal{K}(x) = \mathrm{mode}(\{y'' : (x'', y'') \in N_x\}) \tag{10}$$

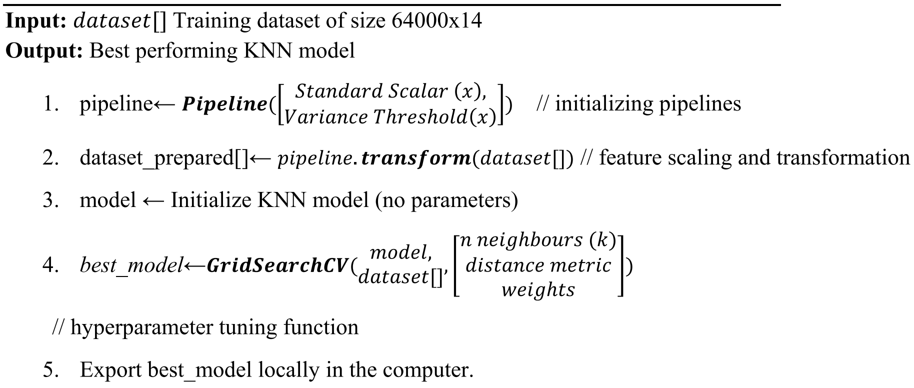
$\mathcal{K}(x)$ is the uniform KNN classifier which does not consider the distance in general for prediction. $\mathcal{K}_w(x)$ is a weighted KNN that considers weights (weights are assigned to each point such that they are inversely proportional to the distance from the query point) and predicts with respect to the weighted condition. The equation is given by

$$\mathcal{K}_w(x) = \mathcal{F}(\{y'' : w_i(x'', y'') \in N_x, w_i \in W\}) \tag{11}$$

where $\mathcal{F}()$ is the distance weighted voting function and W is the weights vector that contains weights corresponding to each neighbour point in N_x.

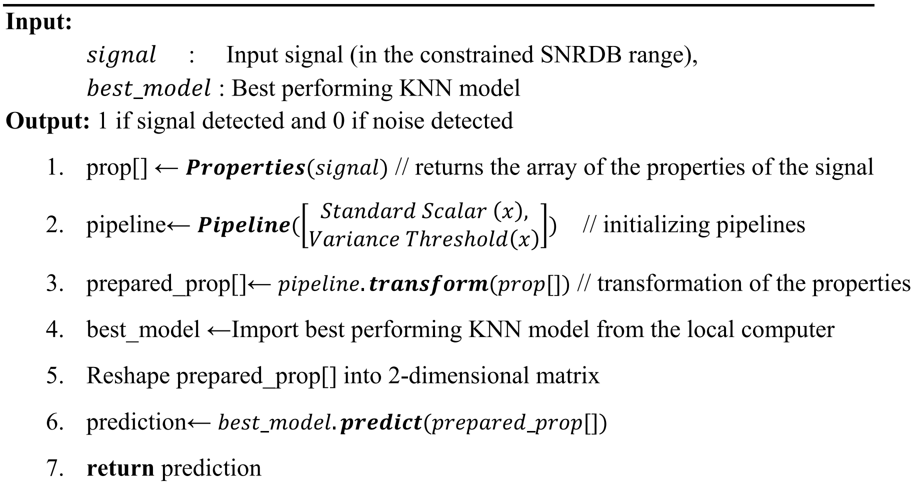
Algorithms 1, 2, and 3 showcase how machine learning is incorporated in the energy detection algorithm. *Pipeline*([]) function takes in the standard scaler and variance threshold transformations in the single pipeline and the *pipline.transform*() function takes in x_i and performs the respective transformations and produces and clean input array for training the model. The *model.predict*() takes an input dataset and makes a prediction using the trained model *model*.

Algorithm 1 Construction and Training of the KNN model



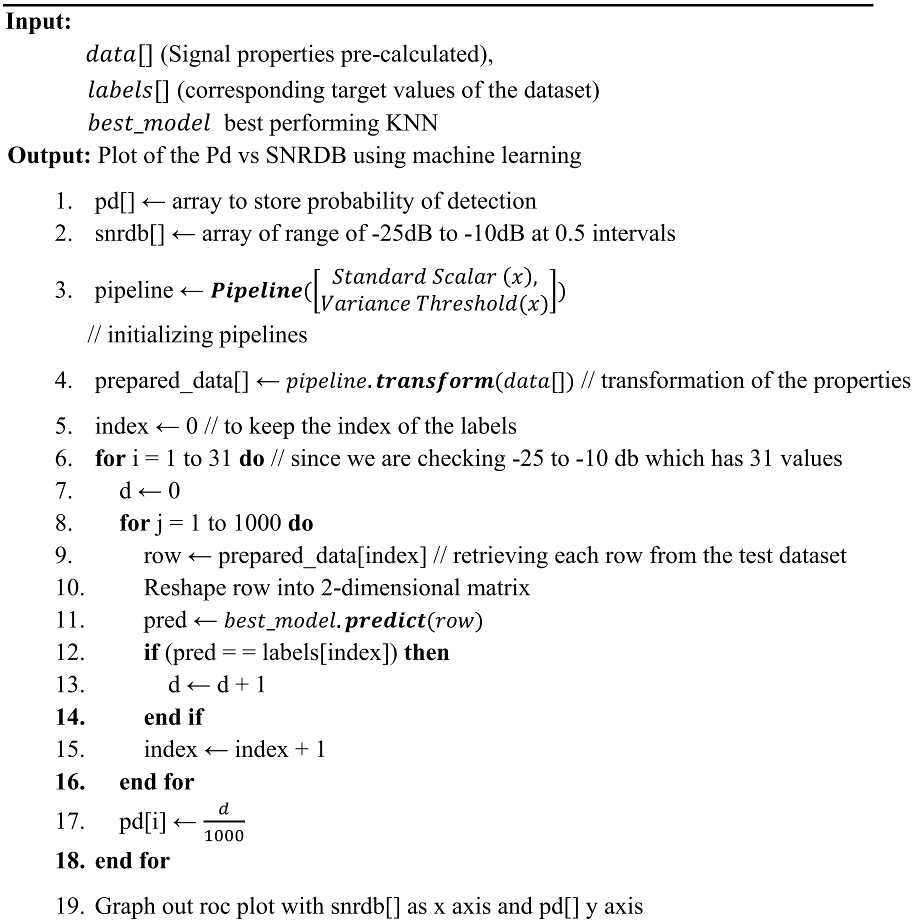
```
Input: $dataset[]$ Training dataset of size 64000x14
Output: Best performing KNN model

pipeline← $\boldsymbol{Pipeline}(\begin{bmatrix} Standard\ Scalar\ (x), \\ Variance\ Threshold(x) \end{bmatrix})$ // initializing pipelines
dataset_prepared[]← $pipeline.\boldsymbol{transform}(dataset[])$ // feature scaling and transformation
model ← Initialize KNN model (no parameters)
$best\_model$←$\boldsymbol{GridSearchCV}(\begin{matrix} model, \\ dataset[] \end{matrix}, \begin{bmatrix} n\ neighbours\ (k) \\ distance\ metric \\ weights \end{bmatrix})$
// hyperparameter tuning function
Export best_model locally in the computer.
```

Algorithm 2 Energy Detection using KNN for a single signal



```
Input:
    $signal$ : Input signal (in the constrained SNRDB range),
    $best\_model$ : Best performing KNN model
Output: 1 if signal detected and 0 if noise detected
    prop[] ← $\boldsymbol{Properties}(signal)$ // returns the array of the properties of the signal
    pipeline← $\boldsymbol{Pipeline}(\begin{bmatrix} Standard\ Scalar\ (x), \\ Variance\ Threshold(x) \end{bmatrix})$ // initializing pipelines
    prepared_prop[]← $pipeline.\boldsymbol{transform}(prop$[]) // transformation of the properties
    best_model ←Import best performing KNN model from the local computer
    Reshape prepared_prop[] into 2-dimensional matrix
    prediction← $best\_model.\boldsymbol{predict}(prepared\_prop$[])
    return prediction
```

Algorithm 3 Energy Detection performance measurement for the proposed KNN model



```
Input:
    $data$[] (Signal properties pre-calculated),
    $labels$[] (corresponding target values of the dataset)
    $best\_model$ best performing KNN
Output: Plot of the Pd vs SNRDB using machine learning
    pd[] ← array to store probability of detection
    snrdb[] ← array of range of -25dB to -10dB at 0.5 intervals
    pipeline ← $\boldsymbol{Pipeline}(\begin{bmatrix} Standard\ Scalar\ (x), \\ Variance\ Threshold(x) \end{bmatrix})$
    // initializing pipelines
    prepared_data[] ← $pipeline.\boldsymbol{transform}(data$[]) // transformation of the properties
    index ← 0 // to keep the index of the labels
    for i = 1 to 31 do // since we are checking -25 to -10 db which has 31 values
        d ← 0
        for j = 1 to 1000 do
            row ← prepared_data[index] // retrieving each row from the test dataset
            Reshape row into 2-dimensional matrix
            pred ← $best\_model.\boldsymbol{predict}(row)$
            if (pred = = labels[index]) then
                d ← d + 1
            end if
            index ← index + 1
        end for
        pd[i] ← $\frac{d}{1000}$
    end for
    Graph out roc plot with snrdb[] as x axis and pd[] y axis
```

5 Results and discussion

The dataset D (Eq. 3) is a training dataset that is used to train the KNN model. The direct approach of training will not provide ample data about the best parameter to be used to attain the best performance. For this purpose, the work used Grid Search Cross-Validation Technique to train many instances of KNN for different parameter combinations [8]. Table 1 contains simulation parameters that were used for the research.

In our research, we have considered 3 parameters: number of neighbours k, distance metric c, and the type of KNN t, for hyperparameter tuning to find the most suitable parameter set that produces the best accuracy.

$$k = \{10, 30, 50, 100, 150, 200, 250, 300\}$$

$$c = \left\{\text{Euclidean, Manhattan, Minkowski}\right\}$$

$$t = \{1, W\}$$

Each of them is a vector and to find the model with the highest performance, we must try all the combinations of (k, c, t). The performance metric that is used for training is detection accuracy.

Since the higher the accuracy, the better the model. Therefore, we use the *argmax* function. Now say $\mathcal{G}()$ is a function that gives the training accuracy of KNN defined using p, q, and r. The final equation to find the best KNN classifier is

$$\begin{aligned} \arg \max_{p\in k, q\in c, r\in t} & \mathcal{G}(p,q,r) \\ &= \left\{p,q,r \in k,c,t | \forall(u,v,w) \in (k,c,t) : \mathcal{G}(p,q,r) \right. \\ &\geq \mathcal{G}(u,v,w)\} \end{aligned} \tag{12}$$

Equation (12) runs a total of $\left|k\right| \cdot |c| \cdot |t| = 8 \cdot 3 \cdot 2 = 48$ iterations to find the best-performing model.

Figure 5 shows the best combinations of parameters among the 48 other combinations for the model. The k value of 200 along with the Manhattan distance metric and uniform KNN type performed better by a slight margin than the weighted KNN.

This KNN estimator was finalized for testing.

An out-of-sample dataset was fed into the trained KNN model, and the binary predictions were collected (1 indicating the presence of PU signal and 0 indicating its absence).

Table 1 Simulation parameters

Parameters	Values
SNR variation	− 25 to − 10dB
Channel	AWGN
Transformation techniques	Standard scaling, variance thresholding
Algorithm	K-nearest neighbours
No of neighbours	10, 30, 50, 100, 150, 200, 250, 300
Distance metrics	Euclidean, Manhattan, Minkowski
Algorithm type	Uniform KNN, weighted KNN

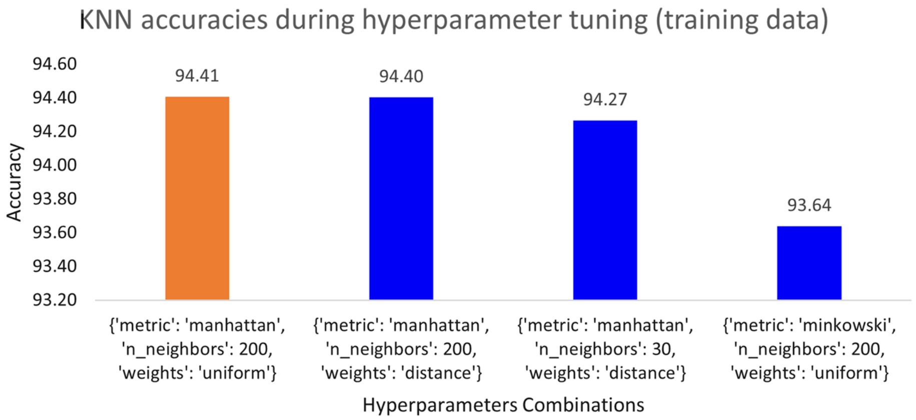


Fig. 5 Bar graph depicting KNN accuracy for 4 selected hyperparameter combinations

The classification results given by the model were compared against the actual results to calculate various performance measures.

Figure 6 compares the validation and out-of-sample testing accuracy of the best-performing KNN model. Table 2 shows the various performance measures of the KNN model, namely the accuracy, $F1$ score, and ROC–AUC score.

Detection accuracy, also called classification accuracy, is the most basic performance metric for a classifier model. It measures the number of correct predictions out of the total predictions.

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{total predictions}} \tag{13}$$

Accuracy gives an overview of how well the model is performing. The proposed KNN model performs with an accuracy of 94.58 during the testing phase. But to get a more accurate measurement, we work with F1 score and ROC–AUC score.

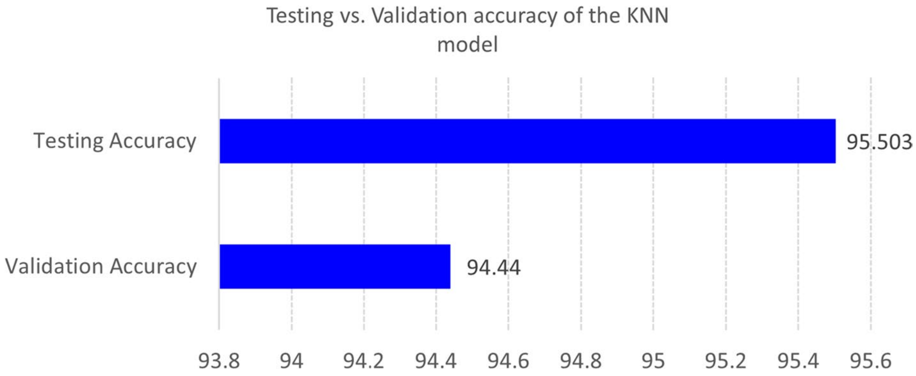


Fig. 6 Bar graph depicting the testing accuracy and validation accuracy of the best-performing KNN model

Table 2 Performance metrics of the best KNN classifier for validation and testing data

	Accuracy	F1 score	ROC–AUC score
Validation data	94.90	94.87	94.83
Testing data	94.58	94.46	94.50

$F1$ score is a combination of the precision and recall scores. The precision score is the accuracy of all the positive predictions the model has predicted. Recall is the ratio of the positive instances that are correctly detected by the classification model. It is also called sensitivity or the True Positive Rate. Equations (14) and (15) show the formula of precision and recall, respectively.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

Precision shows the ability to classify classes model correctly, while recall shows the efficiency of the classification. Since both are crucial to understanding the model well, the harmonic mean of precision and recall is the $F1$ score as shown in Eq. (16).

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{16}$$

This is because a regular mean treats all values equally, the harmonic mean gives weight to the lower values. This means that the $F1$ score is high only if both the recall and precision scores are high thereby giving a more effective metric for analysing the KNN model. The proposed KNN model produces a convincing $F1$ score of 94.87 and 94.46 in its validation and testing phases, respectively.

The ROC–AUC is usually derived from the area under the ROC curve. The ROC curve shows the performance of a model at all classification thresholds. The curve is plotted using true positive rate (TPR), which is the recall and false positive rate (FPR).

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{17}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{18}$$

A perfect ideal classifier gives a ROC–AUC score of 1, and a purely random classifier has a score of 0.5. The proposed KNN model gives a ROC–AUC score of 94.83 and 94.50 in its validation and testing phases, respectively.

Figure 7 shows the confusion matrix of the KNN model laid out on a heat map. The confusion matrix provides insights on how well the model performs in terms of true negative, false positive, false negative and true positive. The dimension of the matrix depends on the number of classes to be classified, and in this case, it is a 2×2 matrix.

The CED and other optimized versions of the system perform satisfactorily well in the higher ranges of SNR. Hence, this paper only focuses on analysing and improving its performance in lower SNR ranges (between -25 and -10dB). The performance of the CED system and CED system with entropy technique can be seen in Figs. 8 and 9 [5]. It can be easily seen that the performance of CED is very poor compared to the ED + EN system [23]. Though, the entropy technique too has been incorporated into

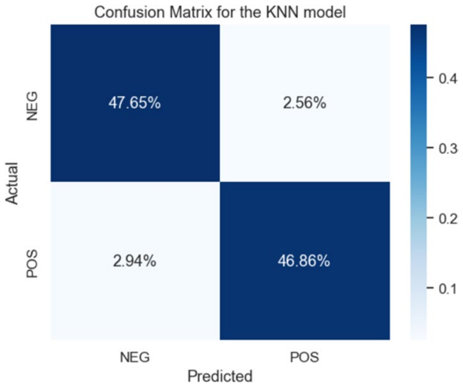


Fig. 7 Confusion matrix of the KNN classifier

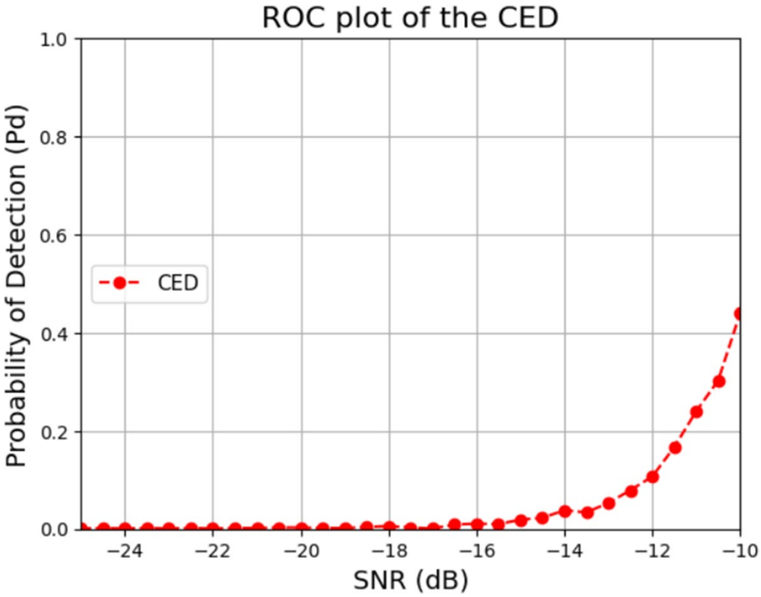


Fig. 8 ROC plot of SNR (dB) versus Pd for ED

the CED system to improve its performance in the lower SNR ranges, but still its performance has only gone up by 18.4% [2].

Figure 10 shows the performance of the KNN model, against the performance of CED and ED + EN. Upon observation, it can be seen that the model has performed far better than the existing systems. Moreover, it has performed very well throughout the entire range of SNR values. This can be confirmed by also observing that the false positives and the false negatives have been reduced down to nearly 2.5% for the KNN model as shown in the confusion matrix in Fig. 7. The reduction in these values indicates a drop in the false alarm's probability in the system proposed thereby showing that the system in the proposal functions better. When comparing the result with [1], the model is performed much better at low sample size of 64,000. Their performance is 83.9%, while the proposed model gives out a 94.5% accuracy. The novelty of the model is that we have built the model with 14 features, which are properties of the signal.

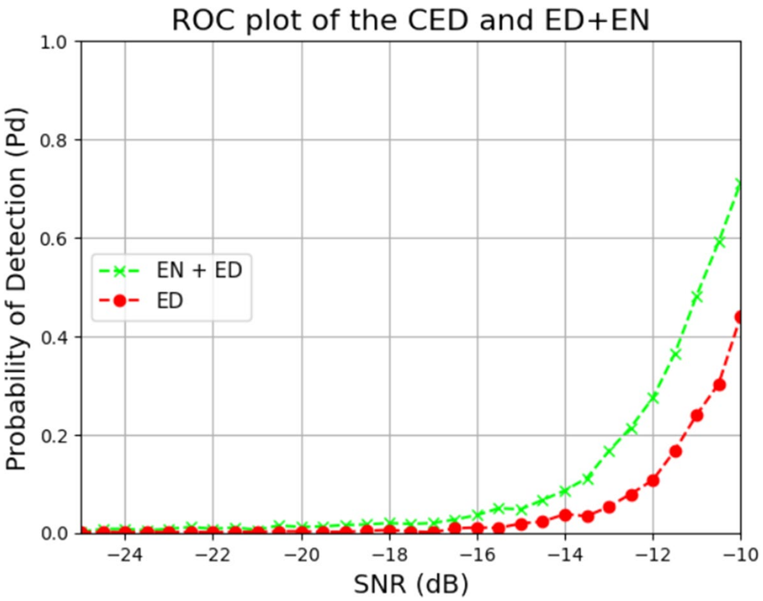


Fig. 9 ROC plot of SNR (dB) versus Pd for CED and ED + EN

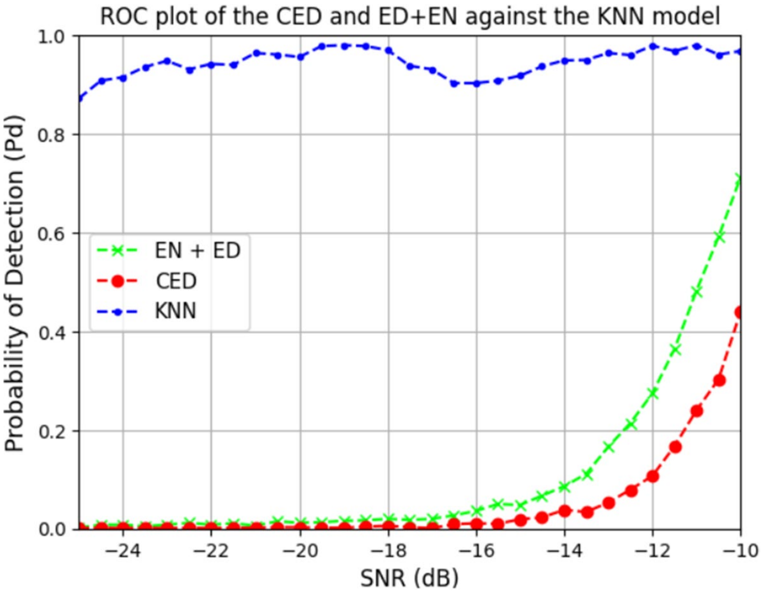


Fig. 10 ROC plot of SNR (dB) vs Pd for ED and ED + EN against KNN model

So far, we have been observing the SNR vs Probability of detection graph to conduct the performance analysis of the CED, ED + EN and the proposed KNN model. Another approach by which we study the performance of the conventional and the proposed model is by plotting the false alarm probability vs. the detection probability graph.

Figure 11 compares the operating characteristics of the KNN model against the operating characteristics of the conventional and ED + EN model at a higher SNR value of -10dB. At this value, all the systems perform satisfactorily well with the KNN

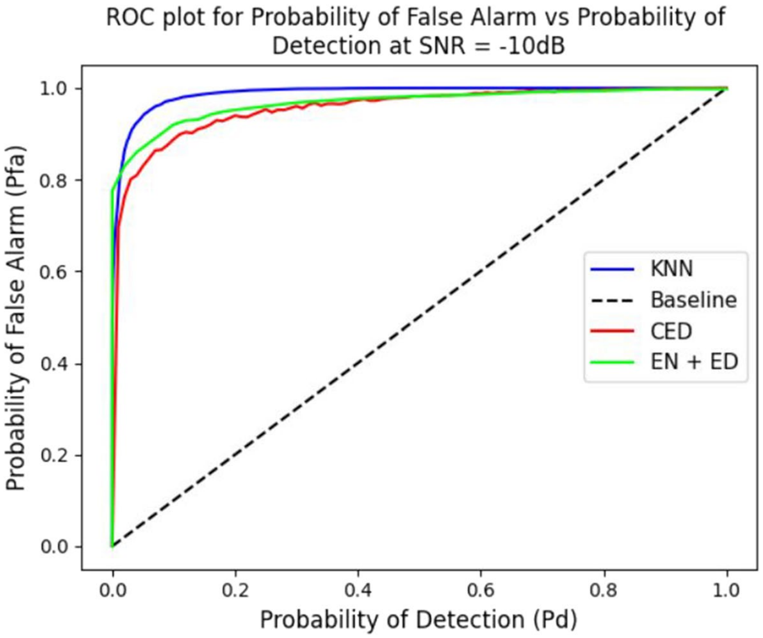


Fig. 11 ROC plot of Pfa vs Pd for all systems (CED, EN + ED and KNN) at SNR = − 10 dB

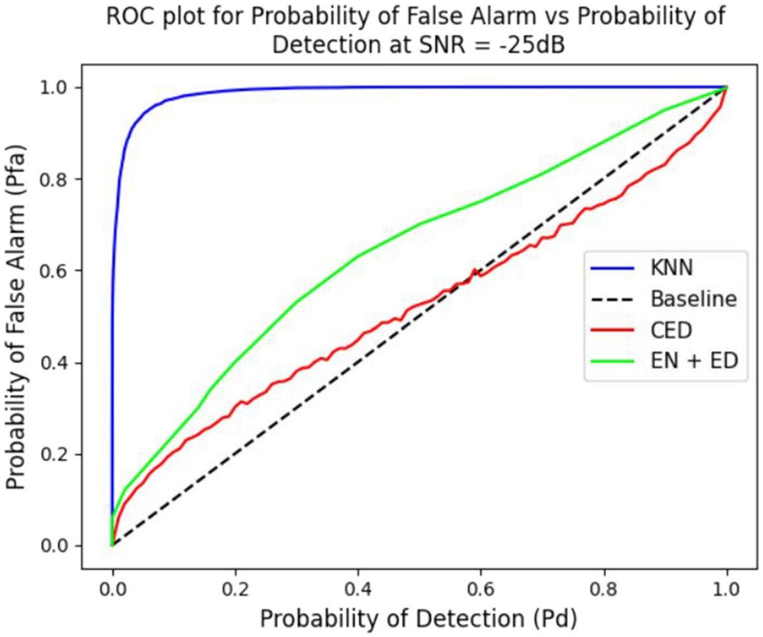


Fig. 12 ROC plot of Pfa vs Pd for all systems (CED, EN + ED and KNN) at SNR = − 25 dB

model having a slight edge over the others. We can also infer that the conventional models seem to perform well under the higher values of SNR (dB).

However, at lower SNR values, the performance of both the conventional and the entropy systems drops drastically, while the KNN model clearly outperforms the former models which becomes evident from Fig. 12. Hence, we can conclude from

Figs. 10, 11 and 12 that the proposed system shows high levels of detection accuracy and is an effective system for spectrum sensing in CRN at extremely low SNR ranges.

This paper contributes towards the improvement of the detection system by incorporating machine learning to detect PU signals in the environment. Although, having a minor trade-off in the complexity of the proposed model in terms of computation power, it is negligible when compared to the improved detection precision of the PU signal.

The processing time for the 3 simulations was performed on the out of sample dataset as shown in Table 3. CED and ED + EN is much faster as it only requires to calculate energy of signal and then a comparison logic to find if PU is present or not. In case of KNN model, the model has to be trained. The training for 64,000 samples took 20.56 min. Once the model has been trained, each prediction takes similar time to predict the detections. This difference in detection for the proposed KNN model does not outweigh the performance of the model. The model needs to be trained only once after which the model can be saved for practical purposes.

6 Conclusion

This paper provides a system that outperforms the conventional energy detector (CED) and its optimized counterparts (entropy method), especially in the low ranges of SNR values. This system is developed by incorporating ML, wherein many different models were trained to improve the performance. Out of them, the KNN algorithm, which was the best-performing model, was incorporated into the CED system [12]. The system detects the presence of PU signal with high precision both in the lower and higher ranges of SNR. This newly proposed system takes in different properties of the signal as features and learns to distinguish these signals between a PU signal and a noise signal [18]. The KNN algorithm is chosen for the research as it is widely known for its simplicity and adaptability. However, it comes under the lazy learning category of algorithms. It stores the whole dataset during the training stage, thereby increasing the training and prediction time and consuming lots of memory. Moreover, when implemented for real-world purposes, the KNN model must be trained individually to understand the specific noise characteristics for each environment. An efficient method to solve this concern would be to use transfer learning using a base model. It will be evaluated in the future works of this study. In the ever-so changing world, dealing with the dynamic nature of the wireless channel, which constantly changes in practice, poses a significant challenge in wireless communications [27]. Machine learning techniques that this research proposed itself are a good start into finding intelligent solutions for this challenge. There are various other capabilities of machine learning like transfer learning (as mentioned before), dynamic channel modelling, deep learning and even generative machine learning, that could enhance wireless communication in much efficient way even as the

Table 3 Simulation time of various detection methods over an out-of-sample dataset of 30,000 samples (s)

	CED	ED + EN	KNN
Training	–	–	1233.6
Calculation/prediction/detection	5.42	5.89	8.28

communication mode keep evolving. These approaches enable systems to make intelligent decisions in real-time, enhancing the reliability and performance of wireless communication systems in dynamic environments.

Abbreviations

CR	Cognitive radio
IOT	Internet of Things
PU	Primary user
SNR	Signal-to-noise ratio
ML	Machine learning
KNN	K-nearest neighbours
ED	Energy detection
CED	Conventional energy detection
AWGN	Additive White Gaussian Noise
SU	Secondary user
BPF	Band pass filter
PDF	Probability density function
ROC–AUC	Receiver operator characteristics-area under a curve
CRN	Cognitive radio network

Acknowledgements
Not applicable.

Author contributions
Conceptualization and Methodology done by ASSM and JFA. Original draft preparation, software and validation done by TV, DS, and SRD. All authors have read and agreed to the published version of the manuscript.

Funding
Not applicable.

Availability of data and materials
Not applicable.

Declarations

Ethical approval and consent to participate
Not applicable.

Consent for publication
I, the undersigned, give my consent for the publication of identifiable details, which can include photograph(s) and/or videos and/or case history and/or details within the text ("Material") to be published in the above Journal and Article.

Competing interests
The authors declare no competing interests.

Received: 17 November 2023 Accepted: 20 February 2024
Published online: 27 February 2024

References

1. T.O. Fajemilehin, A. Yahya, K. Langat, Improving energy detection in cognitive radio systems using machine learning. J. Commun. **15**(1), 74–80 (2020)
2. Usman MB, Singh RS, Mishra S, Rathee SA, Improving spectrum sensing for cognitive radio network using the energy detection with entropy method. J. Electric. Comput. Eng. 2022, 2656797 (2022). https://doi.org/10.1155/2022/2656797
3. K. Kockaya, I. Develi, Spectrum sensing in cognitive radio networks: threshold optimization and analysis. EURASIP J. Wirel. Commun. Netw. **2020**, 255 (2020)
4. M. Ranjeeth, S. Anuradha, Performance of fading channels on energy detection based spectrum sensing. Procedia Mater. Sci. **10**, 361–370 (2015)
5. A. Ranjan, B. Singh, Design and analysis of spectrum sensing in cognitive radio based on energy detection, in *2016 International Conference on Signal and Information Processing (IConSIP)*, Nanded, India (2016)
6. S. Nallagonda, S. Suraparaju, S.D. Roy, S. Kundu, Performance of energy detection based spectrum sensing in fading channels. In: *2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011)*, Allahabad, India (2011)
7. O. Kherif, Y. Benmahamed, M. Teguar, A. Boubakeur, S.S.M. Ghoneim, Accuracy improvement of power transformer faults diagnostic using KNN classifier with decision tree principle. IEEE Access **9**, 81693–81701 (2021)

8. R. Ghawi, J. Pfeffer, Efficient hyperparameter tuning with grid search for text categorization using KNN approach with BM25 similarity. Open Comput Sci **9**(1), 160–180 (2019)
9. P. Verma, B. Singh, Threshold optimization in energy detection scheme for maximizing the spectrum utilization. Procedia Comput. Sci. **93**, 191–198 (2016)
10. A.D. Sahithi, E.L. Priya, N.L. Pratap, Analysis of energy detection spectrum sensing technique in cognitive radio. Int. J. Sci. Technol. Res. **9**(1), 1772–1778 (2020)
11. A.D. Sahithi, E.L. Priya, N.L. Pratap, Analysis of energy detection spectrum sensing technique in cognitive radio. Int. J. Sci. Technol. Res. **9**(01), 1772–1778 (2020)
12. M. Abdulsattar, Energy detection technique for spectrum sensing in cognitive radio: a survey. Int. J. Comput. Netw. Commun. **4**, 223–242 (2012). https://doi.org/10.5121/ijcnc.2012.4514
13. S. Force, Spectrum policy task force report. Federal Communications Commission (FCC), vol. 135, no. 2 Washington DC (2002)
14. E. Cadena Muñoz, L.F. Pedraza Martínez, C.A. Hernandez, Renyi entropy-based Spectrum sensing in mobile cognitive radio networks using software defined radio. Entropy **22**(6), 626 (2020)
15. G. Mahendru, A. Shukla, P. Banerjee, A novel mathematical model for energy detection based spectrum sensing in cognitive radio networks. Wirel. Pers. Commun. **110**(3), 1237–1249 (2019)
16. S.V. Nagaraj, Entropy-based spectrum sensing in cognitive radio. Signal Process. **89**(2), 174–180 (2009)
17. J. Nikonowicz, P. Kubczak, Ł. Matuszewski, Hybrid detection based on energy and entropy analysis as a novel approach for spectrum sensing, in *Proceedings of the 2016 International Conference on Signals and Electronic Systems (Icses)*, Krakow, Poland (2016)
18. S. Chaudari, *Spectrum Sensing for Cognitive Radios: Algorithms, Performance, and Limitations* (Aalto Univ. School of Electrical Engineering, Helsinki, 2012)
19. K.G. Ajay, U.G. Raju, P. Aravind, D. Sushma, Intelligent wireless communication system of cognitive radio. Int. J. Emerg. Sci. Eng. **1**(5), 78–82 (2013)
20. M. Lakshmi, R. Saravanan, R. Muthaiah, Energy detection based spectrum sensing for cognitive. Int. J. Eng. Technol. **5**(2), 963–967 (2013)
21. P.K. Verma, S. Taluja, R.L. Dua, Performance analysis of energy detection, matched filter detection & cyclostationary feature detection spectrum sensing techniques. Int. J. Comput. Eng. Res. **2**(5), 2250–3005 (2012)
22. R.F. Ustok, *Spectrum Sensing Techniques for Cognitive Radio Systems with Multiple Antennas* (Izmir Institute of Technology, Urla, 2010)
23. F.D.C. Paisana, *Spectrum Sensing Algorithms for Cognitive Radio Networks* (Universidade Tecnica de Lisboa, Lisbon, 2012)
24. Y. Chen, Improved energy detector for random signals in gaussian noise. IEEE Trans. Wirel. Commun. **9**(2), 558–563 (2010)
25. K. Kim, Y. Xin, S. Rangarajan, Energy detection based spectrum sensing for cognitive radio: an experimental study, in: *Proceeding of Global Telecommunications Conference (GLOBECOM 2010)* (IEEE, 2010), pp. 1–5
26. The Effect of Sampling Rate And Signal-to-Noise Ratio on Methods for the Automated Determination Of Sustained Maximum Amplitudes in Vibration Signals, A Master's Thesis Presented to The Academic Faculty by Nathaniel DeVol Georgia Institute of Technology (2021)
27. P. Pandya, A. Durvesh, N. Parekh, Energy detection based spectrum sensing for cognitive radio network, in *Proceeding of Fifth IEEE International Conference on Communication Systems and Network Technologies (CSNT), 2015* (2015), pp. 201–206

Publisher's Note