SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V13I2P106

Volume 13 Issue 2, 83-98, February 2026

Original Article

Spectrum Sensing-Based Cognitive Radio Networks Model: A Hybrid Deep Learning-based Approach

Nagla Elhaj Babiker[1], Khalid Hamid Bilal[2], Magdi B. M. Amien[3]

[1]*Department of Electronics Engineering (Communications and Control), University of Gezira, College of Engineering and Technology, Wad Madani, Gezira State, Sudan.*
[2]*Department of Electronics Engineering, Omdurman Islamic University, College of Engineering and Technology, Omdurman, Sudan*
[3]*Department of Electrical and Electronics Engineering, University of Khartoum, Faculty of Engineering, Gamma Ave, Khartoum, Sudan.*

[1]*Corresponding Author : nagla.elhaj99@hotmail.com*

Received: 14 November 2025 Revised: 18 December 2025 Accepted: 16 January 2026 Published: 17 February 2026

Abstract - *In cognitive radio, optimum spectrum usage by secondary users depends on the detection of main user signals. There are several civilian and military uses of the ability to categorize incoming wireless signals. Recent impressive accomplishments in the field of wireless research have piqued the interest of researchers in using Deep Learning (DL), which is a subfield of Artificial Intelligence (AI), to solve the problem of modulation categorization. Recently, researchers released an updated version of the dataset, RadioML2016.10a, based on GNU Radio, which replicates the faults of a real wireless channel. This study proposed a Hybrid CNN (Convolutional Neural Network) and LSTM (Long Short-Term Memory) model with a user-defined activation function (CLDNN-Act) for spectral sensing. This study also examines three alternative models to evaluate the effectiveness of the proposed paradigm. The evaluation parameters include recall, F1-score, precision, Bit Error Rate (BER), and signal probability detection.*

Keywords - *Activation function, Cognitive radio, Convolutional Neural Network, Deep Learning, Long Short-Term Memory.*

1. Introduction

1.1. Background

A Cognitive Radio (CR), according to the International Telecommunication Union (ITU), is a radio that can be dynamically designed and configured to utilize the best available wireless channels to minimize congestion and user interference. To accomplish predetermined goals, such a radio automatically finds available channels in the wireless spectrum. However, using its knowledge, it then independently and suitably modifies its operational parameters and protocols. It also modifies its transmission parameters to permit more immediate wireless communications in a specific spectrum band at one location [1]. Owing to the existing concerns, Dynamic Spectrum Access/CR (DSA/CR) arises between spectrum demand growth and spectrum underutilization. It is a solution proposed to address the radio spectrum shortage issue. The aim of DSA/CR is to improve spectrum usage efficiency. This is done by allowing Secondary Users (SUs) to utilize underutilized spectrum bands originally allocated to primary Incumbent Users (IUs). Under DSA, SUs are granted access to spectrum bands with varying priority levels, allowing them to detect and utilize spectrum resources that are not fully utilized by IUs [2].

To meet the requirements for efficient usage of the available limited frequency spectrum, CR technology is a promising one. The CR's two main characteristics are its ability to capture information from its radio environment, as well as its ability to dynamically reconfigure itself and adjust its operational parameters and protocols by using the knowledge it has gained from the radio environment. These characteristics make CR an important technology for meeting high data transmission requirements and advancing next-generation technologies [3]. Deep Learning (DL) shows remarkable success in pattern recognition tasks, including image classification, speech processing, and time-series forecasting. Its application to wireless communications, mainly spectrum sensing, is motivated by several factors. These factors include learning of abstract and discriminative features from raw input data (e.g., IQ samples or spectrograms) without manual feature engineering. This eliminates the need for domain-specific signal knowledge. DL possesses several structural characteristics that are valuable for researchers applying it in the communications field to solve application sharing problems. DL can solve the primary challenge for conventional spectrum sensing methods, detection accuracy in low SNR conditions. It can model non-

linear relationships and maintain performance in boisterous environments, making it a more robust technique in low SNR [4]. Also, DL models can be used on hardware accelerators or lightweight edge devices that are suitable for real-time spectrum sensing. These models provide scalability and real-time potential for rapid inference speeds. CNNs and Recurrent Neural Networks (RNNs), such as LSTM and GRU, have demonstrated strong performance at capturing temporal dependencies in sequential data. These models play an important role in detecting signal presence or modulation patterns over time [5]. Nowadays, wireless settings are more complicated, and there is a growing demand for intelligent and automatic spectrum decision-making. Hence, DL becomes a strong tool for the next generation of cognitive radio networks. Spectrum sensing can change from inflexible, rule-based systems to flexible, intelligent sensing engines that adjust in real time by using deep architectures. This will make better and more dependable utilization of the radio spectrum possible.

1.2. Problem Statement

Orthogonal Frequency Division Multiplexing (OFDM) is a widely adopted modulation scheme in modern wireless communication systems. This is due to OFDM's robustness against multipath fading and spectral efficiency. Integrating OFDM into CR networks offers enhanced spectral agility and efficient usage of the available frequency bands. However, accuracy under Low Signal-To-Noise Ratio (SNR), Multipath Interference, Fading, and Doppler Shifts, and timing and frequency synchronization errors, remains a critical and challenging task. Existing spectrum sensing methods including energy detectors, is highly sensitive to noise uncertainty and cannot distinguish between noise and low-power OFDM signals [6], matched filters require prior data of the primary user's signal structure, which is often unavailable in practice [7], and cyclostationary detector is strong to noise uncertainty, but suffer from computationally expensive and unsuitable for real-time sensing in resource-constrained CR devices [8], provide promising opportunity to Machine Learning and DL methods to improve spectrum sensing performance by leveraging data-driven models that can learn discriminative features automatically from OFDM signal patterns [9]. Therefore, the central problem addressed in this work is to design an accurate, robust, and low-latency spectrum sensing model for OFDM-based CR networks that overcomes the limitations of traditional techniques and standalone DL models, particularly under low SNR and time-varying channel conditions.

1.3. Motivation

Traditional spectrum sensing methods face limitations and constraints, such as sensitivity to noise uncertainty, the requirement for prior knowledge, and high complexity. These approaches include matched filtering, energy detection, and cyclostationary feature detection. These methods motivate the use of DL in spectrum sensing-based cognitive radio for adaptive, data-driven approaches that can generalize across signal types, noise conditions, and environmental variability [10].

1.4. Contributions

Though much research has been done in DL based spectrum sensing and modulation classification, existing models often fail to maintain high accuracy under low SNR conditions. Also, these works face challenges when jointly capturing both spatial and temporal features. Most existing approaches are standalone CNN or LSTM models. Hence, these standalone approaches either focus on spatial pattern extraction or temporal sequence dependencies, but not both. Moreover, limited studies provide a comparison across multiple performance metrics such as BER and detection probability. Thus, this research seeks to explore hybrid DL architectures combining CNNs and LSTM. This hybrid model can jointly model the spatial-frequency and temporal characteristics of OFDM signals in dynamic and noisy wireless environments. This process aids in good results and adapts to real-world wireless environments. The proposed CLDNN-Act model combines CNN and LSTM networks with a custom activation function and Nadam optimizer to enhance spectrum sensing in CR systems. Compared to some existing models, including CNN, Deep Neural Network (DNN), Convolutional Long Short-Term Deep Neural Networks (CLDNN), the proposed model achieves higher detection accuracy, lower BER, and stronger robustness at low SNR levels.

2. Foundations and Related Work

Sensing OFDM signals using DL approaches is an important point in addressing challenges like timing delay, noise uncertainty, and CFO. Various studies have proposed methods to enhance OFDM signal sensing accuracy. One such approach involves utilizing stacked autoencoder networks for spectrum sensing. This approach extracts hidden features of OFDM signals for classification, proving superior strength to noise uncertainty, timing delay, and CFO in comparison to existing methods. Paper [11] proposes two novel OFDM sensing frameworks, SAE-SS and SAE-TF, which utilize DL networks to improve the sensing accuracy of OFDM. It introduces two novel OFDM sensing frameworks, SAE-SS and SAE-TF, utilizing DL networks to enhance sensing accuracy. These methods offer better performance compared to conventional OFDM sensing techniques, showcasing the potential of DL in spectrum sensing. In [12], S. Zheng introduces the OFDM-DetNet method for OFDM signal detection, utilizing raw IQ data as input, which is a confidence-based fusion method. This method enhances detection performance under low SNR conditions and for signals longer than the network input, utilizing raw IQ data as input. The paper [3] highlighted the practical applications of DL in enhancing wireless communication systems in dynamic and challenging environments by developing a CNN for classifying signals into idle, out-of-network users, in-network

users, and jammers based on modulation types. Signal classification by a CNN classifier classifies real signals accurately, with limited comparison with other DL models for spectrum sensing and no detailed discussion on the impact of noise on classification [7, 13] Proposes a spectrum sensing method termed an improved CNN based on LeNet-5 for the spectrum sensing model. This model converts the cyclic spectrum into a grayscale image for solving spectral perception. A Hybrid Spectrum Sensing (HSS) technique including Energy Detector (ED), Cyclostationary Detector (CSD), and Likelihood Ratio Statistics with the ANN trained on TSs for noise-only scenarios, which combines Test Statistics (TSs) from multiple detectors to make decisions on PU activity, demonstrating improved efficiency compared to non-hybrid detection methods [14, 15]. In [7], DNN proposed to determine the optimal energy detection threshold and fusion weights, aiming to adapt effectively to changing wireless channel conditions. This method eliminates the necessity for solving intricate optimization issues, which reduces computational complexity, particularly in large networks. Detection [16] uses CLDNN, introducing a DL based signal detector, DetectNet, that controls the structural information of modulated signals. For this, prior knowledge of channel state information or background noise is not required. The detector shows a comparative performance, overcoming the limitations of conventional energy detectors, such as the SNR due to noise uncertainty. [9] enhances spectrum sensing precision with low SNR detection using CNNs and RNNs, improves detection accuracy in dynamic and diverse spectrum conditions, and enables real-time processing for timely spectrum sensing applications. The review of the existing work is given in Table 1.

Table 1. Existing work review

Ref. No.	Model Used	Study Focus	Contribution	Limitation
[3]	CNN	RF signal classification in dynamic spectrum environments	Proposed a CNN-based classifier capable of identifying signal types under unknown and varying spectrum conditions	Limited evaluation of low-SNR robustness; lacks temporal sequence modeling
[7]	DNN	Energy optimization for hybrid PV-TEG systems using deep learning	Applied DNN to extract maximum power, demonstrating deep learning utility in energy systems	Not focused on communication signals; lacks a modulation classification application
[16]	CLDNN	Spectrum sensing in CR networks	Developed a CLDNN-based detector that learns spatial and temporal features of signals without prior knowledge of CSI or noise	Did not use custom activation or optimizer; generalization across SNR levels was not deeply analyzed.

The current research proposes a hybrid DL model (CLDNN-Act) designed to achieve robust and accurate detection under dynamic and low-SNR conditions for spectrum sensing in CR networks. The model integrates CNNs for spatial–frequency feature extraction and LSTM networks for temporal sequence learning. This enables a complete feature representation of OFDM-based signals. Another novelty of this work is the introduction of a custom activation function, which enhances nonlinearity, gradient flow, and learning stability. Additionally, the model employs the Nadam optimizer for faster and more reliable convergence.

Experimental evaluation using the RadioML2016.10a dataset across 20 SNR levels demonstrates good performance compared to some existing models using parameters such as accuracy, precision, recall, F1-score, and BER. The proposed model gives a practical, lightweight, and noise-resilient solution for real-time spectrum sensing in next-generation CR systems.

3. Proposed Methodology

The pictorial representation of the proposed methodology is illustrated in Figure 1.

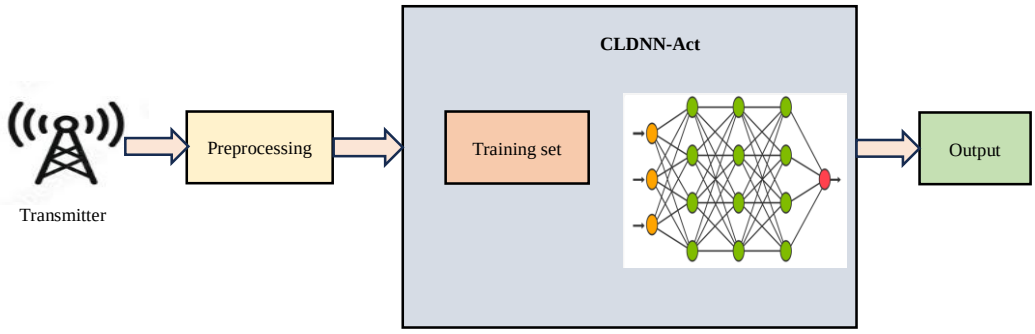

Fig. 1 Block diagram of the proposed approach

3.1. Dataset

This study used the RADIOML 2016.10A dataset to conduct many trials with 20 different SNR values, spanning from -20 dB to 18 dB. Below is a list of all eleven modulation classes, which includes "8PSK", "AM-SSB", "AM-DSB", "BPSK", "GFSK", "CPFSK", "QAM16", "PAM4", "QAM64", "QPSK", "WBFM". There are 11,000 IQ samples (streams of complicated data) in this dataset for every class. A dictionary stores the IQ samples that match the dataset's keys, along with the tuples of modulation class and SNR value [17].

3.2. Data Preprocessing

Data pre-processing began with reorganising the dataset so that it could be represented by a dictionary with a single key. The process began with digitizing the labels, also known as modulation classes, and assigning a unique number to each class. Afterwards, the labels were converted into a one-hot encoding vector, $V_i \epsilon R^n$, where n represents the total dataset classes. This study set all elements in this vector to 0, except for the one whose index value equals the class number, which was set to 1. This study performs this transformation to ensure the input is consistent with the goal function. Each class and SNR value has its own data point in the new dictionary, which is a list of one-hot encoded labels and IQ samples.

3.3. Spectrum Sensing Model

3.3.1. CNN Model

CNNs train their first layers to recognize small-scale "edges" that get more complex as the network deepens. An advantage of CNN is that it is shift-invariant. This means that converted weights in each layer may recognize patterns at any point in the sequence, and a max-pool layer transfers the signal's existence to a higher layer. This quality makes the signal so effective for detection by these networks. The convolution procedure can be expressed as in Equation (1):

$$conv(I,K)_{x,y} = \sum_{i=1}^{n_H} \sum_{j=1}^{n_W} \sum_{k=1}^{n_C} k_{i.j.k} I_{x+i-1,y+j-1,k} \quad (1)$$

Where I denote the input layer weight, K is the convolutional layer weight; H, W, and C are the height, width, and channel of the input. Conversely, in the wireless domain, CNNs process I/Q samples instead of images. Different waveforms of radio signals exhibit different transition patterns in the I/Q plane. As an example, BPSK signals often include transitions between (1,0) and (-1,0), whereas QPSK signals have a very different constellation. In the end, the CNN filters can learn this as a distinct "Signature" of the signal.

3.3.2. LSTM Model

RNNs are an advanced DL model that extracts temporal characteristics. People often use them to model linguistic sequences and multimodal time series. Unlike CNN, which has a fixed input dimension, RNN can analyze sequential data with varying lengths. When it comes to wireless networks, RNN takes into account both the present and past patterns of transition in I/Q samples. By using additional sequence characteristics, RNN achieves excellent performance. This study expresses the basic RNN as in Equations (2) and (3).

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \quad (2)$$

$$y_t = \sigma_h\left(W_y h_t + b_y\right) \quad (3)$$

The activation function σ_h, the weight parameters $\left(W_h, W_y, U_h\right)$, the hidden layer vector, h_t the input vector x_t, and the RNN bias parameters b_h, b_yAll play important roles. When training RNN models using Back Propagation Through Time (BPTT), gradient explosion or gradient disappearance may occur. One of the most famous and extensively used models, LSTM, is one of numerous RNN variations created to solve the issue. The Gated Recurrent Unit (GRU) makes the LSTM architecture much easier to program, which makes it work better than the RNN model with a single recurrent unit [2].

3.3.3 CLDNN-Act Model

CLDNNs are very useful for tasks that need to recognise time-domain signals like audio, video, and images because they have built-in memory that lets them find temporal correlations. This model uses both the CNN layer from the CNN model and the LSTM layer from the LSTM model. The LSTM layer, which sits between the Convolutional and dense layers, is responsible for this fundamental characteristic.

Softmax obtains the probability scores of the modulation classes at the last dense layer output. This study updates the model with a dropout rate of 0.3 and places the maxpooling layers after the Convolutional Layers. The user-defined activation function is created in this work, which is a composite function that combines two existing activation functions. This combines the tf.nn.relu(x) function, which is the Rectified Linear Unit (ReLU) activation function, which outputs x if x is positive, and 0 otherwise. This user-defined custom activation function then combines the tf.sin(x) function, which is a Sine activation function, which outputs the sine of x. The composite function can be described as in equation (4).

$$custom\_act(x) = ReLU\left(sin(x)\right) \quad (4)$$

The user-defined activation function uses the sine function, which causes this nonlinearity. Hence, the model is able to learn complicated correlations. The ReLU function limits the output range by ensuring the output is non-negative, thereby achieving a bounded output. The sine function's smooth derivatives may enhance optimization. Given that merging functions may enhance the model's representational capacity, this function might lead to an increase in that capacity. Table 2 shows the details of the layers used in the proposed CLDNN-Act model.

Table 2. Proposed CLDNN-act model

Layers	Input Parameters
Convolution (Conv2d)	Custom activation
MaxPooling2D	pool_size=(1, 2)
Dropout	Dropout rate 0.3
Conv2D	Custom activation
MaxPooling2D	pool_size=(1, 2)
Dropout	Dropout rate 0.3
Dense	128 hidden layer, Custom activation
Dropout	Dropout rate 0.3
Reshape	(2, 4096)
LSTM	Custom activation
Dropout	Dropout rate 0.3
Dense	128 hidden layer, Custom activation
Dropout	Dropout rate 0.3
Dense	11 hidden layer, softmax

Before the training method began, it created three distinct sets of data. This study used two sets of data: one for training and another for testing. This work allocated 90% to the training set and 10% to the test set.

This work uses a 10% subset of the training data as a validation set to track the model's progress at each training epoch and pinpoint instances of overfitting. Once this trains the model, this research evaluates the accuracy using test data.

This work chose the cross-entropy loss function as this study is addressing a classification job. Additionally, this estimates the model's parameters using the NADAM optimiser, which has a learning rate of 0.0003, rather than the commonly used ADAM optimiser. This study trains all the models for 100 epochs employing a 200-batch size. When training these models, this research relied on Keras' TensorFlow.

4. Results

The novelty of this work lies in the Hybrid CLDNN-Act Architecture, which works with Convolutional feature extraction and temporal sequence modeling simultaneously. Unlike prior methods that use separate CNN or LSTM models, CLDNN-Act improves classification robustness and performance under low SNR conditions.

A custom activation function is also employed that enhances feature discrimination. In this study, CLDNN-Act performance is compared against three other existing models, which include CNN, DNN, and standard CLDNN models.

The significant improvements in recall, precision, and BER are given for all these models. Tables 3, 4, and 5 detail the layer organization of these existing models.

Table 3. Existing CLDNN architecture

Layers	Input Parameters
Convolution (Conv2d)	Custom activation
MaxPooling2D	pool_size=(1, 2)
Dropout	Dropout rate 0.3
Conv2D	Custom activation
MaxPooling2D	pool_size=(1, 2)
Dropout	Dropout rate 0.3
Dense	128 hidden layer, Custom activation
Dropout	Dropout rate 0.3
Reshape	(2, 4096)
LSTM	Custom activation
Dropout	Dropout rate 0.3
Dense	128 hidden layer, Custom activation
Dropout	Dropout rate 0.3
Dense	11 hidden layer, softmax

Table 4. Existing CNN architecture

Layers	Input Parameters
Convolution (Conv2d)	Relu activation
BatchNormalization	-
MaxPooling2D	pool_size=(1, 2)
Dropout	Dropout rate 0.3
Conv2D	Relu activation
BatchNormalization	-
MaxPooling2D	pool_size=(1, 2)
Dropout	Dropout rate 0.3
Convolution (Conv2d)	Relu activation
BatchNormalization	-
MaxPooling2D	pool_size=(1, 2)
Dropout	Dropout rate 0.3
Convolution (Conv2d)	Relu activation
BatchNormalization	-
MaxPooling2D	pool_size=(1, 2)
Dropout	Dropout rate 0.3
Flatten	-
Dense	128 hidden layer, Relu activation
BatchNormalization	-
Dense	11 hidden layer, softmax

Table 5. Existing DNN architecture

Layers	Input Parameters
Dense	256 hidden layer, Relu activation
Dense	128 hidden layer, Relu activation
Dense	250 hidden layer, Relu activation
Dense	64 hidden layer, Relu activation
Flatten	-
Dense	128 hidden layer, Relu activation
Dense	11 hidden layer, softmax

Table 6 displays the hyperparameters for the four models used in this study. The proposed model uses an improved Nadam optimizer, as shown in the table, instead of the commonly used Adam optimizer.

All the other additional parameters used are also constant throughout all methods. Figure 2 displays the ROC curve for each class. In this figure, the X-axis depicts the Probability of False Alarm (Pfa), which is the proportion of noise signals incorrectly detected as a primary user.

Table 6. Hyperparameters

Hyperparameters	Proposed	CLDNN	CNN	DNN
Optimizer	Nadam	Adam	Adam	Adam
Learning rate	0.0003	0.0003	0.0003	0.0003
Epochs	100	100	100	100
Early stopping monitor	Validation loss (VL)	VL	VL	VL
Early stopping patience	3	3	3	3
Batch size	200	200	200	200

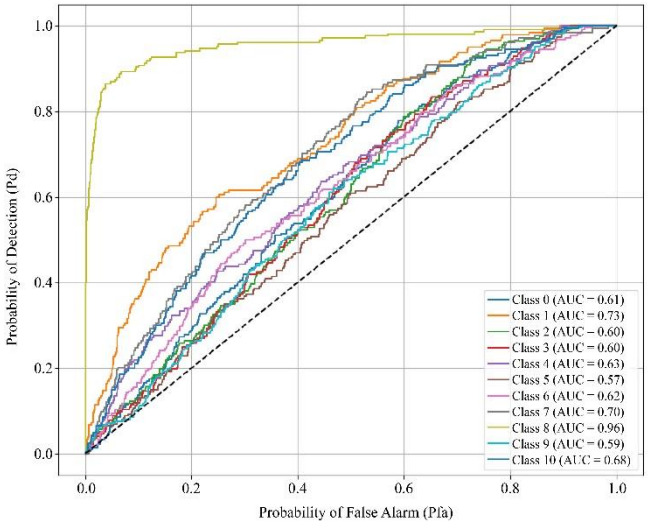

Fig. 2 Receiver Operating Characteristic (ROC) Curves

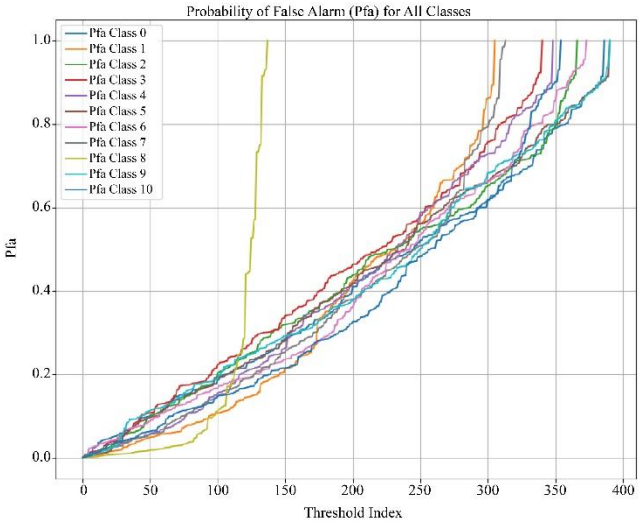

Fig. 3 Probability of False Alarm (PFA) for all classes

The y-axis represents Probability of Detection (Pd), which is the proportion of actual primary user signals correctly detected. Each line represents a separate class (modulation scheme or protocol type). The diagonal line is a random guess baseline where (Area Under the Curve) AUC = 0.5.

The model shows good performance as the curve is close to the top-left corner. Class 8 stands out with AUC = 0.96, which means this class is detected very reliably. Classes 5 and Class 3 have an AUC around 0.57–0.60, showing weaker classification performance. AUC summarizes performance, where if AUC of 1.0 is perfect, AUC greater than 0.9 is excellent, and an AUC approximately equal to 0.5 is poor (random). This plot helps identify which modulations are easily distinguishable and which ones may need more feature enhancement. Figure 3 depicts Pfa across different thresholds.

The x-axis represents the threshold index used to binarize the predicted class probabilities. The y-axis denotes Pfa, which shows how often a given class falsely activates detection when there is no actual PU signal.

Each line represents a class and how PFA changes with different thresholds. PFA usually increases as there is an increase in threshold. Some classes, including Class 8, show a sharp rise in PFA, indicating that they are susceptible to threshold shifts. Other classes show a more gradual increase, which indicates a more balanced and robust behavior in threshold selection.This graph helps with threshold tuning in selecting the best operating point for each class, understanding which classes are more prone to false alarms, and designing detection systems with desired trade-offs (e.g., low PFA for military, high Pd for emergency).

Table 7. Proposed accuracy, loss, validation loss, and validation accuracy

Proposed	Epoch	loss	Accuracy	Validation loss	Validation accuracy (VC)
SNR=2	1	2.3977	0.0927	2.3970	0.0966
	20	1.1349	0.4847	1.1584	0.4511
	40	0.9316	0.5870	0.9454	0.5580

	48	0.9078	0.5927	0.7806	0.6443
SNR=8	1	2.3972	0.1023	2.3929	0.0727
	15	1.3073	0.4542	1.3217	0.4330
	16	1.2823	0.4643	1.2298	0.4818
SNR=-4	1	2.3979	0.0918	2.3958	0.1193
	15	1.5618	0.3600	1.5096	0.3898
	20	1.4742	0.3937	1.4645	0.4148
SNR=6	1	2.3978	0.0880	2.3970	0.1432
	15	1.3057	0.4687	1.2556	0.4545
	21	1.1879	0.4951	1.1930	0.4841
SNR=12	1	2.3980	0.0894	2.3979	0.0739
	20	1.1507	0.5047	1.1438	0.4852
	28	1.0751	0.5263	1.0912	0.5182
SNR=-6	1	2.3980	0.0907	2.3980	0.0784
	15	1.4797	0.3765	1.3812	0.4182
	30	1.1818	0.4857	1.1910	0.5011
	38	1.1024	0.5249	1.1116	0.5443
SNR=-20	1	2.3980	0.0874	2.3981	0.0568
	3	2.3980	0.0857	2.3982	0.0568
SNR=-18	1	2.3980	0.0835	2.3980	0.0830
	3	2.3979	0.0888	2.3981	0.0920
SNR=16	1	2.3948	0.0985	2.3583	0.1409
	20	1.2906	0.4702	1.2780	0.4557
	28	1.1588	0.4912	1.2015	0.4625
SNR=10	1	2.3953	0.1038	2.3551	0.1193
	20	1.1425	0.4904	1.1317	0.5216
	40	0.9939	0.5428	0.9413	0.5682
	43	0.9672	0.5462	0.9372	0.5773
SNR=4	1	2.3978	0.0917	2.3951	0.1216
	20	1.2047	0.4957	1.1546	0.5080
	23	1.1390	0.5115	1.1069	0.5227
SNR=-2	1	2.3980	0.0888	2.3980	0.0830
	20	1.3500	0.4428	1.2716	0.4591
	31	1.1586	0.5040	1.1475	0.4830
SNR=-8	1	2.3980	0.0929	2.3980	0.0841
	15	1.5426	0.3720	1.4663	0.4068
	24	1.4371	0.4005	1.3879	0.4409
SNR=-12	1	2.3980	0.0866	2.3980	0.0898
	3	2.3980	0.0923	2.3982	0.0898
SNR=0	1	2.3976	0.0922	2.3939	0.0852
	20	1.3932	0.4312	1.3479	0.4443
	40	1.0491	0.5327	1.0259	0.5295
	48	0.9394	0.5629	0.9587	0.5477
SNR=-16	1	2.3979	0.0919	2.3981	0.0750
	3	2.3979	0.0917	2.3989	0.0670
SNR=-10	1	2.3980	0.0913	2.3981	0.0750
	3	2.3978	0.0944	2.3986	0.0898
SNR=14	1	2.3954	0.0985	2.3686	0.1102
	15	1.3686	0.4354	1.2635	0.4830
	30	1.1178	0.5003	1.0211	0.5568
	35	1.0684	0.5232	1.1026	0.5409
SNR=18	1	2.3953	0.1054	2.3688	0.1511
	15	1.3562	0.4525	1.2483	0.5091

	30	1.0370	0.5374	0.9618	0.5580
	40	0.9284	0.5658	1.1521	0.5125
SNR=-14	1	2.3980	0.0876	2.3979	0.0909
	6	2.3979	0.0861	2.3979	0.0841

Viewing this table 7 for the proposed results reveals that an increase in SNR results in an increase in accuracy and a decrease in loss. For SNR-20, the VL is 2.3982, and the VA is 0.0568. The VA increases to 0.0920 for SNR-18, but the VL shows a point difference from SNR-20 loss.

For SNR-16, the values show some fluctuations, with VL increasing to 2.3989 and VA decreasing to 0.0670 when compared to SNR-20 and SNR-18. On the other hand, SNR-14 experiences a decrease in VL to 2.3979 and an improvement in VA to 0.0841. Similarly, SNR-12 experiences an increase in VL to 2.3982 and a decrease in VA to 0.0898. SNR-10 shows an increase in VL to 2.3986 and a decrease in VA to 0.0898. For SNR-8, the VL is 1.3879, and the VA is 0.4409. SNR-6 observes a VL of 1.1116 and a decrease in VA to 0.5443. For SNR-4, the VL is 1.4645, and the VA is 0.4148. For SNR-2, the VL is 1.1475, and the VA is 0.4830. The VA is 0.5477, and the VL decreases to 0.9587 for SNR = 0. For SNR 2, the VL is 0.7806, and the VA is 0.6443. For SNR 4, the VL is 1.1069, and the VA is 0.5227. The VA is 0.4841, and the VL improves to 1.1930 for SNR 6. The VA is 0.4818, and the VL increases to 1.2298 for SNR 8. For SNR 10, the VL is 0.9372, and the VA is 0.5773. For SNR 12, the VL is 1.0912, and the VA is 0.5182. SNR 14 experiences a VL of 1.1026 and an increase in VA to 0.5409. For SNR 16, the VL is 1.2015, and the VA is 0.4625. Finally, for SNR 18, the VL is 1.1521, and the VA decreases to 0.5125.

Table 8. CLDNN accuracy, loss, validation loss, and validation accuracy

CLDNN	Epoch	loss	accuracy	Validation loss	Validation accuracy
SNR=2	1	2.3980	0.0854	2.3979	0.0864
	10	1.5812	0.3682	1.5488	0.3722
	20	1.3072	0.4520	1.3219	0.4403
	21	1.2678	0.4649	1.2945	0.4449
SNR=8	1	2.3980	0.0851	2.3980	0.0790
	10	1.6822	0.3303	1.6306	0.3517
	20	1.3666	0.4435	1.3265	0.4665
	30	1.1221	0.5118	1.0696	0.5307
	37	1.0424	0.5311	1.0615	0.5290
SNR=-4	1	2.3980	0.0893	2.3981	0.0778
	15	1.4214	0.4004	1.3335	0.4420
	30	0.9870	0.5672	0.9180	0.6017
	45	0.8673	0.6099	0.8057	0.6574
	54	0.7849	0.6463	0.7654	0.6545
SNR=6	1	2.3980	0.0871	2.3981	0.0807
	15	1.3864	0.4357	1.2356	0.4989
	30	1.0920	0.5298	0.9779	0.5710
	45	0.9894	0.5544	0.8836	0.6045
	60	0.8792	0.6088	0.8157	0.6210
	75	0.8056	0.6337	0.7608	0.6313
	83	0.7592	0.6517	0.7610	0.6392
SNR=12	1	2.3980	0.0871	2.3980	0.0886
	10	1.5708	0.3589	1.4834	0.3682
	20	1.3224	0.4533	1.2183	0.4756
	29	1.1842	0.4930	1.3385	0.4301
SNR=-6	1	2.3979	0.0916	2.3979	0.0824
	10	1.9965	0.2474	1.8342	0.2960
	20	1.3193	0.4423	1.1951	0.5057
	29	1.1934	0.4905	1.1613	0.4994
SNR=-20	1	2.3980	0.0868	2.3981	0.0875
	3	2.3979	0.0950	2.3983	0.0869
SNR=-18	1	2.3980	0.0929	2.3985	0.0784

	3	2.3978	0.0949	2.3990	0.0750
SNR=16	1	2.3980	0.0905	2.3980	0.0903
	10	1.5820	0.3636	1.5199	0.3773
	20	1.3830	0.4308	1.3134	0.4472
	30	1.1540	0.4969	1.0451	0.5386
	40	1.0056	0.5250	0.9345	0.5358
	46	0.9370	0.5507	0.9977	0.5136
SNR=10	1	2.3979	0.0866	2.3980	0.0892
	15	1.4127	0.4163	1.3369	0.4466
	30	1.0774	0.5156	1.0499	0.5267
	45	0.9598	0.5541	0.9217	0.5545
	59	0.8947	0.5737	0.8764	0.5727
SNR=4	1	2.3979	0.0878	2.3978	0.0943
	10	1.5644	0.3700	1.4357	0.4233
	20	1.2204	0.4811	1.2445	0.4858
	26	1.1538	0.5014	1.2066	0.4886
SNR=-2	1	2.3980	0.0912	2.3979	0.0892
	10	1.6553	0.3357	1.5812	0.3534
	20	1.3692	0.4378	1.3239	0.4437
	30	1.2150	0.4855	1.2099	0.4795
	40	1.1326	0.5153	1.1620	0.4920
	44	1.1144	0.5190	1.2063	0.4722
SNR=-8	1	2.3980	0.0911	2.3980	0.0943
	15	1.6125	0.3690	1.4830	0.4085
	23	1.5029	0.3980	1.4024	0.4091
SNR=-12	1	2.3980	0.0885	2.3979	0.1023
	3	2.3979	0.0945	2.3980	0.0869
SNR=0	1	2.3979	0.0933	2.3980	0.0778
	15	1.4718	0.3895	1.3875	0.4324
	30	1.1525	0.4979	1.0655	0.5392
	45	0.9797	0.5429	0.9474	0.5545
	52	0.9431	0.5517	0.9272	0.5665
SNR=-16	1	2.3980	0.0923	2.3979	0.0977
	3	2.3979	0.0925	2.3980	0.0847
SNR=-10	1	2.3980	0.0925	2.3980	0.0898
	3	2.3979	0.0949	2.3981	0.0898
SNR=14	1	2.3979	0.0922	2.3978	0.0920
	15	1.3756	0.4382	1.2851	0.4528
	30	1.0856	0.5124	1.0383	0.5256
	34	1.0520	0.5173	1.0178	0.5182
SNR=18	1	2.3978	0.0885	2.3969	0.1608
	15	1.3461	0.4530	1.2748	0.4705
	30	1.0890	0.5197	0.9966	0.5540
	32	1.0699	0.5246	1.0316	0.5369
SNR=-14	1	2.3980	0.0835	2.3983	0.0847
	3	2.3979	0.0916	2.3988	0.0841

Table 8 for CLDNN reveals that an increase in SNR results in a decrease in loss and an increase in accuracy. For SNR-20, the VL is 2.3983, and the VA is 0.0869. The VA is 0.0750 for SNR-18, but the VL is 2.3990.

For SNR-16, the VL decreases to 2.3980, and VA increases to 0.0847. SNR-14 experiences a decrease in VL to 2.3988 and also a decrement in VA to 0.0841. Then, SNR-12 experiences a decrease in VL to 2.3983 and an increase in VA to 0.0869. SNR-10 experiences a decrease in VL to 2.3981 and an increase in VA to 0.0898. For SNR-8, the VL is 1.4024, and the VA is 0.4091. For SNR-6, it observes a VL of 1.1613 and a decrease in VA to 0.4994. For SNR-4, the VL is 0.7654, and the VA is 0.6545. For SNR-2, the VL is 1.2063, and the

VA is 0.4722. The VA is 0.5665, and the VL decreases to 0.9272 for SNR-0. For SNR 2, the VL is 1.2945, and the VA is 0.4449. For SNR 4, the VL is 1.2066, and the VA is 0.4886. The VA is 0.6392, and the VL decreases to 0.7610 for SNR 6. The VA is 0.5290, and the VL increases to 1.0615 for SNR 8. For SNR 10, the VL is 0.8764, and the VA is 0.5727. For SNR 12, the VL is 1.3385, and the VA is 0.4301. SNR 14 experiences a VL of 1.0178 and an increase in VA to 0.5182. For SNR 16, the VL is 0.9977, and the VA is 0.5136. Finally, for SNR 18, the VL is 1.0316, and the VA increases to 0.5369.

Table 9. CNN accuracy, loss, validation loss, and validation accuracy

CNN	Epoch	loss	accuracy	Validation loss	Validation accuracy
SNR=2	1	2.2948	0.2641	2.4458	0.0955
	3	1.2210	0.5038	3.4112	0.0989
SNR=8	1	2.2649	0.2834	2.6706	0.0886
	3	1.4424	0.4601	4.9022	0.0866
SNR=-4	1	2.1914	0.2746	2.4741	0.0932
	3	1.2969	0.5092	3.2179	0.0932
SNR=6	1	2.1827	0.2810	2.9122	0.1006
	3	1.6454	0.4024	4.8399	0.1006
SNR=12	1	2.2665	0.2751	2.5730	0.0932
	3	1.4496	0.4412	4.5514	0.0932
SNR=-6	1	2.5277	0.2021	2.4502	0.0943
	3	1.6373	0.4040	2.6887	0.0977
SNR=-20	1	3.0827	0.0912	2.4167	0.1034
	3	2.8298	0.0962	2.5019	0.1034
SNR=-18	1	3.0852	0.0869	2.4343	0.0892
	3	2.8262	0.1031	2.5991	0.0977
SNR=16	1	2.2549	0.2781	2.5362	0.1034
	3	1.4493	0.4494	4.4932	0.1034
SNR=10	1	2.2282	0.2773	2.6035	0.0767
	3	1.4381	0.4516	4.3646	0.0767
SNR=4	1	2.3058	0.2639	2.5359	0.0864
	3	1.4080	0.4690	3.8557	0.1040
SNR=-2	1	2.1849	0.2810	2.4364	0.0960
	3	1.2211	0.5145	3.0717	0.0989
SNR=-8	1	2.6790	0.1793	2.5196	0.0903
	3	1.7840	0.347	3.8890	0.0903
SNR=-12	1	3.0193	0.1139	2.4569	0.0926
	3	2.4333	0.2085	2.5973	0.0926
SNR=0	1	2.2438	0.2751	2.5337	0.0983
	3	1.2201	0.5094	3.5698	0.0983
SNR=-16	1	3.1053	0.0871	2.4656	0.0852
	3	2.7997	0.1007	2.5545	0.0807
SNR=-10	1	2.7973	0.1518	2.4504	0.0943
	3	2.0999	0.2649	2.9167	0.0943
SNR=14	1	2.2613	0.2710	2.5841	0.0847
	3	1.4031	0.4696	5.1019	0.0847
SNR=18	1	2.1536	0.2926	2.6442	0.0898
	3	1.4238	0.4561	4.7325	0.0898
SNR=-14	1	3.0921	0.0929	2.4509	0.0898
	3	2.6507	0.1513	2.5594	0.0898

Table 9 for CNN reveals that an increase in SNR results in a decrease in loss and an increase in accuracy.

For SNR-20, the VL is 2.5019, and the VA is 0.1034. The VA is 0.0977 for SNR-18, and the VL is 2.5991. For SNR-16, the VL is decreasing to 2.5545, and the VA is decreasing to 0.0807. The SNR-14 experiences an increase in VL to 2.5594 and an improvement in VA to 0.0898. Similarly, SNR-12 experiences an increase in VL to 2.5973 and an increase in VA to 0.0926. SNR-10 experiences an increase in VL to 2.9167

and also an increase in VA to 0.0943. SNR-8 has the VL as 3.8890 and the VA as 0.0903. SNR-6 has a VL of 2.6887 and an increase in VA to 0.0977. For SNR-4, the VL is 3.2179, and the VA is 0.0932. For SNR-2, the VL is 3.0717, and the VA is 0.0989. The VA is 0.0983, and the VL improves to 3.5698 for SNR = 0. For SNR 2, the VL is 3.4112, and the VA is 0.0989. For SNR 4, the VL is 3.8557, and the VA is 0.1040. The VA is 0.1006, and the VL improves to 4.8399 for SNR 6. The VA is 0.0866, and the VL increases to 4.9022 for SNR 8. For SNR 10, the VL is 4.3646, and the VA is 0.0767. For SNR 12, the VL is 4.5514, and the VA is 0.0932. SNR 14 experiences a VL of 5.1019 and a decrease in VA to 0.0847. For SNR 16, the VL is 4.4932, and the VA is 0.1034. Finally, for SNR 18, the VL is 4.7325, and the VA increases to 0.0898.

Table 10. DNN accuracy, loss, validation loss, and validation accuracy

DNN	Epoch	loss	accuracy	Validation loss	Validation accuracy
SNR=2	1	2.3903	0.1369	2.3612	0.1432
	20	1.0715	0.5646	1.4764	0.4091
SNR=8	1	2.3761	0.1268	2.2911	0.1852
	20	1.0534	0.5913	1.3308	0.4739
	28	0.8651	0.6597	1.3701	0.4955
SNR=-4	1	2.3951	0.1125	2.3832	0.1705
	16	1.4017	0.5014	1.9500	0.3114
SNR=6	1	2.3770	0.1418	2.3003	0.1477
	20	0.9220	0.6388	1.2926	0.4932
SNR=12	1	2.3771	0.1284	2.3385	0.0966
	15	1.1745	0.5375	1.3540	0.4784
	30	0.7367	0.7025	1.0713	0.5807
SNR=-6	1	2.3955	0.1133	2.3896	0.1341
	10	1.8068	0.3827	2.1365	0.2443
SNR=-20	1	2.3982	0.0923	2.3979	0.0955
	3	2.3979	0.0934	2.3980	0.0955
SNR=-18	1	2.3980	0.0889	2.3979	0.0807
	3	2.3979	0.0922	2.3982	0.0807
SNR=16	1	2.3963	0.0922	2.3932	0.1170
	20	1.1005	0.5578	1.3918	0.4477
SNR=10	1	2.3776	0.1309	2.3246	0.1693
	20	0.9013	0.6426	1.2580	0.5045
	30	0.6567	0.7328	1.1675	0.5659
SNR=4	1	2.3868	0.1188	2.3442	0.1216
	20	1.0405	0.5865	1.3847	0.4500
	30	0.7033	0.7196	1.3772	0.4875
SNR=-2	1	2.3885	0.1208	2.3464	0.1864
	16	1.1151	0.6005	1.7817	0.3693
SNR=-8	1	2.3981	0.0889	2.3979	0.0830
	11	1.9497	0.3350	2.2445	0.2080
SNR=-12	1	2.3980	0.0894	2.3979	0.0795
	3	2.3979	0.0926	2.3981	0.0693
SNR=0	1	2.3848	0.1188	2.3383	0.1636
	17	1.0297	0.6205	1.6384	0.3864
SNR=-16	1	2.3981	0.0894	2.3981	0.0875
	3	2.3978	0.0943	2.3984	0.0875
SNR=-10	1	2.3980	0.0902	2.3980	0.0841
	3	2.3979	0.0865	2.3980	0.0807
SNR=14	1	2.3875	0.1322	2.3496	0.1580
	20	0.8457	0.6649	1.2863	0.5205
	26	0.6891	0.7216	1.2616	0.5420
SNR=18	1	2.3811	0.1490	2.3215	0.1420
	20	0.9594	0.6221	1.2528	0.5011
	26	0.7811	0.6958	1.1871	0.5273

SNR=-14	1	2.3981	0.0957	2.3982	0.0807
	3	2.3978	0.0968	2.3984	0.0807

Table 10 for DNN reveals that an increase in SNR results in a decrease in loss and an increase in accuracy. For SNR-20, the VL is 2.3980, and the VA is 0.0955. The VA is 0.0807 for SNR-18, but the VL shows some point difference.

For SNR-16, the values show some fluctuations, with VL increasing to 2.3984 and VA also increasing to 0.0875. On the other hand, SNR-14 has the same VL as SNR-16 and a decrement in VA to 0.0807. Similarly, SNR-12 experiences a decrease in VL to 2.3981 and a decrease in VA to 0.0693.

SNR-10 experiences a decrease in VL to 2.3980 and an increase in VA to 0.0807. For SNR-8, the VL is 2.2445, and the VA is 0.2080. For SNR-6, this observes a VL of 2.1365 and the same VA as SNR-8. For SNR-4, the VL is 1.9500, and the VA is 0.3114. For SNR-2, the VL is 1.7817, and the VA is 0.3693. The VA is 0.3864, and the VL decreases to 1.6384 for SNR = 0. For SNR 2, the VL is 1.4764, and the VA is 0.4091. For SNR 4, the VL is 1.3772, and the VA is 0.4875.

The VA is 0.4932, and the VL decreases to 1.2926 for SNR 6. The VA is 0.4932, and the VL increases to 1.3701 for SNR 8. For SNR 10, the VL is 1.1675, and the VA is 0.5659. For SNR 12, the VL is 1.0713, and the VA is 0.5807. SNR 14 experiences a VL of 1.2616 and a decrease in VA to 0.5420. For SNR 16, the VL is 1.3918, and the VA is 0.4477. Finally, for SNR 18, the VL is 1.1871, and the VA increases to 0.5273. Figures 4 and 5 show the VL and VA graph comparison.

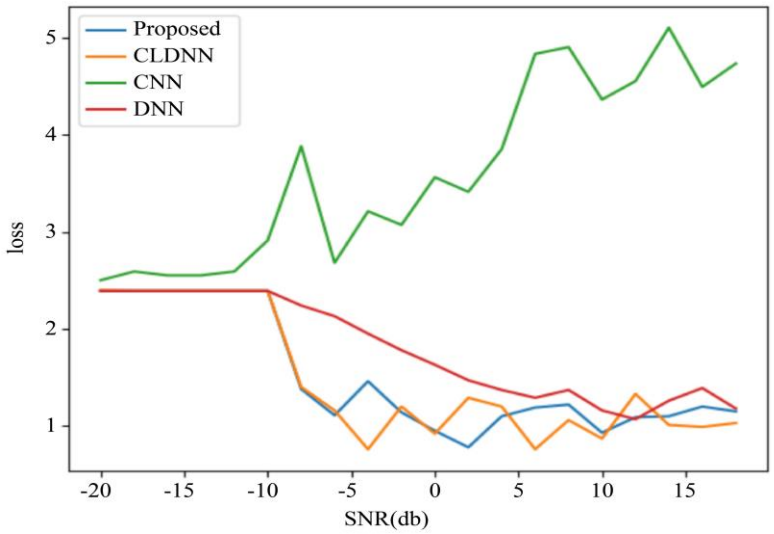

Fig. 4 Validation loss graph

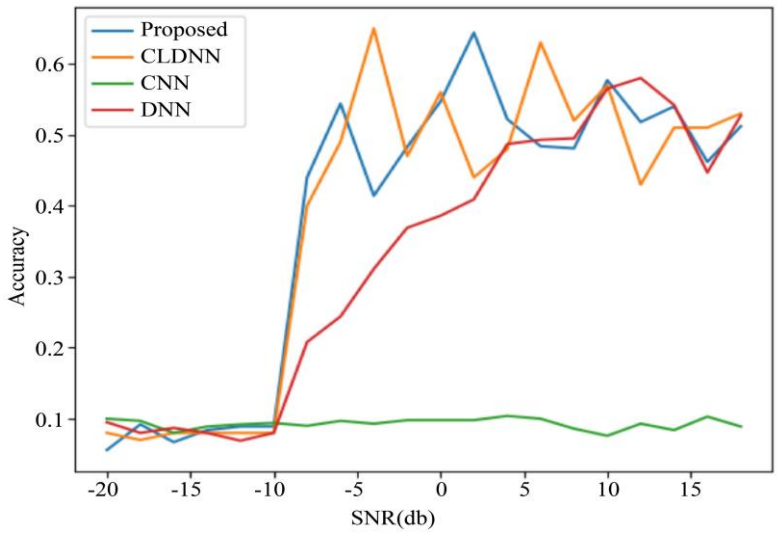

Fig. 5 Validation accuracy graph

Table 11. Comparison table

		Signal probability detection (SPD)	BER	Precision	recall	F1-score	accuracy
Proposed	SNR=2	0.6609	0.7615	0.67	0.66	0.64	0.66
	SNR=8	0.4481	1.2728	0.42	0.45	0.41	0.45
	SNR=-4	0.3813	1.5338	0.39	0.39	0.35	0.38
	SNR=6	0.4695	1.2196	0.43	0.47	0.43	0.47
	SNR=12	0.5240	1.0490	0.53	0.52	0.51	0.52
	SNR=-6	0.5154	1.1063	0.50	0.52	0.48	0.52
	SNR=-20	0.0981	2.3980	0.01	0.09	0.02	0.10
	SNR=-18	0.0804	2.3982	0.01	0.09	0.01	0.08
	SNR=16	0.5072	1.1390	0.50	0.51	0.49	0.51
	SNR=10	0.5436	0.9816	0.53	0.54	0.50	0.54
	SNR=4	0.5077	1.1499	0.51	0.51	0.49	0.51
	SNR=-2	0.4772	1.2060	0.47	0.48	0.44	0.48
	SNR=-8	0.4036	1.4479	0.40	0.40	0.36	0.40
	SNR=-12	0.0786	2.3981	0.01	0.09	0.01	0.08
	SNR=0	0.5577	0.9616	0.55	0.57	0.53	0.56
	SNR=-16	0.0836	2.3984	0.01	0.09	0.01	0.08
	SNR=-10	0.0859	2.3982	0.01	0.09	0.01	0.09
	SNR=14	0.5322	1.0369	0.52	0.52	0.49	0.53
	SNR=18	0.5681	0.8805	0.56	0.57	0.54	0.57
	SNR=-14	0.0777	2.3983	0.01	0.09	0.01	0.08
CLDNN	SNR=2	0.4650	1.2195	0.46	0.46	0.44	0.47
	SNR=8	0.5263	0.9990	0.53	0.54	0.51	0.53
	SNR=-4	0.5995	0.8336	0.61	0.60	0.55	0.60
	SNR=6	0.6236	0.8543	0.61	0.62	0.61	0.62
	SNR=12	0.5181	1.1200	0.50	0.52	0.47	0.52
	SNR=-6	0.4177	1.3366	0.42	0.42	0.37	0.42
	SNR=-20	0.0854	2.3982	0.01	0.09	0.01	0.09
	SNR=-18	0.0854	2.3984	0.01	0.09	0.01	0.09
	SNR=16	0.5618	0.8756	0.54	0.56	0.54	0.56
	SNR=10	0.5781	0.8676	0.56	0.58	0.54	0.58
	SNR=4	0.5186	1.1102	0.51	0.52	0.50	0.52
	SNR=-2	0.5099	1.1366	0.49	0.51	0.47	0.51
	SNR=-8	0.4318	1.4181	0.41	0.44	0.40	0.43
	SNR=-12	0.0863	2.3980	0.01	0.09	0.01	0.09
	SNR=0	0.5568	0.9373	0.54	0.55	0.52	0.56
	SNR=-16	0.0863	2.3981	0.01	0.09	0.01	0.09
	SNR=-10	0.0818	2.3983	0.01	0.09	0.01	0.08
	SNR=14	0.5204	1.0189	0.51	0.52	0.47	0.52
	SNR=18	0.5609	1.0362	0.55	0.56	0.53	0.56
	SNR=-14	0.0859	2.3981	0.01	0.09	0.01	0.09
CNN	SNR=2	0.0886	4.5213	0.01	0.09	0.01	0.09
	SNR=8	0.0881	5.9820	0.01	0.09	0.01	0.09
	SNR=-4	0.0831	3.6599	0.01	0.09	0.01	0.08
	SNR=6	0.0954	9.0770	0.01	0.09	0.02	0.10
	SNR=12	0.0890	5.7924	0.01	0.09	0.01	0.09
	SNR=-6	0.0895	3.0265	0.01	0.09	0.01	0.09
	SNR=-20	0.0968	2.5461	0.01	0.09	0.02	0.10
	SNR=-18	0.0895	2.7192	0.01	0.09	0.01	0.09
	SNR=16	0.0922	6.0755	0.01	0.09	0.02	0.09
	SNR=10	0.10045	5.4179	0.01	0.09	0.02	0.10
	SNR=4	0.0904	4.6563	0.01	0.09	0.02	0.09
	SNR=-2	0.0804	3.6872	0.01	0.09	0.01	0.08

	SNR=-8	0.0877	4.5833	0.01	0.09	0.01	0.09
	SNR=-12	0.0845	2.7493	0.01	0.09	0.01	0.08
	SNR=0	0.0881	4.1866	0.01	0.09	0.01	0.09
	SNR=-16	0.0890	2.6904	0.01	0.09	0.01	0.09
	SNR=-10	0.0986	3.1821	0.01	0.09	0.02	0.10
	SNR=14	0.0931	6.4403	0.01	0.09	0.02	0.09
	SNR=18	0.0900	5.8593	0.01	0.09	0.02	0.09
	SNR=-14	0.0845	2.5767	0.01	0.09	0.01	0.08
DNN	SNR=2	0.4050	1.5017	0.41	0.41	0.41	0.41
	SNR=8	0.5013	1.3328	0.51	0.51	0.50	0.50
	SNR=-4	0.3259	1.8579	0.32	0.33	0.32	0.33
	SNR=6	0.4840	1.3557	0.51	0.48	0.47	0.48
	SNR=12	0.5609	1.1417	0.56	0.56	0.55	0.56
	SNR=-6	0.2463	2.1667	0.25	0.25	0.24	0.25
	SNR=-20	0.0799	2.3983	0.01	0.09	0.01	0.08
	SNR=-18	0.0904	2.3980	0.01	0.09	0.02	0.09
	SNR=16	0.4645	1.3593	0.49	0.47	0.45	0.46
	SNR=10	0.5450	1.2063	0.56	0.55	0.54	0.55
	SNR=4	0.4972	1.2531	0.51	0.49	0.49	0.50
	SNR=-2	0.3600	1.8174	0.37	0.37	0.36	0.36
	SNR=-8	0.2090	2.2754	0.20	0.21	0.19	0.21
	SNR=-12	0.0877	2.3980	0.01	0.09	0.01	0.09
	SNR=0	0.3895	1.6926	0.42	0.39	0.39	0.39
	SNR=-16	0.0799	2.3989	0.01	0.09	0.01	0.08
	SNR=-10	0.08681	2.3981	0.01	0.09	0.01	0.09
	SNR=14	0.5600	1.1834	0.60	0.57	0.55	0.56
	SNR=18	0.5495	1.2015	0.56	0.54	0.54	0.55
	SNR=-14	0.0736	2.3989	0.01	0.09	0.01	0.07

Table 11 describes the comparison of the proposed work with the other three existing works with parameters like BER, SPD, precision, recall, f1score, and accuracy. When noticing the Figures 6 and 7, it is observed that, other than CNN, the other approaches, including the proposed one, are providing better BER and SPD. The reason is that in existing CLDNN, every operation is the same, other than the optimizer and custom activation function.

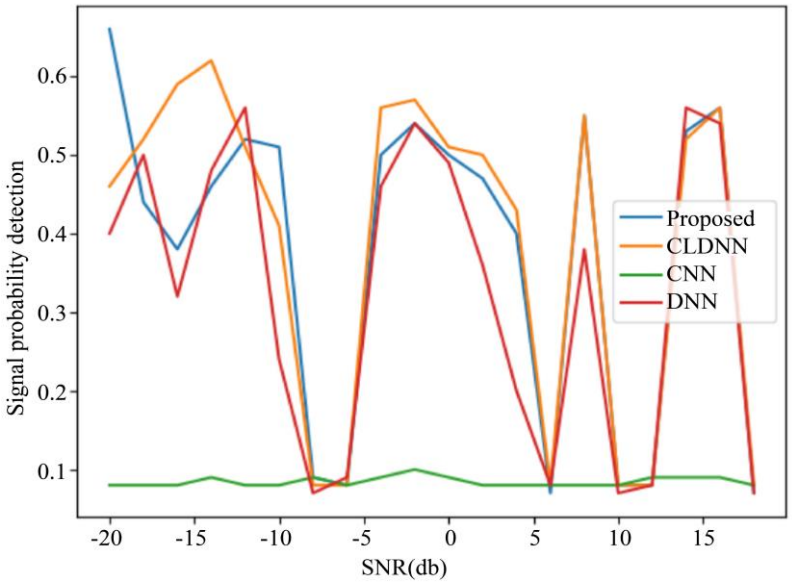

Fig. 6 Signal probability detection comparison graph

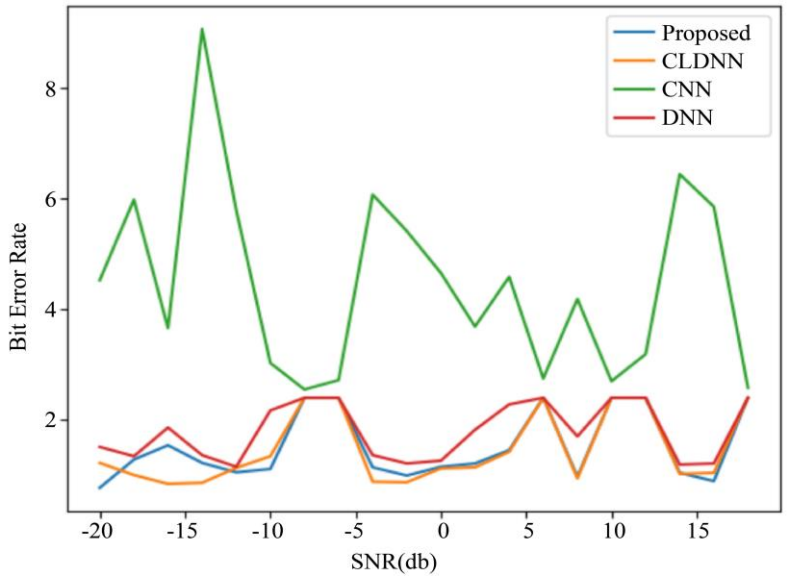

Fig. 7 Bit error rate comparison graph

Also, in DNN, more dense layers are found other than the normally used one or two dense layers. Further, when comparing the proposed, CLDNN, and DNN, the proposed shows some better results. This is due to the use of the Nadam optimizer and a custom user-defined activation function. Also, for some of the parameters, such as the accuracy, precision, recall, f1score, and SPD, with decreasing SNR, the value increases and vice versa. Then, for some parameters, such as the loss and BER with increasing SNR, the value decreases and vice versa.

5. Conclusion and Future Work

For spectrum sensing, this study analyzed three different existing techniques and proposed the CLDNN-Act model. To categorize modulations, the proposed model extracted signal features using a combination of convolutional, residual, and recurrent layers. With the proposed CLDNN-Act model, classification accuracy reaches a new height. Since there is just a slight variation in attained classification accuracy, other existing models like CNN and CLDNN can also do a decent job with the classification task. The inherent characteristics of the layers used in each design are responsible for this subtle variation. Because DNN models have varying depths, the classification accuracy may also change. Regardless, each of these models achieves adequate classification accuracy. This study executes this research using the RadioML 2016. 10A dataset. This aims to learn more about the impact of various layers in the future and investigate more models that have proven effective in modulation recognition tasks.

Funding Statement

Authors should state how the research and publication of their article was funded, by naming financially supporting bodies, followed by any associated grant numbers in square brackets.

Acknowledgments

An Acknowledgements section is optional and may recognise those individuals who provided help during the research and preparation of the manuscript. Other references to the title/authors can also appear here, such as “Author 1 and Author 2 contributed equally to this work.

References

[1] Miguel Lopez-Benitez, and Fernando Casadevall, “Signal Uncertainty in Spectrum Sensing for Cognitive Radio,” *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1231-1241, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[2] Joseph Mitola, and Gerald Q. Maguire, “Cognitive Radio: Making Software Radios More Personal,” *IEEE Personal Communications*, vol. 6, no. 4, pp. 13-18, 1999. [CrossRef] [Google Scholar] [Publisher Link]

[3] Yi Shi et al., “Deep Learning for RF Signal Classification in Unknown and Dynamic Spectrum Environments,” *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Newark, USA, pp. 1-10, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Sadaf Nazneen Syed et al., “Deep Learning Approaches for Spectrum Sensing in Cognitive Radio Networks,” *2022 25th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Herning, Denmark, pp. 480-485, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] Vatsala Sharma, and Sunil Joshi, "A Literature Review on Spectrum Sensing in Cognitive Radio Applications," *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, pp. 883-893, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Kenneth Kimani, and Morris Njiraine, "Cognitive Radio Spectrum Sensing Mechanisms in TV White Spaces: A Survey," *Engineering, Technology & Applied Science Research*, vol. 8, no. 6, pp. 3673-3680, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] Noman Mujeeb Khan et al., "Analysis of Deep Learning Models for Estimation of MPP and Extraction of Maximum Power from Hybrid PV-TEG: A Step towards Cleaner Energy Production," *Energy Reports*, vol. 11, pp. 4759-4775, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Mohamed Saber et al., "A Cognitive Radio Spectrum Sensing Implementation based on Deep Learning and Real Signals," *Innovations in Smart Cities Applications Volume 4*, pp. 930-941, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ya Tu et al., "Large-scale Real-world Radio Signal Recognition with Deep Learning," *Chinese Journal of Aeronautics*, vol. 35, no. 9, pp. 35-48, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Ankush Jolly, Payam Khorramshahi, and Tabish Saeed, "*Improvements to Modulation Classification Techniques using Deep Learning*," Noiselab, University of California, San Diego, 2020. [Google Scholar]

[11] Qingqing Cheng et al., "Sensing OFDM Signal: A Deep Learning Approach," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7785-7798, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[12] Shilian Zheng et al., "OFDM Sensing based on Deep Learning," *Physical Communication*, vol. 61, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Guangliang Pan, Jun Li, and Fei Lin, "A Cognitive Radio Spectrum Sensing Method for an OFDM Signal based on Deep Learning and Cycle Spectrum," *International Journal of Digital Multimedia Broadcasting*, vol. 2020, pp. 1-10, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Abbass Nasser et al., "A Deep Neural Network Model for Hybrid Spectrum Sensing in Cognitive Radio," *Wireless Personal Communications*, vol. 118, no. 1, pp. 281-299, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Na Wang et al., "Multidimensional CNN-LSTM Network for Automatic Modulation Classification," *Electronics*, vol. 10, no. 14, pp. 1-14, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Jiabao Gao et al., "Deep Learning for Spectrum Sensing," *IEEE Wireless Communications Letters*, vol. 8, no. 6, pp. 1727-1730, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] Dataset, DeepSi, 2024. [Online]. Available: https://www.deepsig.ai/datasets/