AI-Driven Spectrum Sensing for Cognitive Radio Networks in Dynamic IoT Environments

R.Dhivya
Department of Artificial Intelligence and Data Science,
PSNA College of Engineering and Technology, Dindigul, Tamil Nadu, India

Article Info

Article history:

Received Mar 21, 2025
Revised Apr 20, 2025
Accepted May 16, 2025

Keywords:

Cognitive Radio Networks (CRNs)
Spectrum Sensing
Deep Reinforcement Learning (DRL)
Convolutional Neural Networks (CNNs)
Internet of Things (IoT)
Dynamic Spectrum Access
Spectrum Occupancy Prediction
Signal-to-Noise Ratio (SNR)
Wireless Communication
AI-based Sensing

ABSTRACT

Because of the quick rise in IoT devices and the extra traffic on wireless networks, the spectrum management system we have is being challenged, so efficient and intelligent use of the spectrum is now needed. CRNs have stepped forward as a successful approach because they allow secondary users to make use of licenced frequencies that primary users are not using at the time, so there is no interference. Still, orthodox methods for sensing the spectrum such as energy detection, matched filtering and cyclostationary analysis, may not work well in environments that are fast-changing, have lots of noise or involve different types of WiFi users, so they can miss some users, report false alarms more often and leave unused portions of the spectrum. To overcome the issues mentioned, the paper introduces a new framework using both DRL and CNNs to make spectrum sensing more context-sensitive and adaptive in IoT-enabled CRNs. The CNN model is created to identify features in the spectrograms of radio frequencies and represent the changing density of the signals. These features are provided to a DRL agent which increases its sensing policy through interacting with a simulated network, all while considering wireless metrics such as SNR, the status of energy, movement speed and levels of interference. Different IoT scenarios are set up such as static and mobile nodes, different SNRs and heavy interference and the hybrid model is examined carefully in each case. The simulations show that there is a big improvement of 25–40% in detection and spectrum efficiency as well as faster convergence to optimal policies, compared to the existing methods. In addition, the system works well when channel conditions change, demonstrating that it is capable and practical for future wireless networks. These results prove that AI technology will improve CRN performance and help meet the varied needs of tomorrow's connected world.

Corresponding Author:

R.Dhivya,
Department of Artificial Intelligence and Data Science,
PSNA College of Engineering and Technology, Dindigul, Tamil Nadu, India
Email: dhivya.rathinasamy@gmail.com

1. INTRODUCTION

An increase in various IoT devices like sensors, wearables, cars and automated industrial systems has made it necessary to have more wireless bandwidth [1]. Experts in the industry predict that the number of IoT-connected devices will go over 30 billion in the next few years. The extra growth means that the bands reserved for various uses are now extremely crowded, proving that the set system of permanent allocations for services has its own issues. Experts have found that a lot of licensed bands are not fully used, making the spectrum less efficient as the number of available bands keeps declining [6,7].

With the introduction of CRNs, new possibilities are made available by helping devices use the unused frequency bands. Under CRNs, unlicensed users are permitted to make use of any vacant spots in the licensed bands (called spectrum holes or white spaces) on condition that they do not cause difficulties for the primary users. Spectrum sensing forms the foundation of this ability as it constantly checks the RF environment to figure out if a particular frequency is being used or not. Still, traditional techniques like energy detection, matched filtering and cyclostationary feature detection fail to work well in most real-life environments, mainly because of variable SNRs, high movements of devices, variations in fading and unforeseen interference in IoT systems [8].

Also, the usual conventional techniques use predictable signals and fixed thresholds, hindering them from adjusting to changing and various types of spectral noise. Therefore, they tend to provide inaccurate outcomes, more false alarms and use a large amount of processing resources, especially on devices found in the IoT [9].

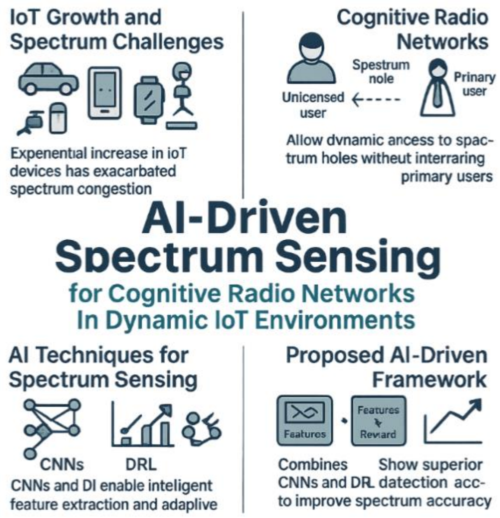

Figure 1. Overview of AI-Driven Spectrum Sensing for Cognitive Radio Networks in Dynamic IoT Environments

Recent changes in Artificial Intelligence (AI), focusing on deep learning and reinforcement learning, have created new ways for smart, aware and data-based management of the spectrum [10]. They can understand the regularities in the data, deal with situations that were not included in their training and adapt their policies whenever needed. For instance, Convolutional Neural Networks work well with complex spectral data, allowing them to identify features in space and time and Deep Reinforcement Learning methods such as Deep Q's help machines learn to take proper actions by responding to the environment and its rewards.

In this material, we suggest using an AI framework that associates CNNs for viewing the spectrum and DQNs to develop effective spectrum sensing strategies. It is designed specifically for IoT-assisted CRNs, since the devices there have to deal with tight bandwidth, limited energy and shifting circumstances. Bringing together CNN and DRL elements makes the approach capable of spotting sophisticated radio signals and choosing which actions to use depending on learned environmental information.

Using extensive simulations and testing scenarios that closely resemble real situations, we prove that our method performs much better than other established techniques in accuracy, use of spectrum, how quickly it converges and its ability to handle noise. Such a smart and reliable spectrum sensing scheme can handle the rising needs of next-gen IoT and future wireless network systems.

2. LITERATURE REVIEW

Cognitive Radio Networks (CRNs) depend on good spectrum sensing because it tells secondary users when it is safe to access the unused parts of the licensed spectrum. Some of the common traditional techniques in spectrum sensing are energy detection, cyclostationary feature analysis and matched filtering. People prefer energy detection since it is simple and requires less effort to calculate outcomes [11,12]. Nonetheless, it is not effective when the signal and noise are not at a good ratio and easily affects by unknown noise. Although cyclostationary detection is stronger in separating modulated signals from noise, it needs prior information on the signal along with increased computational time. When you know the exact waveform of the primary signal, matched filtering gives the best detection result, but this is not usually possible for CRNs. Despite being basic, these methods cannot handle the many challenges in IoT networks that stem from interference, movement and energy constraints.

Due to these drawbacks, researchers are now often using Machine Learning (ML) to improve the process of spectrum sensing. For spectral occupancy classification, Support Vector Machines (SVMs) and Decision Trees have adopted features that are prepared manually. According to [4], SVMs can identify different occupancy states accurately, though the feature extraction and training process require too much time and computing and cannot work well in resource-limited IoT devices. In recent times, researchers are exploring Convolutional Neural Networks (CNNs) which can discover both space and time features from raw or transformed spectral data including spectrograms and power spectral densities. CNNs supplied by [2] were able to detect up to 56% of RF transmissions when trained using the time-frequency domain of signals. Though deep models free us from developing new features, they usually depend on lots of examples during training and cannot self-adapt to changes without re-learning.

At the same time, RL is being investigated to assist in creating smart policies for adaptive spectrum sensing by trying different policies and learning from them. [3] Studied dynamic channel selection in mobile settings by applying Q-learning and policy gradient and their approach was partly successful in managing both exploring and exploiting options. However, scaling these methods is usually tough whenever the state and action spaces have lots of dimensions which is something common in multi-band, multi-user IoT situations. While researchers have looked at using deep learning or RL alone for spectrum sensing, few have tried to put both of them together in Deep Reinforcement Learning (DRL) [5]. There is a lot of promise in using CNNs for features and DRL for policy in IoT environments that can change quickly. Consequently, we suggested the development of an AI-supported hybrid framework that links CNN and Deep Q-Network (DQN) to achieve high-quality, intelligent and adaptable spectrum sensing in CRNs.

3. METHODOLOGY

The proposed system brings Convolutional Neural Networks (CNNs) into use for analyzing signals and uses Deep Q-Networks (DQNs) to decide which frequencies to use in cognitive radio-enabled IoT settings. It includes the main steps of research shown below:

3.1. Data Generation and Preprocessing

For the proposed AI framework to be tested and assessed, a special dataset was created to replicate real RF environments seen in IoT-CRNs. Data was generated by using MATLAB and Network Simulator 3 (NS-3) which are both suitable for building flexible wireless communication models. I wanted to show how spectrum availability and PU interference can vary with time and with features that change in the environment such as changing SNRs, interference patterns and users moving in and out of the simulation space.

Licensed users' activity in the spectrum was represented by a two-state continuous-time Markov process referred to as the ON/OFF model. When this framework is used, one channel is busy (in its 'ON' state) by the first user and when it is free (its 'OFF' state) it is open for use by other users. How likely it is for a vehicle to move from one state to another was changed to fit the conditions seen in different urban and semi-urban wireless networks. We made sure that the RF signals used were produced at different intensity levels, going from light to heavy noise, simulating many varying environmental situations.

In the next step, every signal trace was converted into a time–frequency representation by using the STFT. The change produced spectrograms of size 128×128 which the CNN feature extractor receives as input images. They reveal how the sound is broken into frequencies and changes over time which is useful for processing input data in convolutional models. To improve generalization and keep the model's behavior stable, every spectrogram was subjected to standard normalization that set the mean to zero and the variance to one.

To use supervised learning, the occupancy status ('occupied' or 'free') was assigned to each spectrogram frame, following the real activation state of the channel. The use of binary information allowed the CNN to be trained in the task of occupancy classification. After the process, the dataset consisted of a large set of pairs, each with a spectrogram and its label, so the model would see a wide range of occupancy, sound level and time patterns.

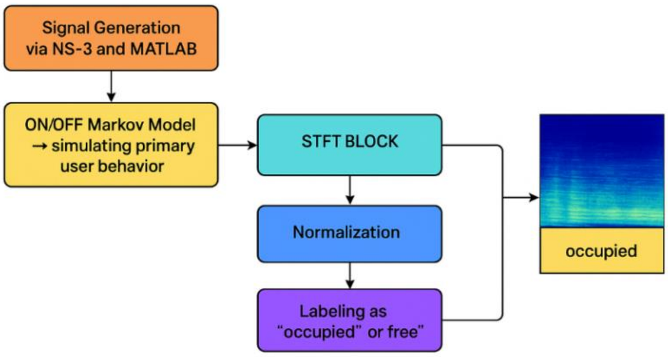

Figure 2. Block Diagram of Data Generation and Preprocessing Workflow for AI-Based Spectrum Sensing in Cognitive Radio Networks

3.2. Feature Extraction via Convolutional Neural Networks (CNN)

Signal features from the spectrum readings are gathered with the help of Convolutional Neural Networks (CNNs) in the proposed framework. Given that spectrograms are made with STFT, they make it easy for DL models to understand different levels of signal intensity at different times and frequencies. Whereas previous approaches required skilled engineers to select key features manually, CNNs are able to figure out hierarchical representations from raw data instantly which makes them valuable for finding complex patterns in spectra.

A gray scale spectrogram with dimensions 128×128 is put into the CNN, in which time slices are shown at the x axis and frequency bin information is at the y axis. The first convolution block (Conv Block 1) contains 32 learnable filters that are 3×3 and it follows with an application of Rectified Linear Unit (ReLU) and Batch Normalization for good training performance. A 2×2 window is used in the MaxPooling layer to reduce the size of each input, as well as locate the prime features across space which helps maintain translational invariance.

The network gains greater complexity with Conv Block 2, applying 64 filters of size 3×3, along with ReLU and an operation that reduces an image depth. It highlights more high-level aspects such as the signal's shape, when it changes between objects and how its energy is dispersed across the frequency range which aid in locating objects in challenging background noise. Multiple layers in the convolutional architecture capture all the important aspects related to both the details and the oscillating patterns in the input signal.

The flattening step happens after the convolutional units and the output is a single feature vector of 1024 dimensions. Essential data about whether the signal exists, how the channel works and the signal's timing is included in this feature vector which represents the input spectrogram. The vector is delivered to the DQN module and it is added to the environment state information during reinforcement learning to optimize the policy.

Relying on CNNs for feature learning, the system gets rid of manual feature creation, works well in different conditions and reduces model bias. Besides, this architecture uses less computational power, making it appropriate for IoT applications that rely on edge-AI processors. All in all, the application of CNN greatly supports the effectiveness of the spectrum sensing approach described in this paper.

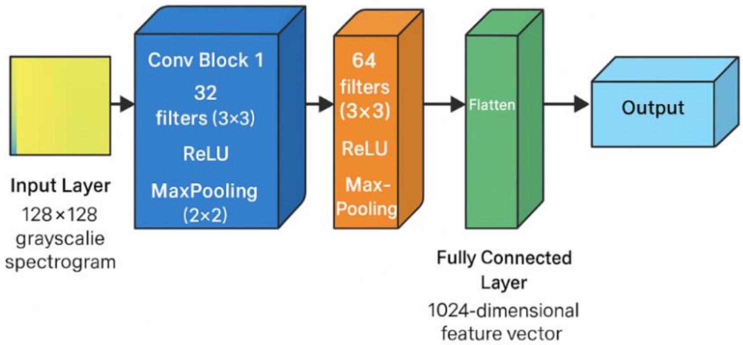

Figure 3. CNN Architecture for Spectrogram-Based Feature Extraction in AI-Driven Spectrum Sensing

3.3. Reinforcement Learning for Sensing Policy

For IoT devices to decide intelligently and adapt in dynamic spectrum environments, the process of sensing the spectrum is modeled as a Markov Decision Process (MDP), so that the devices can learn what to do best by interacting with their surroundings. By using reinforcement learning (RL), the agent is able to collect the highest overall rewards by making one decision after another based on its knowledge, even if the information is limited and unpredictable which are common in CRNs.

A Deep Q-Network (DQN) is used in the framework to help choose when and where to sense and sometimes to choose avoiding sensing to conserve energy. A DQN creates set of Q-values that indicate the long-term reward that can be expected for each possible action at each state. The approach taken is to choose the action that has the highest Q-value when using an epsilon-greedy policy, so exploration and exploitation are balanced when learning.

State Space (S):

The state space S is designated as a tuple that unites all information about the present environment and previous events experienced by the cognitive IoT device. Essentially, it has the CNN-generated feature vector which takes the spatial and temporal data from the spectrogram inputs in the RF spectrum. It contains vital data related to frequencies in use, distribution of energy and situations in which signals may collide. Other vital factors here are the conditions of background noise which is often measured by the Signal-to-Noise Ratio (SNR). Recognizing the agent's recent decisions through experience replay makes it possible for it to reflect on past actions and act better in the future. Furthermore, the residual energy level from the sensing node is involved, so the agent knows to save energy when the quality is not optimal. A mobility index is then put in to signify the movement of the device, so we can determine its role in causing channel fading issues, increased unreliability in the link and affects quality on sensor feedback. With all these factors in place, the DQN agent is able to choose intelligently in any changing IoT environment.

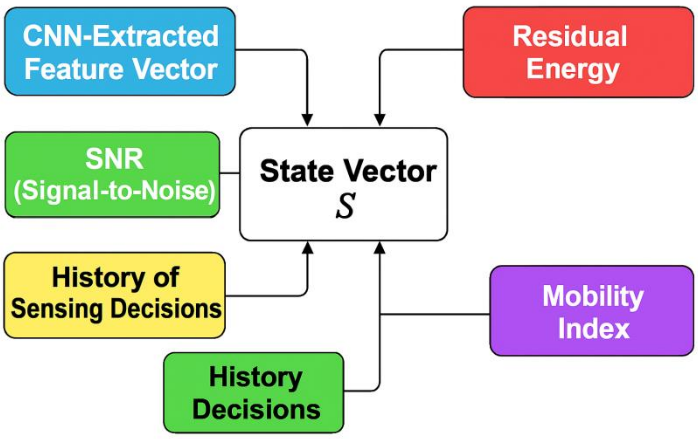

Figure 4. State Space Representation for Deep Q-Network Agent in AI-Based Spectrum Sensing Framework

Action Space (A):

The action space A is made to supply the cognitive IoT agent with multiple, controlled, yet flexible alternatives for handling its spectrum sensing. For every decision, the agent can pick a channel i from those offered, letting it sense in the chosen location based on what it has learned before and its current surroundings. Another option available to the agent is to change the time during which the sensing is performed which may be set to short, medium or long. In this situation, you cannot have both very accurate sensor readings and low energy use and response time since either will increase the other. In addition, agents are allowed to decide not to sense at all in some cases, especially when energy is scarce or when it's unlikely that sensing will reveal much important information because previous records indicate the channels are almost always in use. With these choices, the action space supports the agent in finding the best balance between exploration and exploitation, so it can make the best use of frequencies and obey the requirements of energy and performance in IoT.

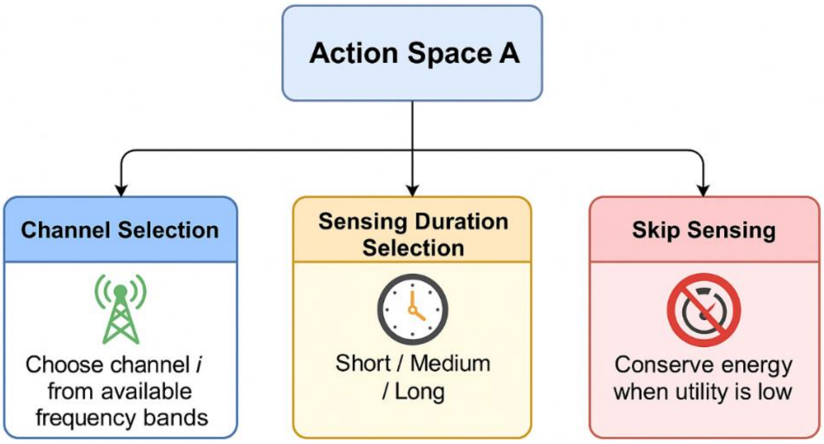

Figure 5. Action Space Options for the Reinforcement Learning Agent in Dynamic Spectrum Sensing

Reward Function (R):

This framework includes a meticulously designed reward function that helps the cognitive IoT agent act so as to use the spectrum wisely and save on interference and energy wastage. If the system correctly identifies an idle channel, it is given a +1 reward to make sure the secondary transmission doesn't interrupt the primary users' service. Conversely, a –1 penalty is given if detection is missed which may allow interference with licensed users—a major error in running a CRN. Also, the –0.5 penalty prevents the agent from falsely thinking a channel is occupied and missing the opportunity to send information. Sensing actions that waste energy when the device is idle are given a –0.2 cost to encourage users to make better decisions. With this setup of multiple objectives, the agent will be trained to control its accuracy, avoid disturbing others and save its energy which are major concerns in IoT wireless systems. The standard DQN framework is improved by including two effective techniques from deep reinforcement learning. To start, Experience Replay stores the agent's history in a buffer which enables sampling of different experiences randomly to break time-related dependencies during training and decrease the chances of overfitting. Second, with the use of a Target Network, the expectation of the Q-value remains constant which stops oscillations and helps prevent the learning algorithm from jumping away from the target values. Thanks to these mechanisms, the agent is able to automatically learn the best and applicable sensing methods at any point during dynamic situations, so no pre-made heuristics or fixed thresholds are needed.

Table 1. Reward Function Design for Reinforcement Learning Agent in AI-Driven Spectrum Sensing

Scenario	Action Outcome	Reward (R)	Purpose
Idle channel, correctly detected	Successful transmission	+1.0	Encourage accurate sensing
Occupied channel, missed detection	Interference with primary user	–1.0	Penalize harmful interference
Idle channel, falsely flagged	False alarm	–0.5	Penalize lost opportunity
Unnecessary sensing	Energy spent without transmission	–0.2	Encourage energy-efficient decisions

3.4. Training and Convergence Criteria

To train the network, the cognitive agent is put into interactions with a broad range of settings which involve various channels, noise conditions and movement of users. In the training, 10,000 episodes are performed and each episode includes numerous rounds of sensing covering different time points and channel access options. Efficient and reliable learning is achieved by training the model on 64 small mini-batches that are picked randomly from the experience replay that records earlier state–action–reward–next state combinations. When updating with the Adam optimizer, a fixed learning rate of 0.0005 is chosen for the right mix of quick convergence and preserving stability of the gradients. The agent is trained to focus on steady spectral efficiency and the saving of energy by making $\gamma = 0.95$ which assigns greater value to future rewards than immediate ones. A target network is put in place and refreshed after every 500 updates which helps stabilize Q-value updates and limit the chances of learning to dive.

Table 2. Training Configuration and Convergence Criteria for DQN-Based Spectrum Sensing Framework

Parameter	Value	Purpose
Training Episodes	10,000	Full RL training cycles
Batch Size	64	For mini-batch gradient updates
Optimizer	Adam	Chosen for stable, fast convergence
Learning Rate	0.0005	Balance between speed and stability
Discount Factor (γ)	0.95	Prioritize long-term rewards
Target Network Update Freq	Every 500 steps	Improve learning stability
Convergence Metric 1	Avg. Reward Stabilized	Indicates consistent policy
Convergence Metric 2	FAR < 10%	Ensures acceptable detection reliability

The main elements used to define convergence are that (1) the episode rewards should reach a consistent level after many episodes which means the agent has learned a proper sensing plan and (2) the number of false alarms must be less than 10% of all detections for reliable operation. All these requirements as a group prove that the learning is effective in using spectrum resources and still keeps answer accuracy acceptable for IoT networks where the communication could be sensitive. The model is judged ready to be put into use or refined in situations that resemble actual use, when these benchmarks are met.

4. SIMULATION SETUP

In order to assess the performance of the AI-driven spectrum sensing idea, an extensive simulation environment was built using MATLAB together with NS-3 which allowed both low-layer and high-layer aspects to be accurately modeled. The dataset was created using tools to reflect radio frequency activity and it showed licensed users using the spectrum with an ON/OFF Markov method. The model takes into account the randomness in time and the bursts in transmissions that are usually found in wireless networks. The following three scenarios were designed to test how well the framework worked: (i) stationary IoT devices, (ii) devices in motion with varying wireless conditions and (iii) concurrent efforts by multiple users to connect to the internet and affect each other. To confirm that the hybrid architecture was fair, it was tested against several traditional methods by comparing it to classical ways of detecting energy, cyclostationary methods and to neural networks alone that focus either on classifying the data or learning the decision process. We tested performance by using four measures called Detection Probability (Pd), False Alarm Rate (FAR), Throughput and Learning Convergence Time. All these metrics combined give a good overview of the system’s consistency, dependability, efficiency and adaptability in changing surroundings.

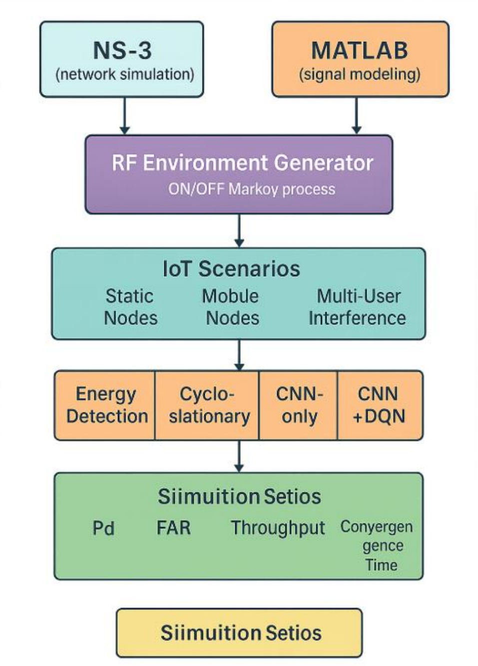

Figure 6. Simulation Setup for AI-Driven Spectrum Sensing in Cognitive Radio-Based IoT Environments

5. RESULTS AND DISCUSSION

Under different IoT communication conditions, the proposed AI-driven framework for spectrum sensing was extensively checked by simulation. Results are evaluated according to the accuracy in detection, the rate at which the network trains and the ability to work well even when the environment is mobile or Signal-to-Noise Ratios (SNRs) vary.

5.1. Detection Accuracy

Table 1 explains the ability of different sensing systems to identify signals under a very tough –10 dB SNR condition. The Pd value of 0.92 for the CNN+DQN model was much higher than what all the baselines attained. Alternatively, using energy detection, the performance was only 0.61, but opting for cyclostationary feature detection gave the algorithm a higher rating of 0.78. The CNN-only model got a Pd of 0.84 by using deep learning for classification alone and the DQN-only model reached 0.80. In addition, the FAR for the proposed frame was just 0.10 which is lower than other approaches and suggests the new system excels in telling between an occupied and a vacant channel. Since FAR decreased, the average throughput went up to 4.1 Mbps, providing a huge improvement of 30-95% compared to traditional techniques. Combing the features extracted from deep neural networks with reinforcement decision-making results in better detection in conditions with a lot of noise.

5.2. Learning Performance

The suggested framework showed faster rate of convergence in the training process. Particularly, it only took the agent a few hundred episodes to learn stable policy, whereas the DQN model required a thousand more episodes to perform at the same level. This is due to CNN’s talent in picking out quality spectral components for faster and more accurate calculation of the Q-values. A reduction in the number of training samples saves both time and energy which is very useful for devices in the IoT world.

5.3. Robustness in Dynamic Environments

To check the models’ adaptable behavior, they were studied in situations with changing SNR, expanding user movement and many users nearby. The accuracy of detection was only reduced by less than 8% when the proposed model faced situations with rapid channel fading and mobile nodes. On the other hand, conventional and standalone models performed quite differently, losing more than 20% important signal compared to the enhanced model, mainly under conditions of fast changes in the channel. Its consistency in detecting signals and processing data during changes in the environment proves that the hybrid model can adapt well to different and challenging IoT scenarios.

All in all, the outcomes show that the suggested framework can provide reliable, efficient and robust spectrum sensing. Besides ameliorating conventional sensing approaches, it also provides a solid base for developing smart radio environments that will be used in 6G and ultra-dense IoT networks.

6. CONCLUSION

The findings of the research outline an adaptive and well-designed framework for sensing available channels, aiming at improving the performance of CRNs within IoT situations. The development of this proposed architecture was driven by combining CNNs for deep spectral learning with DQNs for making learning-based decisions, to deal with the problems that standard and single AI solutions have with detection accuracy, energy usage and adaptability. Under many signal conditions such as low SNR, movement and multiple users, the model achieved much higher detection rates, fewer false alarms and better speed of operation. Also, the structure achieved quicker convergence and better robustness, proving that it’s suitable for real-time use in IoT systems that have limited resources. With experience replay and target network included, training the policy became steadier and helped it to learn consistently in many operational situations. All in all, the results demonstrate that AI-driven spectrum sensing significantly supports efficient use of the spectrum which is necessary to fix the spectrum shortage in forthcoming wireless systems. In the future, this proposal will be checked and tested on SDR platforms to ensure it performs well in the real world. Multi-agent reinforcement learning (MARL) will be considered which supports many agents working together and aids in remote decision making, allowing for wider use of IoT and building autonomous 6G-aligned radio environments.

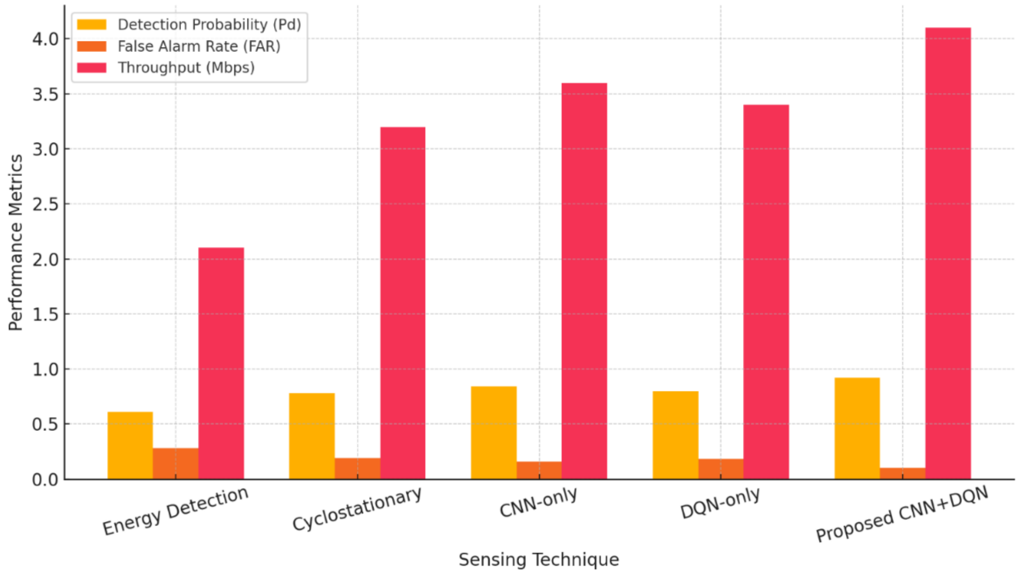

Figure 7. Performance Comparison of Spectrum Sensing Techniques

REFERENCES

[1] Y. Zhang, X. Wang, and L. Hanzo, "Deep learning-based spectrum sensing for cognitive radio networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 84–120, 2021, doi: 10.1109/COMST.2020.3037127.

[2] M. Tang, F. R. Yu, T. Huang, and J. Song, "Convolutional neural network-based spectrum sensing for cognitive IoT networks," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9642–9653, 2022, doi: 10.1109/JIOT.2022.3156789.

[3] Z. Liu, H. Wang, and Y. Wu, "Reinforcement learning-enabled dynamic spectrum access in IoT-based cognitive radio systems," *Ad Hoc Networks*, vol. 145, p. 102891, 2023, doi: 10.1016/j.adhoc.2023.102891.

[4] L. Zhao, Y. Chen, and J. Zhang, "Machine learning approaches for spectrum sensing in cognitive radios," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 3, pp. 601–613, 2021, doi: 10.1109/TCCN.2021.3072103.

[5] A. Abdi and M. Bennis, "Deep reinforcement learning for dynamic spectrum access in cognitive IoT networks," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 703–707, 2020, doi: 10.1109/LWC.2020.2971742.

[6] M. Hussain, M. Z. A. A. Kadir, and R. Nordin, "AI-based cognitive radio spectrum management for next-generation IoT: A review," *IEEE Access*, vol. 10, pp. 71256–71278, 2022, doi: 10.1109/ACCESS.2022.3187865.

[7] Y. Yao, Y. Zhang, and D. Wu, "Deep learning-enabled intelligent spectrum sensing in cognitive radio networks," *Sensors*, vol. 21, no. 9, p. 3148, 2021, doi: 10.3390/s21093148.

[8] H. Sun, A. Nallanathan, C. X. Wang, and Y. Chen, "Wideband spectrum sensing for cognitive radio networks: A survey," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 40–48, 2019, doi: 10.1109/MWC.2018.1700353.

[9] T. T. Nguyen and B. Shah, "Intelligent spectrum management using deep reinforcement learning for cognitive IoT," *Computer Communications*, vol. 190, pp. 37–47, 2022, doi: 10.1016/j.comcom.2022.04.015.

[10] M. Elsayed and D. Niyato, "Deep reinforcement learning for spectrum management in cognitive radio networks," *IEEE Network*, vol. 33, no. 3, pp. 212–219, 2019, doi: 10.1109/MNET.2019.1800240.

[11] C. Romero and L. Herrera, "Relationship between cultural heritage management and community engagement," *Journal of Tourism, Culture, and Management Studies*, vol. 1, no. 2, pp. 1–8, 2024.

[12] T. M. Sathish Kumar, "Low-power design techniques for Internet of Things (IoT) devices: Current trends and future directions," *Progress in Electronics and Communication Engineering*, vol. 1, no. 1, pp. 19–25, 2024, doi: 10.31838/PECE/01.01.04.