

Article

Spectrum Sensing in Cognitive Radio Internet of Things Networks: A Comparative Analysis of Machine and Deep Learning Techniques

Akeem Abimbola Raji *  and Thomas Otieno Olwal 

Department of Electrical Engineering, F'SATI, Tshwane University of Technology, Pretoria 0001, South Africa

* Correspondence: rajiakeemabimbola@gmail.com or rajiaa@tut.ac.za

Abstract

The proliferation of data-intensive IoT applications has created unprecedented demand for wireless spectrum, necessitating more efficient bandwidth management. Spectrum sensing allows unlicensed secondary users to dynamically access idle channels assigned to primary users. However, traditional sensing techniques are hindered by their sensitivity to noise and reliance on prior knowledge of primary user signals. This limitation has propelled research into machine learning (ML) and deep learning (DL) solutions, which operate without such constraints. This study presents a comprehensive performance assessment of prominent ML models: random forest (RF), K-nearest neighbor (KNN), and support vector machine (SVM) against DL architectures, namely a convolutional neural network (CNN) and an Autoencoder. Evaluated using a robust suite of metrics (probability of detection, false alarm, missed detection, accuracy, and F1-score), the results reveal the clear and consistent superiority of RF. Notably, RF achieved a probability of detection of 95.7%, accuracy of 97.17%, and an F1-score of 96.93%, while maintaining excellent performance in low signal-to-noise ratio (SNR) conditions, even surpassing existing hybrid DL models. These findings underscore RF's exceptional noise resilience and establish it as an ideal, high-performance candidate for practical spectrum sensing in wireless networks.

Keywords: cognitive radio; deep learning; internet of things; machine learning; spectrum sensing

1. Introduction

The evolutionary development of mobile communication over the years has given renewed impetus to the development of a variety of Internet of Things (IoT) devices. IoT plays a prominent function in many sectors, including transportation, agriculture, health-care, sports, banking, and security [1–7]. The demand for wireless spectrum has surged drastically due to rapid growth in IoT, cloud computing, mobile subscribers, and smart and android phones [8–10], exacerbating spectrum scarcity. This scarcity has heightened interest in spectrum sensing within the cognitive radio (CR)-enabled IoT network, which aims to identify unused spectrum bands for opportunistic access. Conventional spectrum assignment policy grants licensed or primary users (PUs) exclusive rights to a specific frequency band. Through spectrum sensing, unused spectral resources within frequency bands are identified and allocated to IoT devices operating as opportunistic users. This approach facilitates and strengthens the growth of IoT communications. There are different approaches to spectrum sensing; the most preferred, due to robustness to shadowing and fading, is the cooperative spectrum sensing [11–13].



Academic Editor: Minseok Kim

Received: 21 December 2025

Revised: 25 January 2026

Accepted: 29 January 2026

Published: 6 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Cooperative spectrum sensing is a technique in wireless communications, wherein multiple cognitive radio devices collaborate to identify the vacant PUs in a given spectrum band. This approach mitigates the limitations of individual sensing by leveraging the spatial diversity and collective intelligence of multiple sensors. Cooperative detection methods are broadly categorized into hard and soft decision fusion schemes. In the first scheme, a subset of opportunistic (secondary) users performs local sensing of PUs and forwards their results to a fusion center [14–20]. The fusion center then communicates the decision back to all secondary users (SUs). The advantage of this method is the utilization of fewer communication resources for local sensing [12]. The latter category utilizes all the opportunistic users to scan the primary network for local information and transfer the results to the fusion center. It consumes more communication resources due to the number of users that are involved in local sensing. The soft decision fusion scheme achieves higher sensing gain than the hard decision fusion scheme, as it leverages complete statistical information from each observation to maximize the probability of correct detection for a given false alarm rate. The traditional methods of spectrum sensing include energy detection (ED) [21,22], matched filter [23,24], cyclostationary feature detection [25,26], Eigen value detection [27,28], and graph theory [29]. ED offers an easy and fast way of determining unused spectrum in a communication network. This approach experiences performance degradation under wireless channel noise conditions, and is further affected by multipath fading and terrestrial disturbances [25,30]. When the information about the transmitted signal from the licensed user is available, matched filter is regarded as the optimum technique for spectrum detection [11]. The method necessitates detailed prior knowledge of PU features for successful deployment. Prior knowledge of parameters such as packet size, modulation type, and bandwidth is required before implementation. Cyclostationary feature detection delivers excellent spectrum detection results, but it is computationally demanding [31]. Graph theory provides a powerful mathematical tool for modeling and solving spectrum detection problems in CR networks. It enables cooperative, distributed, and intelligent spectrum sensing by representing relationships between CRs and their environments [29].

To enhance spectrum utilization, numerous researchers have explored the application of machine learning (ML) and deep learning (DL) to mitigate spectrum scarcity in wireless communications. The selection of ML and DL algorithms for spectrum sensing is influenced by several factors, including data volume, structure, and availability; resource constraints (computational, memory, and energy); signal environment; model complexity; and operational requirements such as adaptability and deployment flexibility. Research on ML and DL applications for spectrum sensing includes the work presented in [32], where the authors introduce an artificial neural network (ANN) with a single hidden layer and output layer. This ANN architecture reportedly surpasses both conventional ED and improved ED techniques. The study in [33] employs K-means clustering to detect free spectrum, where it is seen that the model outperforms ED and OR fusion scheme. In [34], a novel technique is proposed, where a Bayesian learning algorithm optimizes the training parameters of a random forest (RF) model to enhance its predictive performance. The proposed technique outperforms established baseline methods such as K-nearest neighbor (KNN), support vector machine (SVM), and Gaussian mixture model across key sensing metrics. The authors in [35] consider LeNet, AlexNet, and VGG-16 for detecting vacant spectrum in a radio network. The study shows that VGG-16 achieves the highest detection rate and the lowest false alarm rate. The contribution of the authors in [36] demonstrates the robustness of the denoising autoencoder to detect unused spectrum in integrated sensing and communication (ISAC) networks. The study in [37] considers variational, deep, and long short-term memory (LSTM) autoencoders for wireless signal detection. The authors demonstrate that

the deep autoencoder achieves the shortest sensing time and the highest recall, precision, and F1-score. The authors in [38] introduce a novel technique that uses ANN to boost the performance of ED and matched filter schemes. The results show that the ANN-enhanced ED and matched filter techniques are robust to noise in wireless channels and achieve superior spectrum detection performance compared to their standalone counterparts. The study in [39] demonstrates the robustness of the recurrent neural network (RNN) and the convolutional neural network (CNN) against channel noise. It is revealed that the two DL techniques surpass the matched filter and ED models, within a signal-to-noise ratio (SNR) range of -5 dB to 0 dB. The authors in [40] demonstrate that LSTM outperforms both cyclostationary feature detection and ED methods, achieving the highest probability of detection and the lowest probability of false alarm within the SNR range of -25 dB to -10 dB. In [41], the authors propose a hybrid CNN–RNN, enhanced by transfer learning for vacant spectrum detection. The hybrid model demonstrates superior detection probability. According to [42], combining CNN and LSTM into a hybrid model yields a higher detection probability than using DNN, LSTM, or CNN models individually. The authors in [43] also propose combination of CNN and LSTM for spectrum detection, with the resulting model achieving a higher detection probability across noisy and noise-free environments. In [44], the authors propose a DLSenseNet that combines the features of CNN and LSTM for detecting vacant spectrum bands. It achieves a higher detection rate and a lower false alarm rate than the convolution long short-term deep neural network (CLDNN), LeNet, LSTM, DetectNet, residual network (ResNet), and Inception. The summary of these approaches is presented in Table 1.

Table 1. Existing ML and DL spectrum detection approaches.

Source	Approach	Dataset	Channel	Metrics	Findings
[32]	ANN	Experimental measurement	FM channel, E-GSM, DCS, and UHF television	Probability of detection and false alarm probability	The study demonstrates that ANN-based detector achieves a higher probability of detection and a lower false alarm rate than the traditional energy detector across all evaluated scenarios
[33]	K-means clustering	Band pass shift keying (BPSK) signal	$\alpha - K - \mu$ fading channel	Probability of detection	K-means clustering is shown to surpass ED and OR fusion scheme
[34]	Hybrid RF and Bayesian learning technique	Monte Carlo simulation	Rayleigh fading channel	Probability of detection and accuracy	The hybrid technique is shown to outperform SVM, KNN, and GMM. However, the authors fail to compare the performance of the hybrid model with traditional detection techniques
[35]	AlexNet, LeNet, and VGG-16	Orthogonal frequency division multiplexing (OFDM) signal	Rayleigh fading channel	Probability of detection and computational time	VGG-16 achieves the highest detection probability and lowest false alarm. In addition, all the CNN techniques produce better sensing results than traditional models
[36]	Autoencoder	OFDM	Rayleigh fading channel	Accuracy	The study demonstrates the robustness of denoising autoencoder to noise and adaptation to stochastic wireless channel. Comparison with traditional techniques is not reported by the authors

Table 1. Cont.

Source	Approach	Dataset	Channel	Metrics	Findings
[37]	Variational, deep and LSTM autoencoders	16-length quadrature amplitude modulation (QAM) signal, 64-QAM, and 256-QAM signals	Wireless fidelity and long term evolution signals	Precision, recall, and F1-score	Deep autoencoder achieves faster detection of spectrum vacancies. However, a comparative analysis with established techniques such as ED is not provided
[38]	ANN	256-QAM signal	Rayleigh fading channel	Probabilities of detection and false alarm, capacity, and bit error rate	ANN-enhanced ED and matched filtering methods outperform standalone counterparts
[39]	CNN and RNN	512-fast Fourier transform (FFT) signal	Rayleigh fading channel	Probabilities of detection and false alarm, capacity, and bit error rate	CNN and RNN models achieve superior detection accuracy, compared to traditional techniques like matched filter and energy detection
[40]	LSTM	Quadrature phase shift keying (QPSK), 16-phase shift keying (PSK), 4-QAM, and 16-QAM	Rayleigh fading channel	Probabilities of detection and false alarm	The authors establish the superiority of LSTM over cyclostationary and energy detection
[41]	CNN-RNN and transfer learning	64-QPSK, 128-QPSK, and QAM signals (64 and 128 sample lengths)	RadioML2016.10b datasets	Probabilities of detection and false alarm	The study achieves higher probability of detection than LSTM, CNN, and DNN. The authors do not compare the proposed technique with traditional ED method
[42]	CNN-LSTM	64-QPSK, QAM16, QAM 64, and BPSK	Independent and identically distributed channel	Probability of detection	The study demonstrates the superiority of hybrid CNN-LSTM architecture over ED and individual LSTM, DNN, and CNN models
[43]	CNN-LSTM	64-QPSK	Rayleigh fading channel	Probability of detection	The study demonstrates improved detection probability under noise and noise-free environments. A performance comparison between the proposed technique and energy detector is not provided by the authors
[44]	DLSenseNet	64-QPSK, 128-QPSK, and QAM signals (64 and 128 sample length)	RadioML2016.10b datasets	Probabilities of detection and false alarm	The proposed technique achieves better spectrum detection accuracy than CNN, LSTM, DetectNet, and LeNet models. The authors do not compare the proposed model with the traditional spectrum sensing schemes
This work	RF, KNN, SVM, CNN, Autoencoder, and ED	QPSK (64 sample length)	Rayleigh fading channel	Probabilities of detection, false alarm and missed detection, accuracy, F1-score, and training time	The current study demonstrates the superiority of RF over ML/DL counterparts and the conventional ED method, particularly in challenging low-SNR scenarios

While Table 1 shows that ML and DL have been applied to spectrum sensing both independently and in hybrid forms, a systematic comparison of their respective strengths and limitations for detecting vacant spectrum is absent from the literature. In addition,

the current literature lacks a principled framework for choosing between the modeling paradigms of ML and DL in spectrum sensing. To bridge these gaps, we present a comparative study that evaluates both approaches not only on detection accuracy but also on computational efficiency and robustness to channel non-idealities, offering a practical decision matrix for network designers.

The study advances spectrum sensing research through the following specific contributions:

- We present a direct performance comparison of three classical ML algorithms (KNN, SVM, RF) and two DL architectures (CNN, Autoencoder) against the traditional ED baseline.
- Our assessment employs a comprehensive suite of metrics, including standard probabilities of detection, false alarm, and missed detection alongside the more informative F1-score, which exposes trivial spectrum classification and quantifies the trade-offs between interference avoidance and spectral efficiency.
- Our empirical results identify RF as the most robust algorithm, outperforming both its ML/DL counterparts and the conventional ED method, particularly in challenging low-SNR scenarios.
- We contextualize RF's superiority by comparing it with state-of-the-art techniques from the literature and synthesize our results into practical performance guidelines for cognitive radio system designers.

The paper is structured as follows: Section 1 provides the background and motivation. Section 2 details the methodology. Results and discussion are presented in Section 3, and the conclusion is given in Section 4. The list of symbols and acronyms utilized throughout the paper is presented in Table 2.

Table 2. List of definition of symbols and acronyms.

Acronyms	Definition
ANN	Artificial neural network
AWGN	Additive white Gaussian noise
BPSK	Binary phase shift keying
CNN	Convolutional neural network
CPU	Central processing unit
DL	Deep learning
FFT	Fast Fourier transform
FM	Frequency modulation
GSM	Global system for mobile communication
IoT	Internet of Things
ISAC	Integrated sensing and communication
KNN	K-nearest neighbor
LTE	Long term evolution
LSTM	Long short-term memory
ML	Machine learning
OFDM	Orthogonal frequency division multiplexing
PU	Primary user
PSK	Phase shift keying

Table 2. *Cont.*

Acronyms	Definition
QAM	Quadrature amplitude modulation
QPSK	Quadrature phase shift keying
ReLU	Rectified linear unit
RF	Random forest
SU	Secondary user
SNR	Signal-to-noise-ratio
UHF	Ultra high frequency
A_0	PU is absent
A_1	PU is present
C	Regularization parameter
H_k	Wireless channel
n_estimators	Number of trees
n_neighbors	Number of nearest points
n_k	AWGN
k	Number of SU that participates in spectrum sensing
x	Transmitted signal from the PU
y_k	Received signal
$*$	Convolution operator
f	Output label
$h_n(\cdot)$	Prediction from nth decision tree
$\phi(\cdot)$	Indicator function
$\hat{y}(\cdot)$	Binary decision of RF model
ω	Weighted vector
b	Bias vector
W	Weighted matrix

2. Methodology

Figure 1 illustrates a cooperative CR network comprising a single PU and M SUs. We utilize K out of M SUs to scan the radio network for vacant spectrum. Each of the K SUs transfers the result of the scanning to the fusion center for decision. The signal received by the K SUs is expressed by Equation (1):

$$y_k = H_k x + n_k \quad (1)$$

where y_k is the received information, H_k is the wireless channel matrix, x is the transmitted signal from the PU, n_k is the additive white Gaussian noise (AWGN), and $k = 1, 2, \dots, K$ is the number of SUs that participates in spectrum sensing.

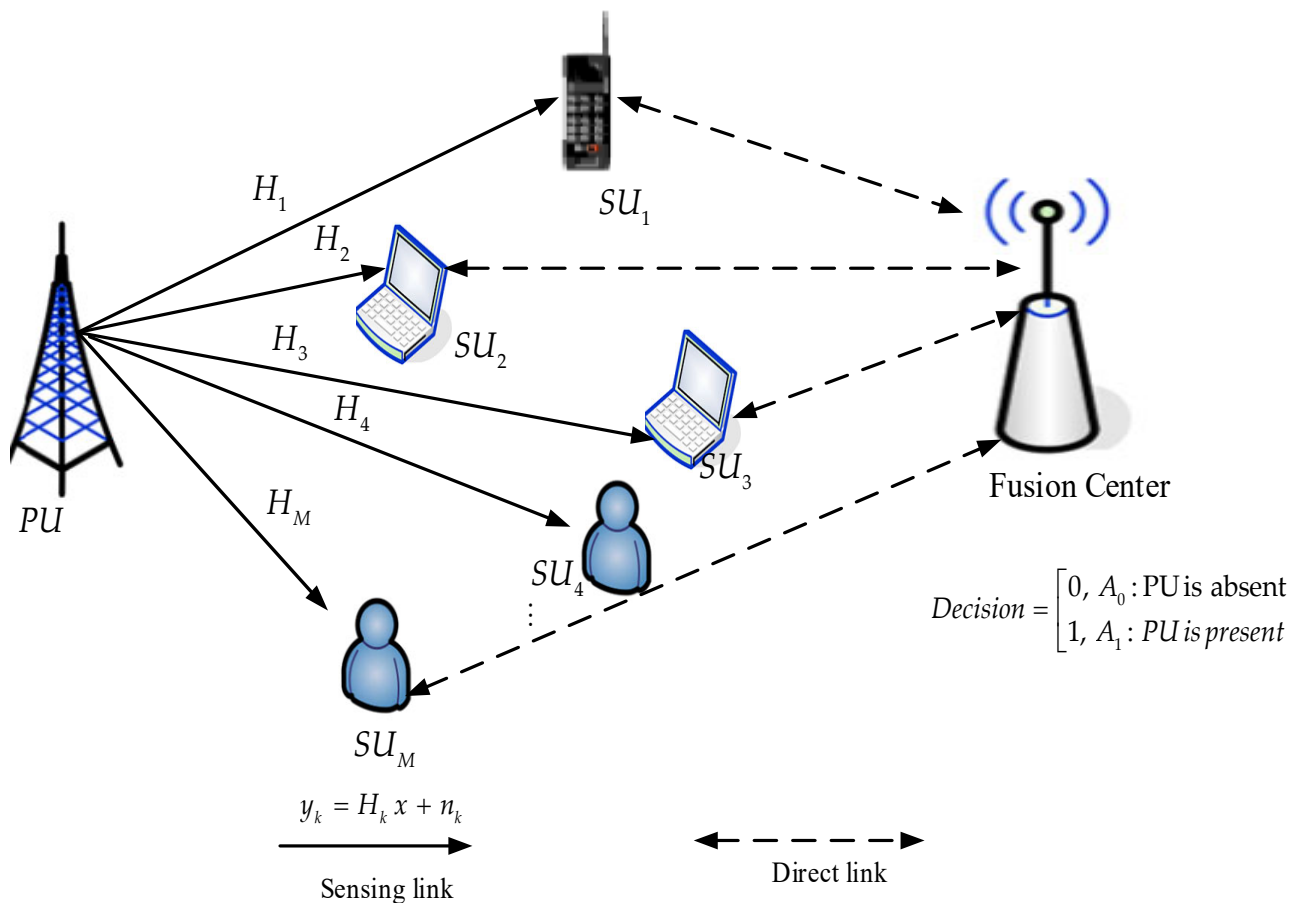


Figure 1. Cooperative sensing model with fusion center, PU, and M SUs.

The decision at the fusion center is expressed by Equation (2) [22,31]:

$$Decision = \begin{cases} 0, & A_0 : y_k = n_k \\ 1, & A_1 : y_k = H_k x + n_k \end{cases} \quad (2)$$

in which A_0 defines the null hypothesis, which represents the absence of the PU (spectrum unoccupied), while A_1 represents the decision that indicates the availability of the PU (spectrum occupied). We consider Equation (2) as classification problem, whose solution is sought using ML and DL techniques. Figure 2 illustrates the proposed ML and DL pipeline for spectrum sensing. The architecture follows a staged workflow comprising data preparation, model training, validation, and testing to ensure systematic development and evaluation.

Stage 1: Data input and signal modeling

In the first stage, a synthetically generated QPSK signal serves as the PU transmission. The dataset consists of 1000 symbols, with 300 samples per SNR level, simulated at a 1 GHz sampling frequency. Each sample is represented as a 64-length sequence, capturing temporal and modulation characteristics essential for sensing tasks.

State 2: Feature extraction and model training

The second stage involves feature extraction and model initialization. To enhance interpretability and performance, we extract a comprehensive set of physically meaningful features from the QPSK signal, including amplitude, phase, energy, spectral characteristics, spectral density, noise variance, peak-to-average power ratio (PAPR), and cyclostationary features. These features are used to train both ML and DL models, ensuring that learning is grounded in signal properties.

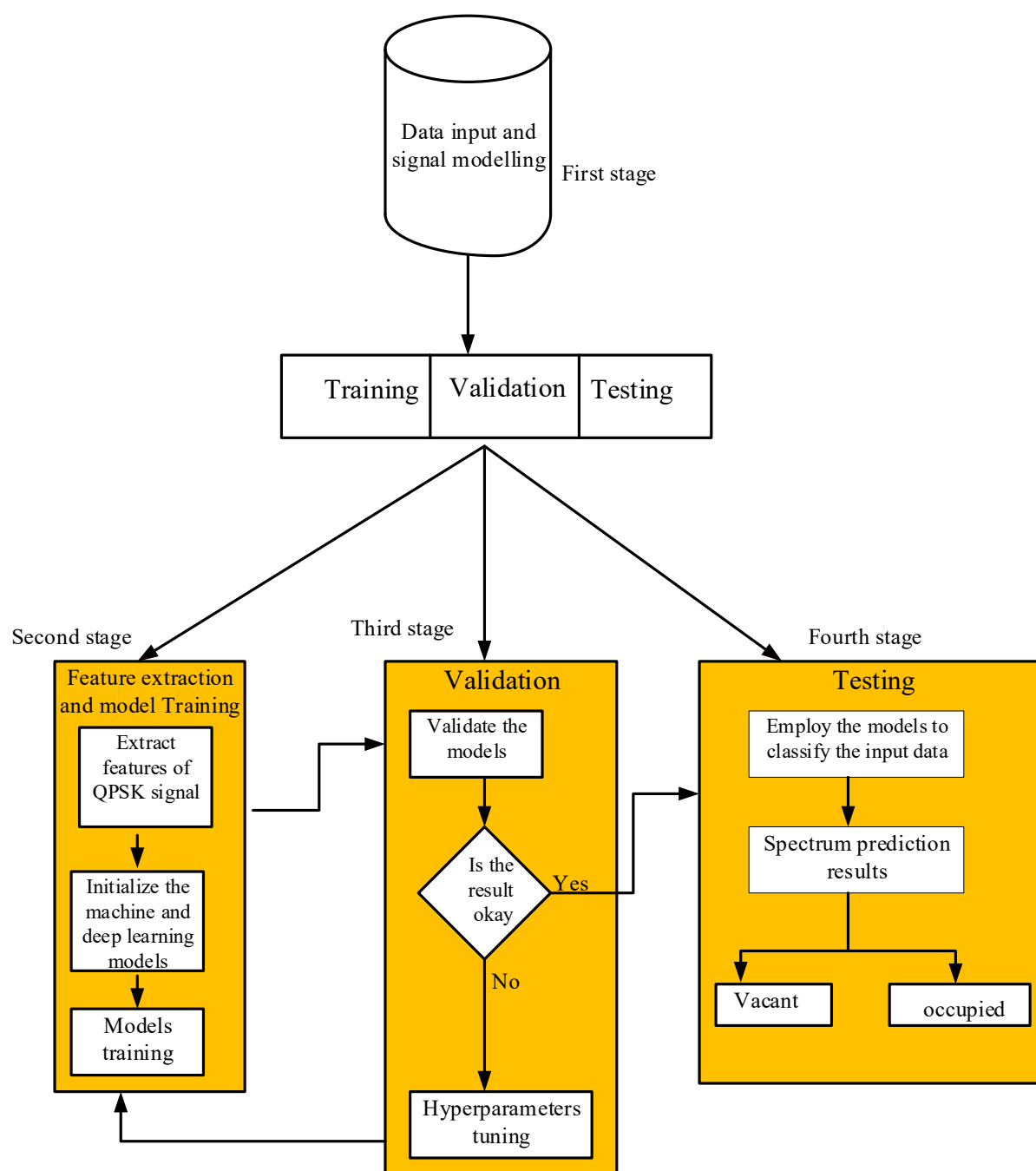


Figure 2. Architecture of ML and DL models for spectrum sensing.

Stage 3: Validation and hyperparameter tuning

The third stage validates the trained models using a separate validation set. Model performance is assessed based on detection probability, missed detection probability, false alarm probability, accuracy, and F1-score. If validation results are unsatisfactory, a hyperparameter tuning loop is triggered to optimize model parameters before retraining. This iterative process ensures robustness and prevents overfitting.

Stage 4: Testing and final classification

In the final stage, the optimized models are deployed on a training dataset to classify received signals as 'vacant' or 'occupied'. This end-to-end evaluation confirms the model's practical utility and generalization capability.

Stage 5: Dataset partitioning for evaluation

To ensure statistically sound evaluation, the dataset is split into 80% for training, 10% for validation, and 10% for testing. This split supports robust model development, while preserving unseen data for final performance assessment.

2.1. Machine Learning (ML) Models

Three ML models are employed to extract features from the PU for spectrum prediction. The models include RF, KNN, and SVM. KNN is chosen for its high accuracy, simplicity, and robust performance in low-noise channel conditions. SVM is considered due to its accuracy at low SNR, efficiency with limited training data, and capability to handle complex and non-linear classification. Finally, RF is selected for its resistance to overfitting, strong noise resilience, and ability to process complex, high-dimensional input data.

2.1.1. Random Forest (RF)

This is a supervised ML technique that combines predictive outputs of decision trees in a forest, where each tree is trained on a subspace of input data [45,46]. The output of RF is determined by the mean of the predictions from all the trees. The RF model is robust to overfitting and excels at handling complex, non-linear, high-dimensional, and noisy datasets [41]. Its strength derives from aggregating the prediction of N individual decision trees, each trained on a random subspace of data features, as defined by Equation (3):

$$\hat{y}(x) = \arg \max_f \sum_{n=1}^N \phi(h_n(a) = f) \quad (3)$$

in which $\hat{y}(x) \in \{0, 1\}$ is the binary decision of RF model, a is the input feature vector for a signal sample, $h_n(a)$ denotes the prediction from n^{th} decision tree, f denotes the output label, and $\phi(\bullet)$ denotes an indicator function that tallies the votes for each decision at the fusion center. The RF model is trained with 100 decision trees ($n_{\text{estimators}} = 100$), a maximum tree depth of 20 ($\text{max_depth} = 20$), and a fixed random seed of 42 ($\text{random_state} = 42$).

2.1.2. K-Nearest Neighbor (KNN)

The KNN algorithm classifies a new data point by first calculating its Euclidean distance to all training instances. It then identifies the K closest points in the training set. The performance of KNN is contingent on three key factors: the chosen distance metric, the value of K , and the underlying structure of the data space [41]. In this study, we train the KNN classifier using 5 neighbors ($n_{\text{neighbor}} = 5$), a uniform weighting scheme, and set $n_{\text{jobs}} = -1$. This configuration enables full parallelization across all CPU cores, significantly improving computational speed while maintaining equal weighting among neighboring points.

2.1.3. Support Vector Machine (SVM)

SVM handles non-linearity in training data. It utilizes radial basis function to project the training data into a high-dimensional space. The interest of SVM is to find a suitable hyperplane that classifies the training data into different groups. The SVM classifier is configured with a regularization parameter $C = 10$ to control the trade-off between margin width and classification error, a radial basis function (RBF) kernel, and a random state (random_state) of 42 for reproducibility. The optimized hyperparameters for training the ML models, as determined through cross validation, are summarized in Table 3.

Table 3. Parameters for training ML models, obtained through cross validation.

Algorithms	Parameters	Values
KNN	n_neighbors	5
	weights	uniform
	n_jobs	−1
RF	n_estimators	100
	max_depth	20
	random_state	42
	min_samples_leaf	5
SVM	kernel	radial basis function
	random_state	42
	regularization parameter (C)	10

2.2. Deep Learning (DL) Models

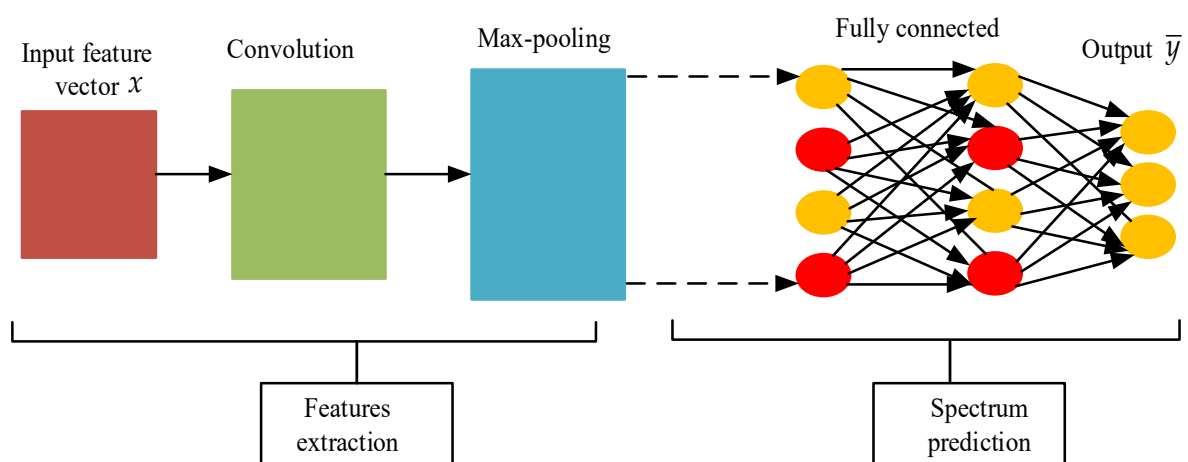
This work considers two DL models: an Autoencoder and a CNN. They are selected for their complementary strengths in feature extraction and signal adaptability. The CNN is employed for its proven ability to capture spatio-temporal features in input data and its robust performance in low-SNR regimes, where conventional methods typically fail [16,39]. The Autoencoder is utilized for its capacity for unsupervised automatic feature learning, which provides flexibility and robustness across diverse datasets.

2.2.1. Convolutional Neural Network (CNN)

The convolutional neural network (CNN), a supervised DL model, comprises convolutional, max-pooling, fully connected, and output layers, as illustrated in Figure 3. The convolutional layer employs multiple small-sized kernels (filters) to perform convolution, extracting salient features from the input data. Let x be an input feature vector and $W \in \mathcal{R}^{N \times N \times M} \times L$ be weighted matrix, where N represents the quantity of the filter, while M and L denote the input and output channel dimensions, respectively. The output of the convolutional layer is then given by Equation (4) [39]:

$$y_k = \varphi_c \left(\sum_{i=1}^k M(x * W_i) + b_i \right) \quad (4)$$

in which $*$ is the convolution operator, φ_c denotes activation function, and b_i denotes the bias unit of the k^{th} convolutional layer.

**Figure 3.** CNN architecture for spectrum sensing.

The max-pooling layer compresses the extracted feature map into small size, retaining the essential features in the extracted data. The output of max-pooling layer is given by Equation (5):

$$z^k = f^k(y^k, G) \quad (5)$$

where z^k is the output of the max-pooling operation, f^k represents the pooling operation function, and G is the magnitude of the pooling window.

The fully connected layers take the output of max-pooling layer and apply a set of neurons to make final classification and prediction, expressed by Equation (6):

$$Z_{FC} = \varphi_{FC} \left(\sum_{i=1}^k (W_{FC}^i * z_{FC}^i) + b_{FC}^i \right) \quad (6)$$

where Z_{FC} is the output of fully connected layers, φ_{FC} represents the activation function, z_{FC}^i is the input into the k^{th} fully connected layer, b_{FC}^i is the bias of each fully connected layer, and $*$ is the convolution operator. The output layer produces the spectrum sensing result for a given input data, which is described by Equation (7):

$$\bar{y} = \varphi_o((Z_{FC})) \quad (7)$$

in which $\bar{y}$ is the CNN output that represents the spectrum detection result and φ_o symbolizes the activation function at the output layer. The loss function for training the CNN is the sparse categorical cross entropy, which calculates the error based on the predicted probability of the true class. The weight and bias are updated using adaptive moment estimation (Adam) algorithm, where the learning rate is 0.001 and the batch size is 64. The Adam optimizer is chosen for training the CNN due to its computational efficiency, straightforward implementation, and low memory requirements [43]. The CNN architecture for spectrum sensing is presented in Table 4, and its corresponding training, validation, and testing procedure is outlined in Algorithm 1.

Table 4. Structure of CNN for spectrum sensing.

Layers	Dimension
Convolutional layer 1 (ConvD1)	(3 by 64)
Activation	Rectified linear unit (ReLU)
Padding	Same
Max-pooling	(2 by 2)
Convolutional layer 2 (ConvD2)	(3 by 128)
Activation	ReLU
Padding	Same
Convolutional layer 3 (ConvD3)	(3 by 256)
Activation	ReLU
Padding	Same
Dense	(128)
Activation	ReLU
Drop-out rate	0.5
Dense	(64)

Table 4. *Cont.*

Layers	Dimension
Activation	ReLU
Dropout	(0.3)
Dense	2
Activation	Softmax

Algorithm 1: Steps for training, validating, and testing a CNN model for spectrum sensing.

Input:

Training dataset Z_{train}
Ground truth label data

Output:

Spectrum sensing result $\bar{y}$

1: Data preprocessing

- 1.1 Normalize the training data to a range of $[-1, 1]$
- 1.2 Divide Z_{train} into training Z_{tr} , validation Z_{val} , and testing Z_{test} sets.

2: CNN architecture design

- 2.1 Define the CNN structure
 - Number of convolutional layers
 - Number of filters
 - Number of pooling layers
- 2.2 Activation functions
 - Hidden layer: $\sigma(x) = \text{ReLU}(x) = \max(0, x)$, x in an input feature vector
 - Output layer: Softmax

3: Model Training

- 3.1 Initialize weights W and bias b
- 3.2 For epoch $t = 1:E$, E is the total number of epochs
 - 3.2.1 Shuffle Z_{tr} to avoid biased training
 - 3.2.2 For each mini-batch $B \subset Z_{tr}$
 - a. Perform convolution
 - b. Apply ReLU
 - c. Perform max-pooling
 - d. Flatten output and pass through fully connected layers
 - e. Compute loss using sparse categorical cross entropy
 - f. Update weights using Adam optimizer

4: Model Evaluation

- 4.1 Evaluate the model on Z_{val} after each epoch
- 4.2 Compute performance metrics

5: Testing

- 5.1 Normalize Z_{test} using the same scaling as the training data
- 5.2 Forward propagate Z_{test} through the trained CNN model
- 5.3 Output spectrum prediction $\bar{y}$

6: Return $\bar{y}$ **2.2.2. Autoencoder**

Autoencoder is a DL model designed to reconstruct its original input from a compressed latent representation. It consists of three components (an encoder, a bottleneck, and a decoder) as shown in Figure 4. An encoder reduces the input data to a smaller dense size

by transforming the high-dimensional input into a lower-dimensional latent representation. This is achieved through successive layers of learned transformations, which employ the rectified linear unit (ReLU) activation function to introduce non-linearity. Given an input feature vector x , the output of the encoder is defined by Equation (8):

$$g = \varphi(wx + a) \quad (8)$$

where g is the encoder's output, φ is the activation function, ω represents the weight, and b denotes the bias vector. The encoder effectively learns the most salient features of the input data while discarding redundancies.

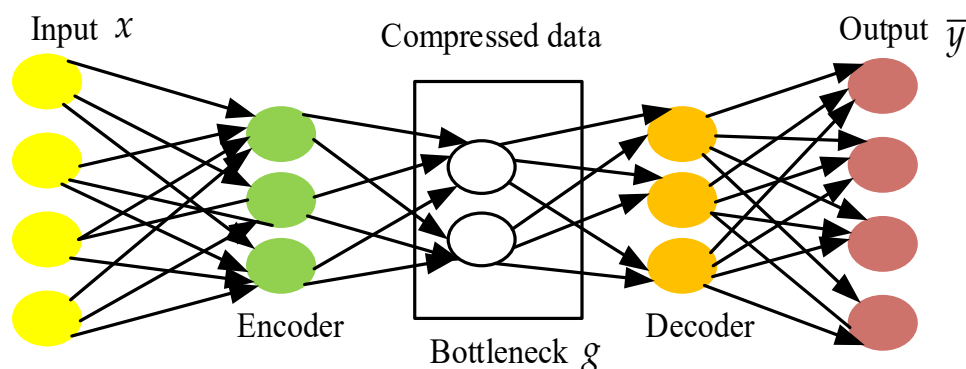


Figure 4. Autoencoder architecture for spectrum sensing.

The bottleneck contains latent space representation g , which corresponds to the compressed input data. The decoder network performs the inverse operation, mapping the latent representation back to a reconstructed version of the original input. The decoder learns to reconstruct the input from the compressed representation, by minimizing the reconstruction loss between the input data vector x and decoder's output $\bar{y}$, which is defined by Equation (9):

$$\bar{y} = \varphi'(w'g + b') \quad (9)$$

where φ' denotes the activation function at the decoder, w' and b' represent the weight and bias vector of the decoder, respectively. The Autoencoder predicts the spectrum by using sparse categorical cross entropy loss to quantify the dissimilarity between the predicted probability distribution over all the classes and the true class label. The model is trained using Adam optimizer with a learning rate of 0.001 and a batch size of 64. The Autoencoder network architecture is detailed in Table 5, and its training, validation, and testing procedure for spectrum sensing is outlined in Algorithm 2.

Table 5. Structure of Autoencoder for spectrum sensing.

Layers	Dimension
Encoder (dense layers)	128, 64, 32
Activation function	ReLU
Bottleneck (dense layer)	16
Activation function	ReLU
Dropout rate	0.3
Decoder (dense layer)	2
Activation	Softmax

Algorithm 2: Steps for training, validating, and testing Autoencoder for spectrum sensing.**Input:**Training dataset Z_{train} Ground truth label : $y \in (0,1)$ (0 : *spectrum vacant*, 1 : *spectrum occupied*)**Output:**Spectrum sensing result: $\bar{y}$ Reconstruction error : ε_{test} **1: Data preprocessing**

1.1 Normalize the training data to a range of [0, 1]

1.2 Partition Z_{train} into training Z_{tr} , validation Z_{val} , and testing Z_{test} sets.**2: Autoencoder architecture design**

2.1 Define layers of encoder and decoder

2.3 Specify activation functions for encoder, bottleneck, decoder, and output

3: Model Training3.1 Initialize training weights W and biases b 3.2 For epoch $t = 1:E$, E is the total number of epochs3.2.1 For each mini-batch Z_{tr}

a. Compute reconstruction error

b. Update the weights and biases using Adam optimizer

4: Model validation4.1 Evaluate on validation set Z_{val} after each epoch

-Compute reconstruction error

4.2 Perform spectrum sensing decision

4.3 Compute performance metrics

5: Testing5.1 Normalize Z_{test} using the same parameters as training

5.2 Encode and decode test samples

5.3 Compute reconstruction errors

5.4. Classify spectrum sensing decision

6: Return $\bar{y}$, ε_{test}

The optimized hyperparameters for training the CNN and Autoencoder models are detailed in Table 6.

Table 6. Summary of optimized parameters for training CNN and Autoencoder models.

Parameters	CNN	Autoencoder
Loss function	Sparse categorical cross entropy	Sparse categorical cross entropy
Number of epochs	30	30
Batch size	64	64
Learning rate	0.001	0.001
Activation function (hidden layers)	ReLU	ReLU
Activation function (output layer)	Softmax	Softmax
Optimizer	ADAM	ADAM

For benchmark comparison, the traditional ED technique is also evaluated within the same framework.

2.3. Energy Detection (ED)

ED is a conventional spectrum sensing technique that predicts spectrum occupancy by measuring the received signal energy. The decision metric is computed by averaging the squared magnitude of the received signal over a specified observation interval, as defined in Equation (10):

$$\text{Energy} = \frac{1}{M} \sum_{a=1}^M |y(a)|^2 \quad (10)$$

where $y(a)$ symbolizes the signal received by the SUs and $a = 1, 2, 3 \dots, M$ symbolizes the received signal length. The technique compares the energy to a predetermined threshold, where it is assumed that the spectrum band is not vacant if the energy level is above a threshold; otherwise, it is deemed vacant. The block diagram for predicting the spectrum through ED is depicted in Figure 5.

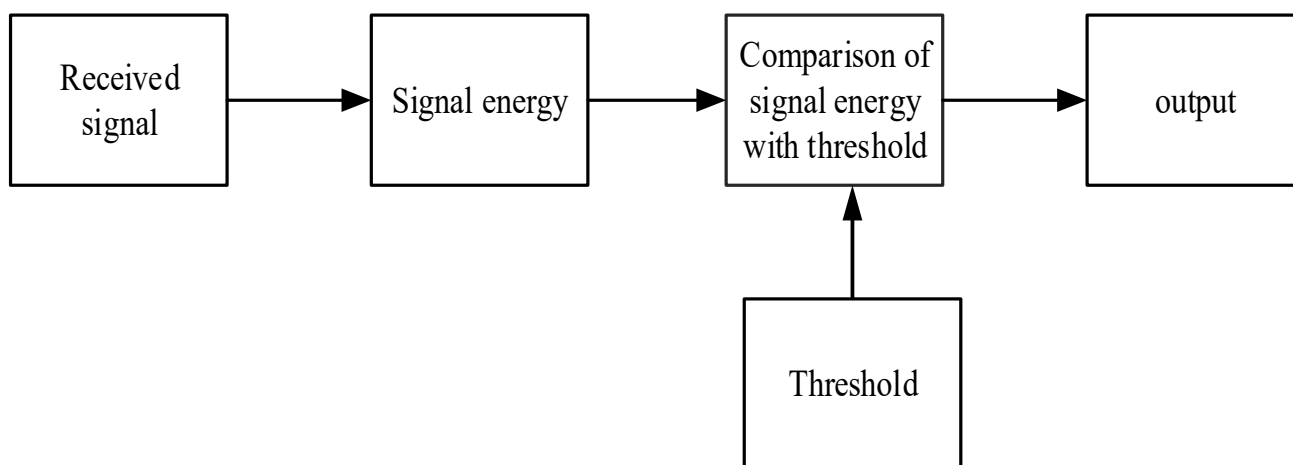


Figure 5. Spectrum sensing via energy detection.

The principal advantages of ED are its applicability to a wide variety of modulated signals [38] and its lower computational cost compared to other traditional techniques.

2.4. Evaluation Metrics

The performance of ML and DL techniques is examined through probabilities of detection, false alarm and missed detection, accuracy, and F1-score, which are the standard metrics for assessing spectrum sensing techniques.

2.4.1. Probability of Detection

This parameter measures the model's capability to correctly detect an active PU. It is defined by Equation (11):

$$\text{Probability of detection} = \frac{TP}{TP + FN} \quad (11)$$

where TPs (true positives) are instances which correctly identify PU as being active and FNs (false negatives) are instances where an active PU band is present in the channel but is not detected by the sensing algorithm.

2.4.2. Probability of False Alarm

This metric quantifies the rate at which an SU incorrectly assumes that the spectrum band is occupied when it is actually vacant. It is expressed as the ratio of false positive

detections to the total number of times the spectrum is truly vacant, which is given by Equation (12):

$$\text{Probability of false alarm} = \frac{FP}{FP + TN} \quad (12)$$

in which *FPs* (false positives) are incorrect ‘occupied’ decisions and *TNs* (true negatives) are correct ‘vacant’ decisions

2.4.3. Probability of Missed Detection

This metric quantifies the rate at which an SU incorrectly assumes that the spectrum band is vacant when it is actually occupied. It is expressed as the ratio of missed detections (false negatives) to the total number of times the PU is truly active, defined by Equation (13):

$$\text{Probability of missed detection} = \frac{FN}{TP + FN} \quad (13)$$

where *FNs* (false negatives) are incorrect ‘vacant’ decisions and *TPs* (true positives) are correct ‘occupied’ detections.

2.4.4. Accuracy

This is the ratio of the quantity of correct spectrum predictions to the quantity of spectrum predictions made. That is

$$\text{Accuracy} = \frac{\text{quantity of correct spectrum predictions}}{\text{quantity of spectrum predictions}} \quad (14)$$

2.4.5. F1-Score

The F1-score calculates the harmonic mean between recall and precision. It combines the two parameters into a single value. F1-score is more informative than accuracy because it gives a better assessment of the models’ effectiveness in correctly predicting the absence of the vacant spectrum.

3. Results

This section presents the simulation results demonstrating the performance of ML and DL models using detection probability, missed detection probability, false alarm probability, accuracy, and F1-score. The simulated CR network consists of PU transmitting a QPSK signal with a sample length of 64, which is sensed by the SUs. The channel between the PU and SUs is modeled as Rayleigh fading channel with identically and independently distributed (i.i.d.) gain. AWGN with zero mean and variance equal to the total noise power is applied, and the investigation spans an SNR regime of -20 dB to 20 dB, to characterize model robustness in noisy conditions. All models are developed in Python 3 within Google Colab environment, utilizing standard scientific computing and machine learning libraries. Classical ML models are built with scikit-learn and NumPy, while DL models are developed with Keras and Tensor flow. Data processing and ED analysis are performed using NumPy and SciPy. All simulations are executed on a local computing machine equipped with an Intel Core™ i7-6600U 64-bit processor, operating at 2.81 GHz with 8 GB of RAM. Figure 6 depicts the probability of detection as a function of SNR. The results demonstrate the superiority of the RF algorithm, which achieves the highest probability of detection across the entire SNR range. Notably, RF maintains this performance advantage within the critical low-SNR regime from -20 dB to -4 dB, highlighting its exceptional noise resilience and making it the preferred model for operation in noisy wireless channels. This finding corroborates prior work [47], which also established RF’s superiority over SVM and KNN for spectrum sensing. Furthermore, it is observed in Figure 6 that the DL models (CNN

and Autoencoder) outperform the traditional ED method. This aligns with findings in the literature [39], which reported CNN's superiority over conventional detection methods.

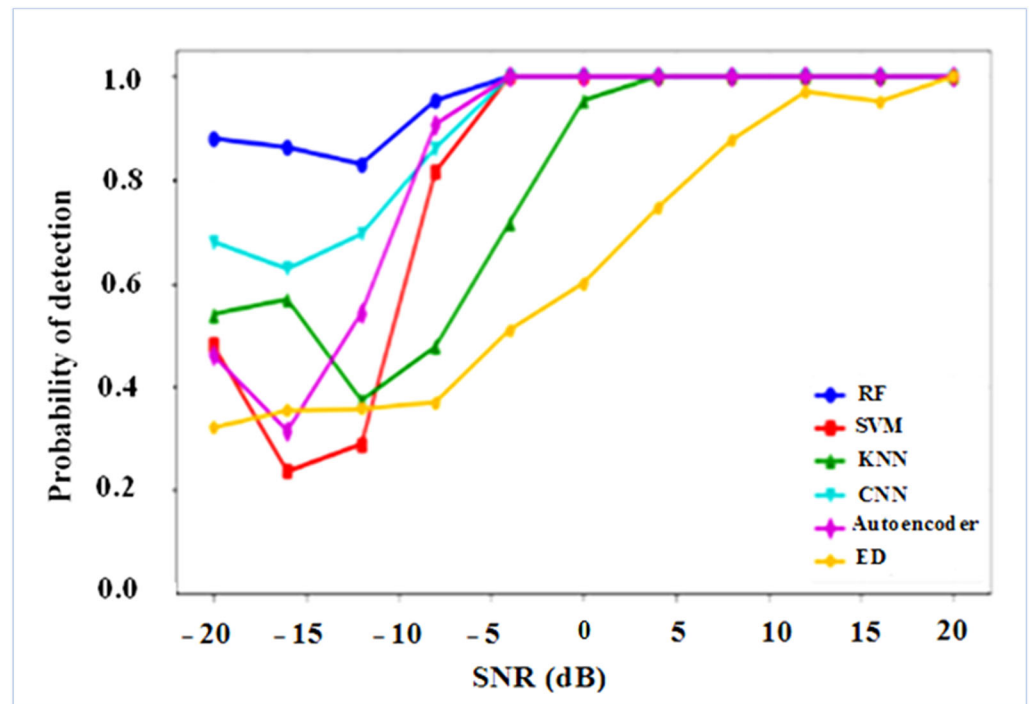


Figure 6. Probability of detection against SNR for RF, SVM, KNN, CNN, Autoencoder, and ED.

The average probability of detection across the SNR regime from -20 dB to 20 dB is quantified in Figure 7. RF attains the highest value at 0.957 (95.7%), followed by CNN (0.8967), the Autoencoder (0.8385), SVM (0.807), KNN (0.7845), and ED (0.6411). In relative terms, RF demonstrates performance gain of 6.3%, 12.38%, 16.23%, 18.02%, and 37.71% over these respective models.

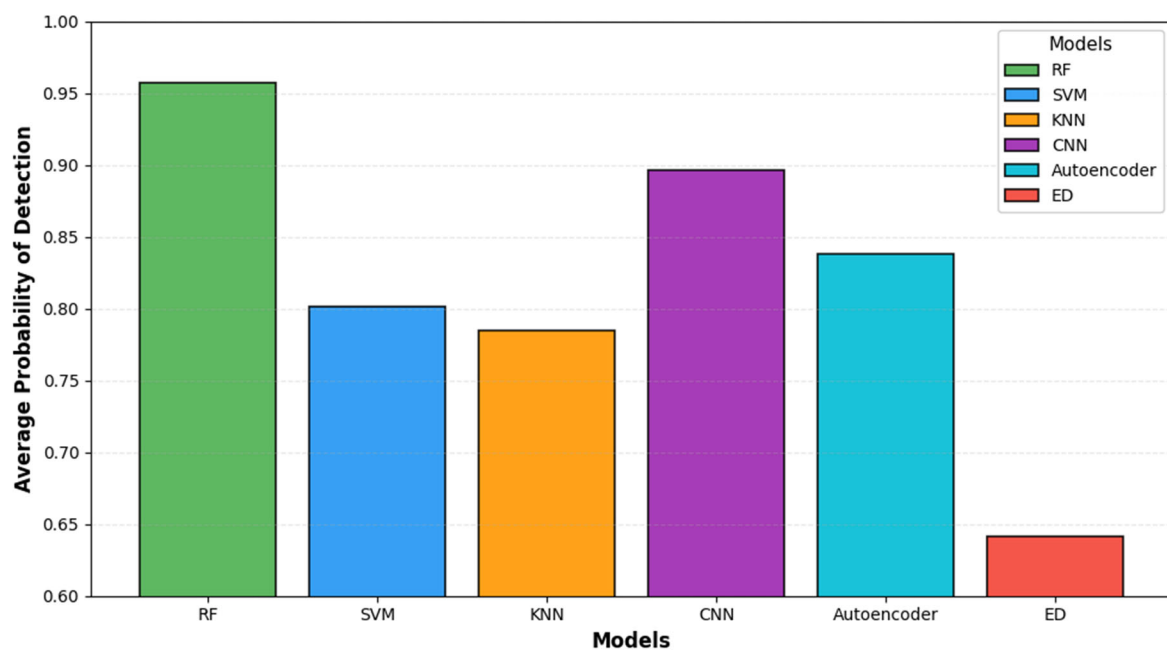


Figure 7. Comparison of average probability of detection of RF, SVM, KNN, CNN, Autoencoder, and ED.

The average probability of false alarm for each model is presented in Figure 8, where it is established that RF achieves the lowest probability of false alarm at 1.44%, followed by SVM at 3.19%. In contrast, traditional ED exhibits the highest false alarm rate at 10.81%, CNN (8.18%), Autoencoder (9.59%), KNN (7.38%), and SVM (3.19%).

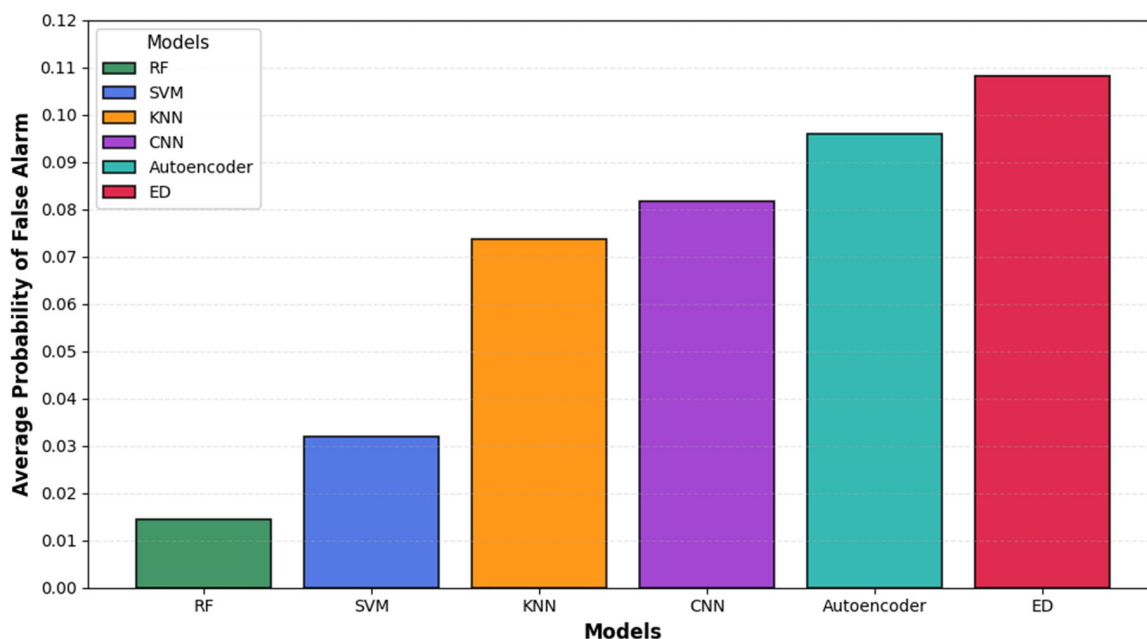


Figure 8. Comparison of probability of false alarm of RF, SVM, KNN, CNN, Autoencoder, and energy detection.

This indicates that RF most effectively minimizes incorrect declaration of PU activity, thereby offering superior interference control and channel access efficiency in the CR network. Figure 9 displays the average probability of missed detection for all the models, where it is observed that RF achieves the least value of probability of missed detection, indicating that it most reliably detects active PU. CNN and Autoencoder follow closely, while traditional ED demonstrates the highest missed detection rate.

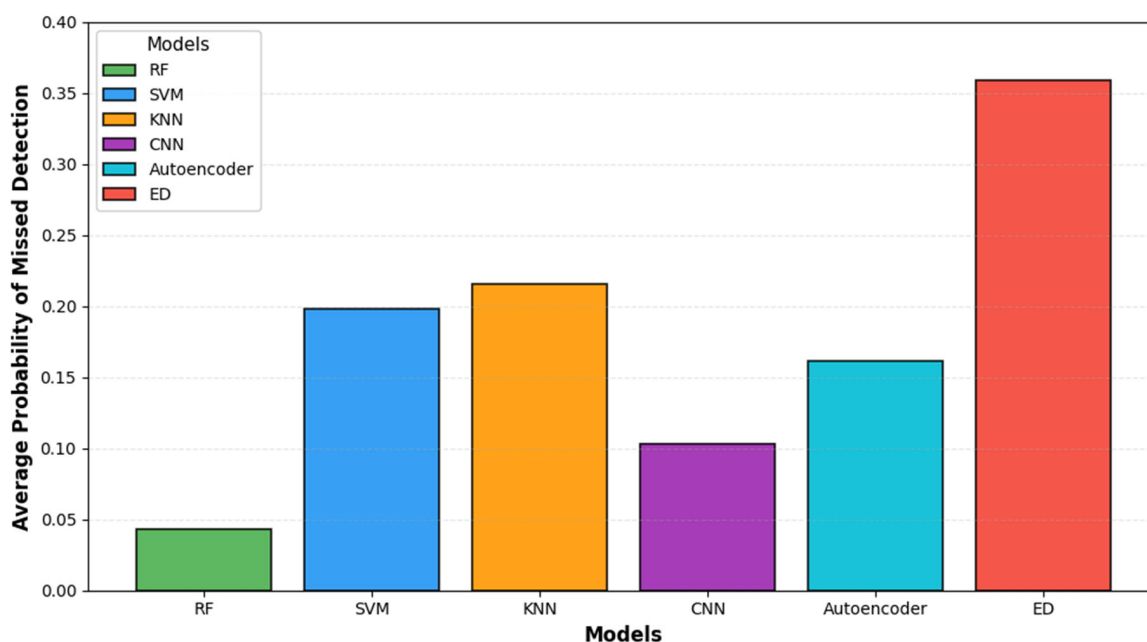


Figure 9. Comparison of probability of missed detection of RF, SVM, KNN, CNN, Autoencoder, and ED.

In addition, it is evident in Figure 9 that RF achieves the lowest probability of missed detection at 4.3%, significantly outperforming CNN (10.33%), the Autoencoder (16.15%), SVM (19.83%), KNN (21.55%), and ED (35.89%). This directly indicates that RF minimizes the loss of available spectral opportunities, while traditional ED incurs substantial spectral inefficiency by failing to detect idle channels. Furthermore, the accuracy of each model across the SNR range is shown in Figure 10. The results confirm that RF maintains superior classification accuracy compared to other techniques, reinforcing its overall robustness as a spectrum sensing solution.

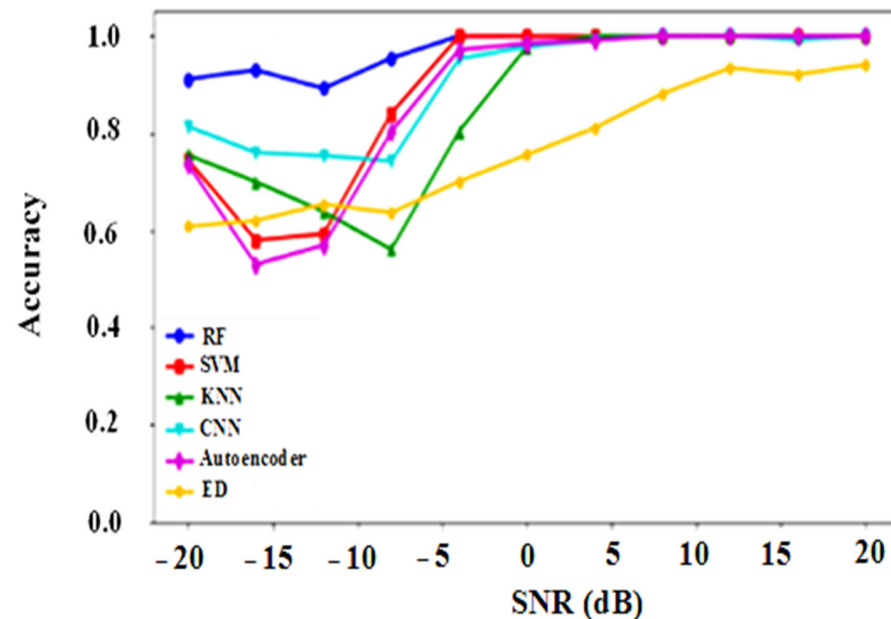


Figure 10. Accuracy against SNR for all the models.

Figure 11 presents the average accuracy for each model, where it is seen that RF achieves the highest accuracy at 96.93%, outperforming CNN (90.77%), SVM (88.71%), the Autoencoder (87.14%), KNN (85.78%), and ED (76.94%). This corresponds to relative performance gain of 6.59%, 8.71%, 10.32%, 11.72%, and 20.82% over these respective models.

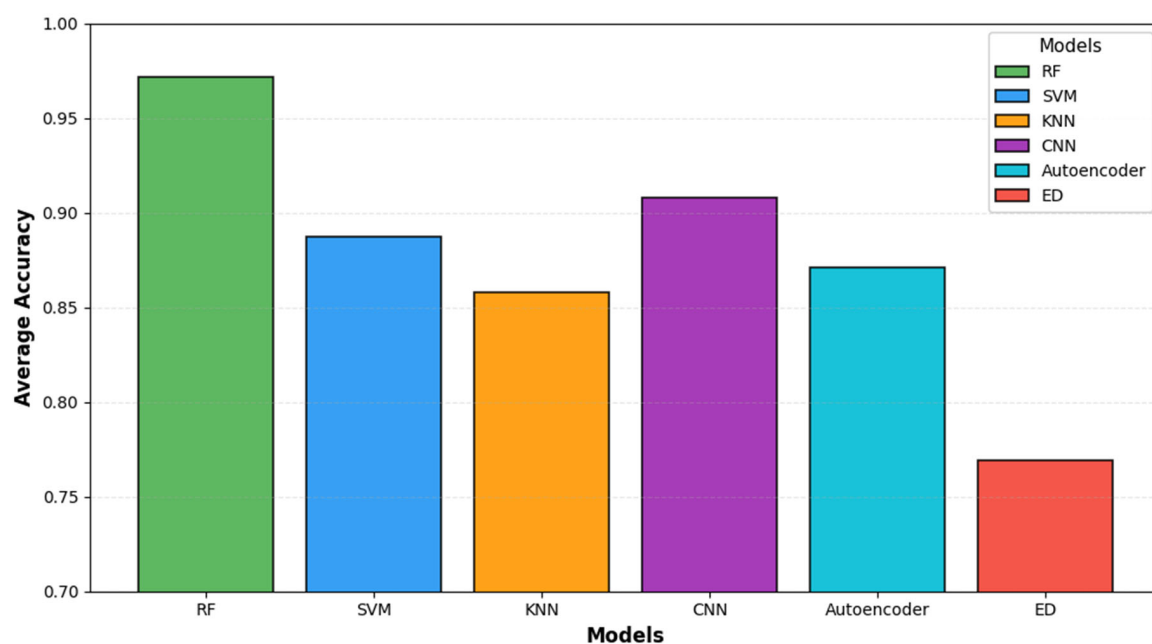


Figure 11. Comparison of accuracy of RF, SVM, KNN, CNN, Autoencoder, and ED techniques.

Figure 12 plots the F1-score against SNR to evaluate model performance under diverse signal conditions. The result reveals that RF produces the most favorable F1-score throughout the tested SNR range, demonstrating consistent reliability in both challenging low-SNR and favorable high-SNR scenarios.

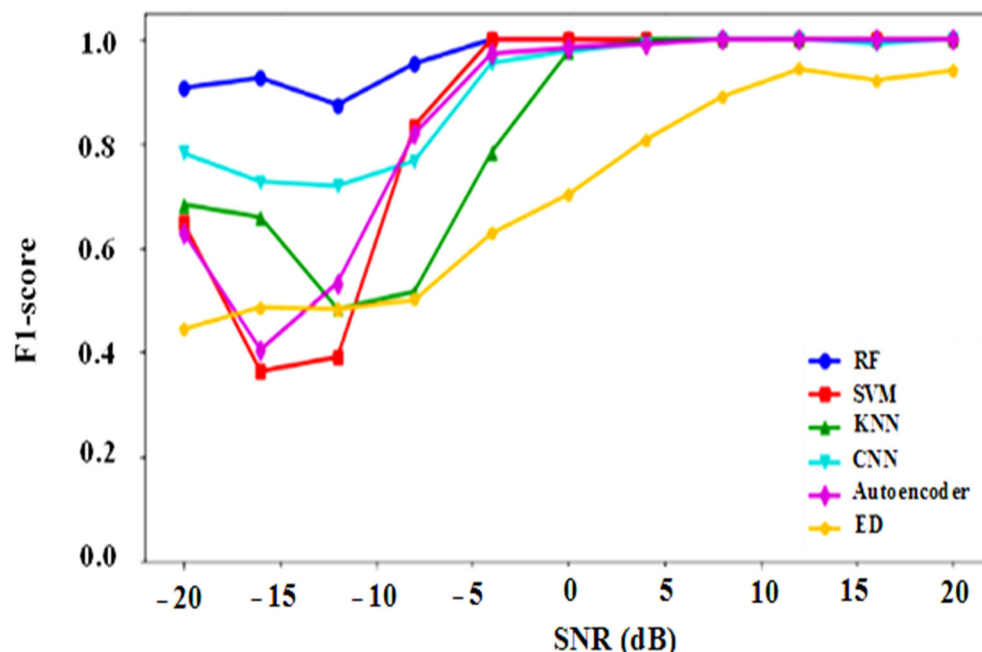


Figure 12. F1-score against SNR for all the models.

Figure 13 presents the average F1-score for each model across the SNR range of -20 dB to 20 dB. It is demonstrated in Figure 13 that RF achieves the highest score at 96.93%, followed closely by CNN (90.1%), while ED yields the lowest F1-score at 70.43%. The F1-scores of Autoencoder, SVM, and KNN are 84.88%, 83.98%, and 82.75%, respectively. This indicates that RF delivers performance gains of 7.04%, 12.05%, 13.36%, 14.63%, and 27.34% over CNN, Autoencoder, SVM, KNN, and ED, respectively. It is well known that a high value of F1 score corresponds to high values of precision and recall. Therefore, a high F1-score, as achieved by RF, indicates an optimal balance of high recall and high precision [41]. In the context of spectrum sensing, recall is the measure of the proportion of vacant spectrum that is identified to be present while precision defines the reliability of the decision that the PU band is not vacant. A high recall is crucial for preventing interference with the operation of the PU while a high precision is necessary for maximizing throughput and spectral efficiency. Thus, RF's superior F1-score confirms its excellent performance in maximizing spectrum access opportunities (high recall) and ensuring that those opportunities are correctly identified (high precision).

Figure 14 presents the computational training times for all evaluated models. The results show a clear trade-off between efficiency and performance. Classical ML models, specifically SVM (0.59 s) and KNN (1.26 s), train rapidly. In contrast, the deep learning architectures require substantially more time, the Autoencoder takes 6.09 s, while the CNN, with its high-density data processing, requires the longest training time of 25.42 s.

RF occupies a middle ground, with a training time of 2.47 s, which is moderately higher than SVM and KNN due to its ensemble process of generating and aggregating multiple decision trees. Crucially, this slight increase in computational cost is justified by its performance. While SVM and KNN exhibit the poorest accuracy, F1-score, and detection probabilities, RF delivers superior results across all key metrics. Therefore, RF achieves the optimal balance between training efficiency and predictive performance. This strong

performance stems from RF's inherent ability to effectively learn discriminative features from the transmitted signal, outperforming both simpler ML models and more complex DL architectures. Consequently, RF emerges as the most practical and high-performing candidate for real-time spectrum prediction in CR networks. In addition, Table 7 provides a comprehensive analysis of computational complexity alongside computing requirements for both traditional ML and DL models. The CNN exhibits the most substantial computational demands in our implementation, which is attributed to the computational intensity of convolution operations executed on CPU hardware without specialized acceleration.

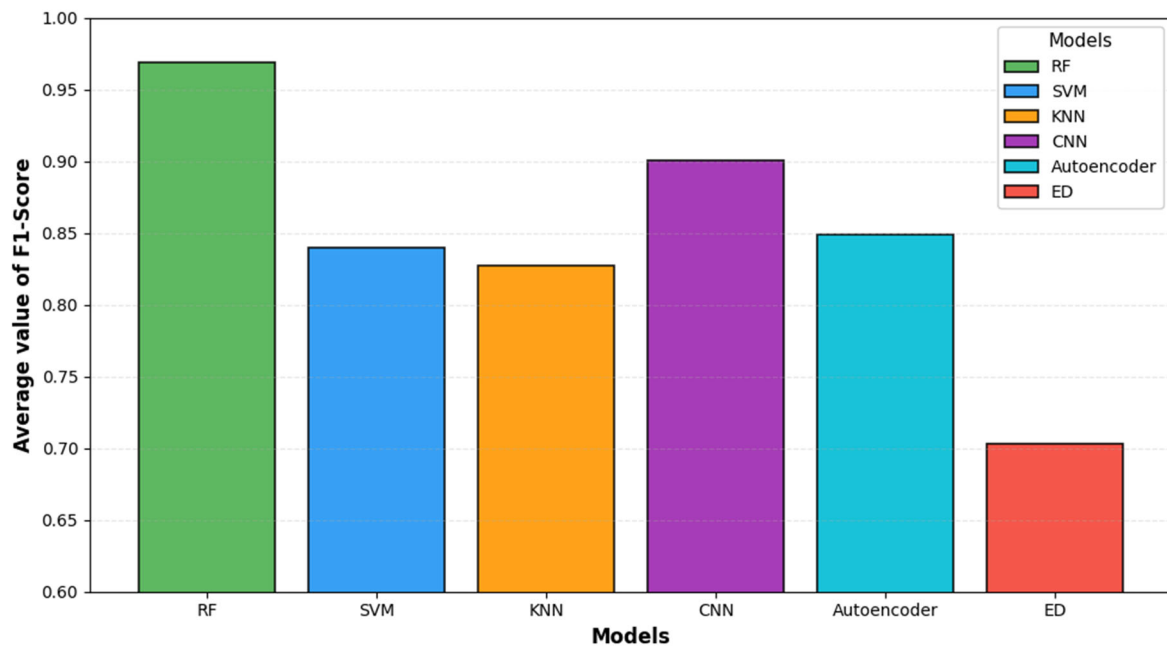


Figure 13. Comparison of F1-score of all the models.

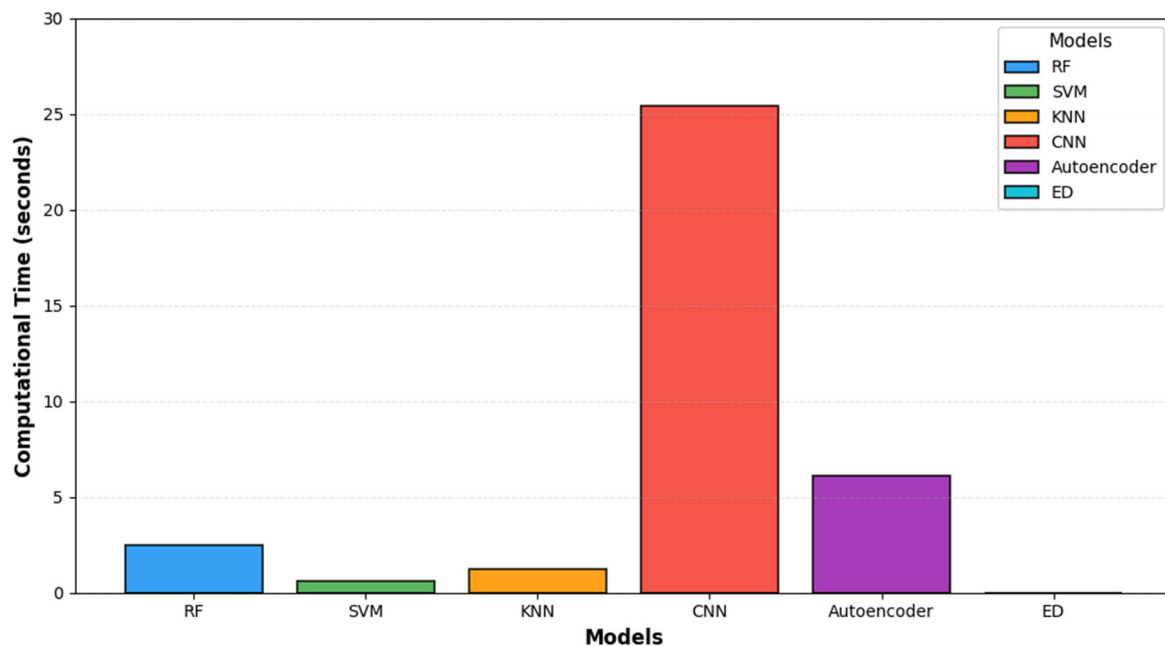


Figure 14. Computational time for RF, SVM, KNN, CNN, Autoencoder, and energy detection.

Table 7. Computational complexity and computing requirement analysis of ML and DL models.

Model	Computational Complexity	Training Time (s)	Memory	Parameter Definition
SVM	$O(m^2 \times q)$	0.59 s	8 MB	m = training samples q = extracted features
KNN	$O(m \times q)$	1.26 s	16 MB	m = training samples q = extracted features
RF	$O\left(\frac{n_{estimators} \times m \times q \times \log(m)}{m \times q \times \log(m)}\right)$	2.47 s	24 MB	m = training samples q = extracted features
Autoencoder	$O(E \times B \times \sum layers)$	6.09 s	112 MB	E = Epochs B = Batch size E = Epochs B = Batch size M = input channel (size of input feature maps) L = output channel (size of output feature representation) h = output height (height of the output feature representation) w = output width (width of the output feature representation) N = kernel/filter size
CNN	$O\left(\frac{E \times B \times \sum N^2 \times M \times L \times h \times w}{L \times h \times w}\right)$	25.42 s	384 MB	

Moreover, we validate the spectrum sensing techniques examined in this study through direct performance comparison with prior sensing studies from the literature. Specifically, the probability of detection achieved by RF (best) model is compared with the results reported in [41–44] under a consistent simulation scenario using a 64-length QPSK signal. The results of the comparative analysis are summarized in Table 8.

Table 8. Comparison of probability of detection of RF with previous study.

QPSK (64 Signal Length)					
Models	CNN–RNN [41]	CNN–LSTM [42]	CNN–LSTM [43]	DLsenseNet [44]	RF
Probability of detection at −20 dB	0.5073 (50.73%)	0.2637 (26.37%)	0.2680 (26.80%)	0.4097 (40.97%)	0.8556 (85.56%)

As shown in Table 8, under the extreme noise condition of −20 dB, the RF model achieves a higher detection probability than the hybrid CNN–RNN, CNN–LSTM, and DLsenseNet architectures reported in [41–44]. This provides strong evidence of RF’s robustness in low-SNR regimes, a critical advantage for practical spectrum sensing.

4. Conclusions

This study conducts a comparative performance evaluation of machine learning (ML) and deep learning (DL) algorithms for vacant spectrum detection in cognitive radio (CR) IoT networks. Within the ML paradigm, we examine random forest (RF), K-nearest neighbor (KNN), and support vector machine (SVM), while convolutional neural network (CNN) and Autoencoder models represent the DL approach. The simulation results, evaluated through accuracy, F1-score, and the probabilities of detection, false alarm, and missed detection, demonstrate that RF outperforms all other models, including CNN, Autoencoder, KNN, SVM, and traditional ED, across every performance metric considered. RF achieves the highest probability of detection, accuracy, and F1-score, while simultaneously recording the lowest probabilities of missed detection and false alarm, underscoring its exceptional

resilience in noisy low-SNR environments. A critical analysis of the training efficiency reveals a clear trade-off. While KNN and SVM train rapidly, their detection performance is suboptimal. Conversely, CNN and Autoencoder deliver relatively better accuracy but incur the longest training times. RF strikes the optimal balance, requiring only marginally more training time than KNN and SVM while delivering superior detection performance that surpasses the DL models. Consequently, RF emerges as the most efficient and effective candidate for practical spectrum sensing, offering the strongest compromise between computational cost and predictive accuracy. The study establishes RF as a robust, high-performance method ideally suited for CR networks. This study does not incorporate time-varying channel states, shadowing, and multi-user interference in the analysis. Future work will address these limitations by exploring generative transformer networks for spectrum sensing in CR networks for enhanced practical relevance.

Author Contributions: Conceptualization, A.A.R. and T.O.O.; methodology, A.A.R. and T.O.O.; investigation, A.A.R. and T.O.O.; writing—original draft preparation, A.A.R.; writing—review and editing, A.A.R. and T.O.O.; funding acquisition, T.O.O. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by the National Research Foundation (NRF), South Africa under the competitive programme for rated researchers with reference/grant number (CPRR240415214276).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors wish to acknowledge the support of Tshwane University of Technology (TUT), South Africa and the National Research Foundation (South Africa).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bhuiyan, M.N.; Rahman, M.; Billah, M.; Saha, D. Internet of Things (IoT): A review of its enabling technologies in healthcare applications, standards protocols, security, and market opportunities. *IEEE Internet Things J.* **2021**, *8*, 10474–10498. [\[CrossRef\]](#)
2. Yang, H.; Zhong, W.-D.; Chen, C.; Alphones, A.; Xie, X. Deep reinforcement learning based energy efficient resource management for social and cognitive internet of things. *IEEE Internet Things J.* **2020**, *7*, 5677–5689. [\[CrossRef\]](#)
3. Shaikh, F.K.; Zeadally, S.; Exposito, E. Enabling technologies for green Internet of Things. *IEEE Syst. J.* **2017**, *11*, 983–994. [\[CrossRef\]](#)
4. Albreem, M.A.M.; El-Saleh, A.A.; Isa, M.; Salah, W.; Jusoh, M.; Azizan, M.M.; Ali, A. Green Internet of Things: An overview. In Proceedings of the 4th IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), Putrajaya, Malaysia, 28–30 November 2017.
5. Khan, A.A.; Rehmani, M.H.; Rachedi, A. When cognitive radio meets the internet of things? In Proceedings of the 2016 International Wireless Communications and Mobile Computing Conference (IWCMC) (IEEE 2016), Paphos, Cyprus, 5–9 September 2016; pp. 469–474.
6. Colakovic, A.; Hadzialic, M. Internet of Things (IoT): A review of enabling technologies, challenges and open research issues. *Comput. Netw.* **2018**, *144*, 17–39. [\[CrossRef\]](#)
7. Raji, A.A.; Orimolade, J.F.; Ewetola, I.A. Design and implementation of internet of things scheme for testing loamy soil. *Turk. J. Eng.* **2025**, *9*, 323–333. [\[CrossRef\]](#)
8. Quang, V.D.; Insoo, K. Deep reinforcement learning based spectrum completion in green cognitive virtualized networks. *IEEE Access* **2021**, *9*, 52193–52201.
9. Raji, A.A.; Orimolade, J.F.; Adejumbi, I.A.; Amusa, K.A.; Olajuwon, B.I. Channel estimation via compressed sampling matching pursuit for hybrid MIMO architectures in millimeter wave communication. *Int. J. Electron. Lett.* **2025**, *13*, 56–70. [\[CrossRef\]](#)
10. Raji, A.A.; Orimolade, J.F.; Amusa, K.A.; Adejumbi, I.A. Channel estimation and MIMO combining architecture in millimeter wave system with few ADC bits. *J. Eng. Appl. Sci.* **2024**, *71*, 40. [\[CrossRef\]](#)
11. Bai, W.; Guoqiang, Z.; Weibing, X.; Yu, W.; Yujun, X. Multi-user opportunistic spectrum access for cognitive radio networks based on multi-head self-attention and multi-agent deep reinforcement learning. *Sensors* **2025**, *25*, 2025. [\[CrossRef\]](#)
12. Miah, S.; Michael, S.; Enda, B. A throughput analysis of an energy-efficient spectrum sensing scheme for the cognitive radio based internet of things. *EURASIP J. Wirel. Commun. Netw.* **2021**, *2021*, 201. [\[CrossRef\]](#)

13. Sarikhani, R.; Keynia, F. Cooperative spectrum sensing meets machine learning: Deep reinforcement learning approach. *IEEE Commun. Lett.* **2020**, *24*, 1459–1462. [\[CrossRef\]](#)
14. Guo, H.; Reisi, N.; Jiang, W.; Luo, W. Soft combination for cooperative spectrum sensing in fading channels. *IEEE Access* **2016**, *5*, 975–986. [\[CrossRef\]](#)
15. Emami, M.; Zarrabi, H.; Jabbarpour, M.R.; Sadat Taheri, M.; Jung, J.J. A soft cooperative spectrum sensing in the presence of most destructive smart PUEA using energy detector. *Concurr. Comput. Pract. Exp.* **2018**, *30*, e4524. [\[CrossRef\]](#)
16. So, J.; Sung, W. Group-based multi-bit cooperative spectrum sensing for cognitive radio networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 10193–10198. [\[CrossRef\]](#)
17. Raji, A.A.; Olwal, T.O. Spectrum sensing in cognitive radio internet of things networks: Applications, challenges and future prospects. *J. Sens. Actuator Netw.* **2025**, *14*, 109. [\[CrossRef\]](#)
18. Islam, H.; Das, S.; Bose, T.; Ali, T. Diode based reconfigurable microwave filter for cognitive radio applications: A review. *IEEE Access* **2020**, *8*, 185429–185444. [\[CrossRef\]](#)
19. Chae, K.; Kim, Y. DS2MA, A deep learning based spectrum sensing scheme for a multi-antenna receiver. *IEEE Wirel. Commun. Lett.* **2023**, *12*, 952–956. [\[CrossRef\]](#)
20. Hlapi, N.M. Enhancing hybrid spectrum access in CR-IoT networks: Reducing sensing time in Low SNR environments. *Mesopotamian J. Comput. Sci.* **2023**, *2023*, 47–52. [\[CrossRef\]](#)
21. Reyes, H.; Subramaniam, S.; Kaabouch, N.; Hu, W.C. A spectrum sensing technique based on autocorrelation and euclidean distance and its comparison with energy detection for cognitive radio networks. *Comput. Electr. Eng.* **2016**, *52*, 319–327. [\[CrossRef\]](#)
22. Kumar, A.; Gaur, N.; Chakravarty, S.; Alsharif, M.H.; Uthansakul, P.; Uthansakul, M. Cyclostationary and energy detection spectrum sensing beyond 5G waveforms. *Electron. Res. Arch.* **2023**, *31*, 3400–3416. [\[CrossRef\]](#)
23. Lv, Q.; Gao, F. Matched filter based spectrum sensing and power level recognition with multiple antennas. In Proceedings of the 2015 IEEE China Summit and International Conference on Signal and Information Processing (China SIP) (IEEE 2015), Chengdu, China, 12–15 July 2015; pp. 305–309.
24. Kalhor, S.; Umrani, F.A.; Khanzada, M.A.; Ali Rahoo, L. Matched filter based spectrum sensing technique for 4G cellular network. *Mehran Univ. Res. J. Eng. Technol.* **2019**, *38*, 973–978. [\[CrossRef\]](#)
25. Saad, M.A.; Mustafa, S.T.; Ali, M.H.; Hashim, M.M.; Bin Ismail, M.; Ali, A.H. Spectrum sensing and energy detection in cognitive networks. *Indones. J. Electr. Eng. Comput. Sci.* **2020**, *17*, 465–472. [\[CrossRef\]](#)
26. Patil, P.; Pawar, P.R.; Jain, P.P.; Manoranjan, K.V.; Pradhan, D. Enhanced spectrum sensing based on cyclostationary feature detector (CFD) in cognitive radio network using fixed & dynamic threshold level. *Saudi J. Eng. Technol.* **2020**, *5*, 271–277.
27. Pillay, N.; Xu, H. Blind eigenvalue-based spectrum sensing for cognitive radio networks. *IET Commun.* **2012**, *6*, 1388–1396. [\[CrossRef\]](#)
28. Miah, M.S.; Rahman, M.M. An eigenvalue and superposition approach based cooperative spectrum sensing in cognitive radio networks. In Proceedings of the 2014 International Conference on Electrical Engineering and Information & Communication Technology (IEEE, 2014), Dhaka, Bangladesh, 10–12 May 2014; pp. 1–7.
29. Badra, E.F.; Bashir, A.A.; Khaeirallah, H.N.; Farag, H.H. Dynamic spectrum co-access in multicarrier-based cognitive radio using graph theory through practical channel. *Appl. Sci.* **2024**, *14*, 10868. [\[CrossRef\]](#)
30. Mu, J.; Jing, X.; Huang, H.; Gao, N. Subspace-based method for spectrum sensing with multiple users over fading channel. *IEEE Commun. Lett.* **2017**, *22*, 848–851. [\[CrossRef\]](#)
31. Guimaraes, D.A. Pietra-Ricci index detector for centralized data fusion cooperative spectrum sensing. *IEEE Trans. Veh. Technol.* **2020**, *69*, 12354–12358. [\[CrossRef\]](#)
32. Patel, D.K.; Lopez-Benitez, M.; Soni, B.; Garcia-Fernandez, A.-F. Artificial neural network design for improved spectrum sensing in cognitive radio. *Wirel. Netw.* **2020**, *26*, 6155–6174. [\[CrossRef\]](#)
33. Samala, S.; Mishra, S.; Singh, S.S. Machine learning based cooperative spectrum sensing in a generalized α - κ - μ fading channels. *J. Sci. Ind. Res.* **2023**, *82*, 219–225.
34. Raghavendra, L.R.; Manjunatha, R.C. Optimizing spectrum sensing in cognitive radio using Bayesian-optimized random forest. *Int. J. Intell. Eng. Syst.* **2023**, *16*, 505–518.
35. Tan, T.; Jing, X. Cooperative spectrum sensing based on Convolutional Neural Networks. *Appl. Sci.* **2021**, *11*, 4440. [\[CrossRef\]](#)
36. Li, Y.; Song, H.; Ren, X.; Zhang, Z.; Cheng, S.; Jing, X. Spectrum sensing meets ISAC: A spectrum detection scheme for ISAC services based on improved denoising Autoencoder and CNN. *Appl. Sci.* **2025**, *5*, 3381. [\[CrossRef\]](#)
37. Kumar, A.; Gaur, N.; Nanthamornphong, A. Hybrid spectrum using Neural network-based MF and ED for enhanced detection in Rayleigh channel. *J. Electr. Comput. Eng.* **2025**, *2025*, 9506922. [\[CrossRef\]](#)
38. Subray, S.; Tsschmben, S.; Gifford, K. Towards enhancing spectrum sensing: Signal classification using Autoencoder. *IEEE Access* **2021**, *9*, 82288–82299. [\[CrossRef\]](#)
39. Kumar, A.; Gaur, N.; Chakravarty, S.; Alsharif, M.H.; Uthansakul, P.; Uthansakul, M. Analysis of spectrum sensing using deep learning algorithms: CNNs and RNNs. *Ain Shams Eng. J.* **2024**, *15*, 102505. [\[CrossRef\]](#)

40. Ajayi, O.O.; Badrudeen, A.A.; Oyedeggi, A.I. Deep learning based spectrum sensing technique for smarter cognitive radio networks. *J. Inven. Eng. Technol.* **2012**, *1*, 64–77.
41. Solanki, S.; Dehalwar, V.; Choudhary, J.; Kolhe, M.L.; Ogura, K. Spectrum sensing in cognitive radio using CNN-RNN and transfer learning. *IEEE Access* **2022**, *10*, 113482–113492. [[CrossRef](#)]
42. Gao, J.; Yi, X.; Zhong, C.; Chen, X.; Zhang, Z. Deep learning for spectrum sensing. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 1727–1730. [[CrossRef](#)]
43. Xie, J.; Fang, J.; Liu, C.; Li, X. Deep learning-based spectrum sensing in cognitive radio: A CNN-LSTM approach. *IEEE Commun. Lett.* **2020**, *24*, 2196–2200. [[CrossRef](#)]
44. Solanki, S.; Dehalwar, V.; Choudhary, J. Deep learning for spectrum sensing in cognitive radio. *Symmetry* **2021**, *13*, 147. [[CrossRef](#)]
45. Eren, B.; Cesur, I. Comparative analysis of machine learning models for CO emission prediction in engine performance. *Sak. Univ. J. Comput. Inf. Sci.* **2025**, *8*, 1–11. [[CrossRef](#)]
46. Demircan, F.; Ekinici, M.; Comertz, Z.; Gediklo, E. Enhanced classification of ear disease images using metaheuristic feature selection. *Sak. Univ. J. Comput. Inf. Sci.* **2025**, *8*, 58–75. [[CrossRef](#)]
47. Arjoune, Y.; Kaabouch, N. On spectrum sensing, a machine learning method for cognitive radio systems. In Proceedings of the 2019 IEEE International Conference on Electro Information Technology (EIT), Brookings, SD, USA, 20–22 May 2019; pp. 333–338.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.