Optimizing the Energy Detection Based Spectrum Sensing through Bernstein Polynomial Approximation and Deep Learning

M. Subbarao [1,*] and N. Venkateswara Rao [2]

[1] Electronics and Communication Engineering, Acharya Nagarjuna University Guntur, Andhra Pradesh, India
[2] Electronics and Communication Engineering, Bapatla Engineering College, Bapatla, Andhra Pradesh, India
Email: subbu.vdp@gmil.com (M.S.); vrao68@gmail.com (N.V.R.)
*Corresponding author

***Abstract*—In the context of energy-based spectrum sensing, effective noise reduction is paramount to enhance the Signal-to-Noise Ratio (SNR) and improve the probability of detection, particularly in cognitive radio networks. This paper presents a novel approach that synergistically combines Fractional Bernstein Approximation, Fast Fourier Transform (FFT), and Convolutional Neural Networks (CNN) to robustly detect signals under noisy conditions. The Fractional Bernstein Approximation is employed as a pre-processing step to smooth the signal and mitigate noise effects. FFT is then utilized to transform the signal into the frequency domain, where CNN is applied to extract features that differentiate between signal and noise. Performance metrics such as SNR improvement, probability of detection, and computational efficiency are analysed under different noise scenarios. Our results demonstrate that the proposed method significantly outperforms other approaches, particularly in low SNR environments, offering a robust and scalable solution for spectrum sensing. The suggested approach is compared in this work against a number of other methods, including CNN with windowing segmentation for Quadrature Phase Shift Keying (QPSK) and 8PSK modulations and cyclisation features. The combination of CNN features and QPSK signals with the Bernstein polynomial approximation maintains a higher P_d (~96%) at lower SNRs (−10 dB) than cyclisation features.**

***Keywords*—fractional bernstein approximation, fast fourier transform, convolutional neural networks, windowing segmentation, cyclisation features**

I. Introduction

Cognitive Radio (CR) has emerged as a promising paradigm to address spectrum scarcity by enabling the dynamic utilization of unused spectral bands. It allows Secondary Users (SUs) to opportunistically access the licensed spectrum of Primary Users (PUs) without causing harmful interference [1, 2]. The effectiveness of CR systems critically depends on accurate spectrum sensing, which determines the presence or absence of PU activity. Among various sensing techniques, energy detection has gained popularity due to its simplicity, low implementation cost, and compatibility with diverse signal environments. However, its performance degrades significantly in low Signal-to-Noise Ratio (SNR) conditions, under fading channels, and in the presence of noise uncertainty [3]. To address these limitations, several alternative approaches have been proposed, including cyclisation feature detection [4], autocorrelation-based detection [5, 6], matched filtering [7, 8], and noise power-based dynamic threshold estimation [9]. While these methods enhance the probability of detection (P_d) and reduce the probability of false alarm (P_f), they often demand prior knowledge of PU signals or incur higher computational complexity. Fourier transform–based spectrum sensing has also been extensively explored, offering efficient frequency-domain analysis. Nevertheless, conventional FFT methods suffer from spectral leakage and resolution loss due to windowing effects, particularly when dealing with weak signals. Overlapping FFT structures have been investigated as a potential remedy [10]. Beyond Fourier methods, Time-Frequency Distributions (TFDs) such as the Short-Time Fourier Transform (STFT), Continuous Wavelet Transform (CWT), and S-Transform provide improved time–frequency localization. Despite their advantages, these methods face inherent challenges, including resolution trade-offs, cross-term interference, and high computational overhead [11, 12]. Signal segmentation plays a critical role as a pre-processing step in enhancing spectral estimation accuracy. Traditional segmentation techniques often lack robustness when handling nonstationary or noisy signals. Recently, polynomial-based methods such as Fractional Bézier Bernstein Polynomial (FBBP) segmentation have shown promise in improving the representation of nonstationary data, offering smooth approximations and noise resilience.

Motivated by these limitations, this study proposes a novel energy-based spectrum sensing framework that integrates three complementary components: (i) FBBP segmentation for improved time-frequency representation, (ii) FFT for efficient spectral analysis, and (iii) a Convolutional Neural Network (CNN) for robust feature extraction and classification. The proposed framework

Manuscript received May 21, 2025; revised July 14, 2025; accepted July 22, 2025; published January 9, 2026.

aims to enhance detection probability and noise robustness under challenging CR scenarios. The remainder of this paper is organized as follows. Section II presents a detailed literature review on deep learning–based spectrum sensing, segmentation methods, and FFT-based sensing approaches. Section III provides a brief overview of the conventional energy detector. Section IV introduces the proposed method, which combines FBBP segmentation, CNN-based feature extraction, and FFT analysis. Section V discusses experimental results and performance comparisons with existing methods. Finally, Section VI concludes the work and outlines potential directions for future research.

II. Literature Survey

Energy detection has long been regarded as one of the most widely adopted spectrum sensing techniques in cognitive radio networks due to its low implementation complexity and adaptability to diverse signal environments. However, its primary limitation lies in the reliable detection of signals at low Signal-to-Noise Ratio (SNR) under conditions of unpredictable noise variance. Classical fast Fourier Transform (FFT)-based detectors improve detection performance when applied to high discrete frequency bins, but this enhancement comes at the cost of degraded frequency resolution [13]. To address these shortcomings, recent efforts have focused on augmenting FFT-based methods. For instance, a parallel FFT framework with dynamic sample weighting guided by machine learning has been proposed to enhance frequency resolution, detection probability (P_d), and adaptability under low SNR conditions. Nevertheless, the increased architectural and implementation complexity poses challenges for practical deployment [13]. Similarly, Yadav *et al.*[14] proposed a hardware-efficient design integrating windowing, averaging, and polyphase filters to reduce computational load without significantly compromising detection accuracy.

In parallel, Deep Learning (DL) approaches have been increasingly applied to overcome the limitations of traditional energy detection. A notable example is the hybrid CNN–Transformer architecture proposed in Ref. [15], which leverages Convolutional Neural Networks (CNNs) for local feature extraction alongside transformer blocks for long-range dependency modeling, thereby improving spectrum utilization efficiency.

Signal segmentation, an essential step in pre-processing and detection, plays a critical role in FFT-based sensing. Early methods relied on statistical hypothesis testing (e.g., t-test and F-test [16]) and clustering algorithms [17, 18]. While these techniques performed reasonably well for stationary or noise-free signals, they exhibited high sensitivity to noise and often required extensive parameter tuning. To address these limitations, wavelet-based segmentation methods, particularly the Wavelet Transform Modulus Maxima (WTMM) approach [19, 20], became popular due to their ability to capture abrupt signal variations across multiple scales. However, their performance depends heavily on the choice of wavelet basis and decomposition depth.

Hybrid methods emerged as a means to combine the strengths of multiple techniques. For example, the integration of fuzzy C-means clustering with wavelet analysis demonstrated improved robustness in noisy environments [21], while discrete wavelet transforms with Principal Component Analysis (DWT-PCA) improved computational efficiency in biomedical applications [22]. More recently, adaptive segmentation methods such as sliding window approaches [23], Temporal Convolutional Networks (TCNs) [24], and CNN-based models [25–30] have achieved high segmentation accuracy in non-stationary and complex signals. Despite these advances, most of these techniques have been validated primarily on speech, biomedical, or synthetic datasets, with limited evaluation in real-world radio environments. Across the literature, three major methodological trends can be observed:

- FFT enhancement and hybridization through parallel processing, adaptive thresholds, and integration with machine learning techniques.
- Deep learning-based segmentation and detection, particularly models that exploit time-frequency representations for improved performance.
- Multiresolution and statistical fusion methods, such as wavelet–clustering hybrids, that enhance robustness under noisy conditions.

Despite these advancements, three important research gaps remain. First, handling low SNR and noise uncertainty continues to be a fundamental challenge, with classical FFT detectors proving inadequate and ML/DL methods hindered by training and hardware complexity. Second, the generalizability of segmentation methods is limited, as most have not been validated in real cognitive radio environments. Finally, the use of polynomial-based approximations in spectrum sensing remains largely unexplored. Bernstein polynomials, in particular, offer smooth and flexible spectral approximations that can effectively handle noise and irregularities while reducing computational complexity [31]. Their application to FFT-based energy detection presents a novel and underexplored research direction.

III. Conventional Energy Detection Method

In Cognitive Radio (CR) networks, energy detection continues to serve as a fundamental spectrum sensing approach for identifying the presence of primary (licensed) users within a specified frequency band. The primary goal of this technique is to detect unoccupied spectral regions, commonly referred to as spectrum holes, which can subsequently be exploited by secondary (unlicensed) users without introducing harmful interference to primary transmissions. The energy detection process typically involves four sequential stages: (i) signal acquisition, (ii) pre-processing, (iii) energy computation, and (iv) threshold comparison. As illustrated in Fig. 1, a digital energy detector consists of several core components, including a noise pre-filter (low-pass filter), an Analog-to-Digital Converter (ADC), a square-law device, an averaging (integration) unit, and a decision-making module [32]. The pre-filter mitigates out-of-band noise

and adjacent-channel interference, while the ADC transforms the continuous-time signal into discrete samples, enabling further spectral analysis—often through a fast Fourier Transform (FFT). The square-law device then estimates the instantaneous signal energy, which is subsequently averaged over a defined observation interval. Finally, the mean energy is compared against a predetermined threshold to infer the presence or absence of a primary user. Although energy detection is computationally efficient and conceptually straightforward, its performance is significantly degraded under low Signal-to-Noise Ratio (SNR) conditions and in scenarios characterized by noise uncertainty. These limitations pose critical challenges for reliable spectrum sensing in practical CR implementations, motivating ongoing research into advanced detection strategies and robust statistical models.

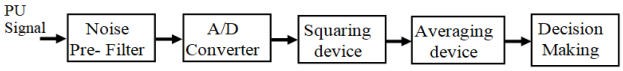

Fig. 1. Block diagram of conventional energy detection method.

The signal detection at the secondary user can be described as a binary hypothesis testing problem with noise, depending on whether the original user is idle or busy is given by.

Hypothesis 0 (h_0): signal is not present.

Hypothesis 1 (h_1): signal is present.

After sampling the received signal, y, the nth sample $y(n)$ can be expressed as [33]:

$$\Rightarrow y(n) = \begin{cases} n_0(m) & : h_0 \\ x(m) + n_0(m) & : h_1 \end{cases} \quad (1)$$

$y(n)$: received sample at time n, $x(m)$: PU signal sample, $n_0(m)$: additive white Gaussian noise (AWGN). The likelihood ratio for the binary hypothesis test in Eq. (1) can be obtained by applying the Neyman-Pearson criterion to the hypothesis issue. it is given by:

$$\Rightarrow \Lambda_{LR} = \frac{f_{y|h_0(x)}}{f_{y|h_1(x)}} \quad (2)$$

Following appropriate filtering, sampling, squaring, and integration in digital implementation, the energy detector's test statistic is provided by:

$$\Rightarrow \Lambda = \sum_{m=1}^{N} |y(m)|^2 \quad (3)$$

where N is number of samples. With a Large N, improved detection is achievable at low SNR. For P_d and P_f, the Central limit Theorem (CLT) provides better approximations. This means that SNR≤ -20dB can be roughly expressed as (e.g., −20 dB), 1+2 γ = 1 or 1+ γ =1. The approximated test statistics at low SNR is expressed as [34]:

$$\Rightarrow \Lambda_{Low} = \begin{cases} \mathcal{N}(N(2\sigma_w^2,), N(2\sigma_w^2)^2) & for\ h_0 \\ \mathcal{N}(N(2\sigma_w^2,)(1+\gamma), N(2\sigma_w^2)^2) & for\ h_1 \end{cases} \quad (4)$$

An established threshold ⅄ is needed to determine whether or not there is a primary user signal. This threshold ⅄ is used to calculate P_d and P_f performance. Because the threshold might range from 0 to ∞, choosing the right one is crucial for an appropriate operating threshold. Both P_f and P_d increase when λ lowers, and both P_f and P_d decrease when λ increases. The chosen threshold on for a given constant P_f can be obtained by using [45]:

$$\Rightarrow \lambda_f = \left(Q^{-1}(P_f)\sqrt{\mathrm{N}})\sqrt{N}2{\sigma_w}^2\right) \quad (5)$$

Another crucial factor in achieving higher false alarm and detection probabilities (P_f and P_d) is the number of samples (N) according to the specified observation frame. The function of the SNR determines the lowest quantity of samples needed for a particular detection probability (P_d) and false alarm probability (P_f). The necessary quantity of samples (M) is provided by:

$$\Rightarrow \mathrm{M} = [Q^{-1}(P_f) - Q^{-1}P_d\sqrt{2\gamma+1}\,]^2\,\gamma^{-2} \quad (6)$$

The necessary number of samples can be roughly calculated using the oscillatory noise at low SNR, is provided by:

$$\Longrightarrow M \simeq \frac{[Q^{-1}(P_f) - Q^{-1}(P_d)]^2}{\left[\gamma - \left(\rho - \frac{1}{\rho}\right)\right]^2} \quad (7)$$

where ρ is a parameter that can represent the noise signal's uncertainty. A detector's practical energy test statistic, which makes use of Noise Power (NP), which is estimated as:

$$\Rightarrow \Lambda_{NP} = \frac{1}{2\tilde{\sigma}_w^2 N}\sum_{n=1}^{N}|s(n)|^2 \quad (8)$$

where, $2\tilde{\sigma}_w^2$ is the estimated noise variance.

The SNR in the case above is obtained as:

$$\Rightarrow \gamma_{min} = \frac{[1-Q^{-1}(Pd)\sqrt{\emptyset}]}{[1-Q^{-1}(Pf)\sqrt{\emptyset}} - 1 \quad (9)$$

where, $\emptyset = \left(\frac{\tilde{\sigma}_w^2}{\sigma_w^2}\right)$:

As with hypothesis under $H0$, noise can be calculated in practice using noise-only samples. When K is the total samples that contain simply noise, then $\emptyset$ can be stated as:

$$\Rightarrow Q = \sqrt{\frac{(N+K)}{NK}} \quad (10)$$

In Energy Detector (ED)-based spectrum sensing, the number of samples and the threshold value represent the two most critical design parameters. These factors directly influence the detector's performance by affecting the noise variance estimation and the achievable Signal-to-Noise Ratio (SNR). In conventional ED implementations, particularly under low-SNR conditions, an infinite number of samples would theoretically be required to achieve the desired probability of detection (P_d) and probability of false alarm (P_f,) for a fixed threshold value. Consequently, determining an optimal threshold necessitates highly precise noise variance estimation, which, if inaccurate, may lead to an increased false alarm rate. A major limitation of energy-based detection techniques lies in their reduced reliability under low-SNR regimes with uncertain noise variance. Specifically: 1). Spectral Resolution Trade-off: The use of Fast Fourier Transform (FFT)-based implementations demands a large number of

samples to achieve finer frequency resolution, which in turn reduces temporal resolution. 2). Stochastic Signal Characteristics: The inherently random nature of the received signal, coupled with uncertainties in threshold selection, degrades detection performance, particularly in low-SNR scenarios. 3). Complexity–Performance Balance: While probability-based ED schemes can achieve near-optimal performance with relatively low computational complexity, they often require extensive training procedures, thereby limiting practical applicability.

These limitations highlight the fundamental challenges associated with ED-based spectrum sensing, particularly in environments characterized by low SNR and unpredictable noise variance, and emphasize the need for more robust detection strategies in cognitive radio networks.

IV. Methodology

Spectrum interpolation achieves optimal performance when the input signal is windowed prior to Fast Fourier Transform (FFT) computation. The application of windowing functions-such as Gaussian, Blackman, and Hanning-enhances the interpolation accuracy by reducing spectral leakage. In particular, combining a Gaussian interpolation scheme with a Gaussian window has been shown to yield significant improvements in spectral estimation accuracy. The standard procedure involves generating N windowing coefficients and multiplying them with the FFT input samples. The application of a window function can also be interpreted as a form of signal segmentation, wherein a continuous signal is divided into smaller segments, each representing a localized portion of the original signal. Signal segmentation, however, poses a fundamental challenge: the accurate determination of segment boundaries. Inefficient segmentation often leads to distortion in the time-frequency representation, thereby limiting the resolution of evolved Time–Frequency Distributions (TFDs). A persistent drawback of most TFD approaches is the inherent trade-off between time and frequency resolution. Inadequate resolution may cause the loss of critical spectral components, which directly reduces the probability of detection (P_d). This issue is particularly pronounced under low Signal-to-Noise Ratio (SNR) conditions, where missing spectral features further degrade detection reliability.

A. Specific Problems Addressed

- Noise and Signal Irregularities: Conventional segmentation methods often exhibit limited robustness in the presence of noise and irregular signal characteristics. These limitations result in reduced accuracy and unreliable performance in practical scenarios. Therefore, a smooth and adaptable approximation framework is required to effectively mitigate noise effects and accommodate irregularities in the signal structure.
- Inefficiencies in Frequency Domain Transformation: The direct application of the Fast Fourier Transform (FFT) to noisy signals frequently introduces spectral leakage, thereby obscuring relevant frequency components and degrading feature extraction quality. To address this, appropriate pre-processing techniques are essential to suppress noise, enhance the visibility of significant spectral features, and ensure a more reliable frequency-domain representation.
- Challenges in Complex Pattern Recognition: Manual feature extraction from FFT spectra is often inadequate for capturing intricate or non-linear signal patterns. Such limitations hinder the classification performance in tasks requiring high sensitivity to subtle variations. Consequently, an automated and efficient feature extraction approach is necessary to improve pattern recognition capabilities and enhance overall classification accuracy.
- Limitations of Traditional Energy Detection: Classical energy detection methods are highly susceptible to false alarms and missed detections due to noise fluctuations and signal irregularities. This reduces their reliability in accurately identifying the presence or absence of signals based on energy patterns. A more robust detection mechanism is therefore needed to provide consistent performance under adverse channel conditions.

B. Proposed Solution

This work addresses these problems through the following steps:

- Signal Segmentation Using Bernstein Polynomials:
- Employing Bernstein polynomials for signal segmentation provides smooth and flexible approximations, effectively handling noise and irregularities.
- FFT and Pre-processing: Transforming the segmented signals to the frequency domain using FFT and applying pre-processing techniques (e.g., logarithmic scaling and smoothing) to enhance spectral features and reduce noise

  CNN for Feature Extraction: Utilizing a CNN to automatically extract meaningful features from the pre-processed FFT output, leveraging the CNN's ability to learn complex patterns and improve classification accuracy.
- Energy Detection: Performing energy detection on the classified segments to accurately identify the presence or absence of signals based on predefined energy thresholds, ensuring reliable spectrum sensing.

This study proposes an integrated framework for signal segmentation and spectrum sensing, as illustrated in Fig. 2, which combines Bernstein polynomial approximation, Fast Fourier Transform (FFT), Convolutional Neural Networks (CNN), and energy detection techniques. The methodology begins with signal segmentation using Bernstein polynomials, which offer smooth and adaptable approximations, thereby reducing the adverse effects of noise and signal irregularities. Following segmentation, each signal segment is transformed into the frequency domain through the FFT.

To ensure reliable spectral representation, the FFT output undergoes a pre-processing stage that employs logarithmic scaling and smoothing techniques. This step

enhances salient spectral features while suppressing noise, thus facilitating more effective feature extraction. The pre-processed spectral data are then analyzed by a CNN, which automatically extracts discriminative features and identifies complex spectral patterns. The CNN's ability to learn hierarchical representations significantly improves classification performance compared to manual feature engineering approaches. Subsequent to feature extraction, energy detection is applied to the classified segments. The energy of each segment is computed, and a threshold-based decision rule is employed: if the calculated energy exceeds a predefined threshold, the presence of a signal is confirmed; otherwise, the segment is classified as signal-absent.

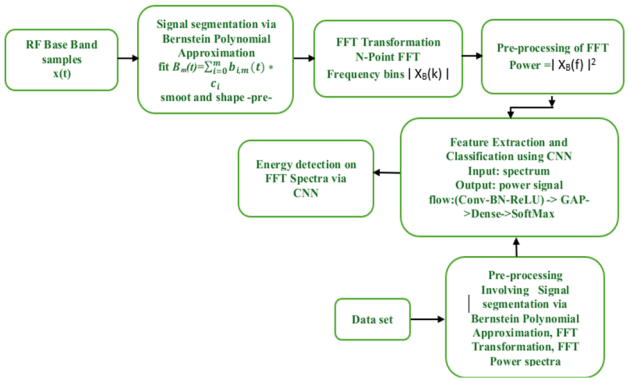

Fig. 2. Block diagram of proposed comprehensive approach.

By integrating Bernstein polynomial-based segmentation, FFT transformation, CNN-driven feature extraction, and robust energy detection, the proposed framework enhances both the accuracy and reliability of spectrum sensing. This holistic approach effectively addresses challenges posed by noise, spectral leakage, and complex pattern recognition, thereby enabling precise segmentation and efficient detection of energy patterns. The results demonstrate that the methodology offers a practical and resilient solution for spectrum sensing in modern communication systems.

Notations and variables used in text and equations are described below

- $x[n]$: Discrete-time received baseband samples.
- $x(t)$: Continuous-time baseband signal.
- $B_m(t) = \sum_{i=0}^{d-1} B_{i.m}(t) \times c_i$: Bernstein polynomial approximation of order m,
- $X[k]$: k^{th} FFT bin of the signal.
- $||X[k]|$: Magnitude spectrum.
- $|X[k]|^2$: Power spectrum.
- T: Energy detection statistic.
- λ: Decision threshold, set from a desired false-alarm probability P_f.
- H_0: Hypothesis — channel is idle (no signal).
- H_1: Hypothesis — channel is occupied (signal present) signal segmentation.

Signal segmentation refers to the process of dividing a continuous signal into smaller segments, each of which captures distinct characteristics of the original signal. A central challenge in this process lies in determining appropriate segment boundaries that accurately reflect the underlying structure and dynamics of the signal. Conventional segmentation approaches often struggle to detect abrupt changes or rapid transitions, thereby limiting their ability to represent local variations effectively. To address this limitation, the present work employs a modified Fractional Bézier-Bernstein Polynomial (FBBP) method for signal segmentation. In this approach, the signal is modeled as a piecewise polynomial function, where each segment is represented by a fractional Bézier curve. A Bézier curve, in general, is a parametric curve defined by a set of control points that govern its curvature and overall shape. Within the proposed segmentation framework, points of inflection in the signal are identified and utilized as control points. This enables the method to more precisely capture local structural variations and dynamic behaviors. A point of inflection corresponds to a location where the curvature of the curve changes direction. Identifying such points involves detecting critical points in the signal and evaluating whether the concavity changes at those locations. By incorporating inflection points as control parameters, the FBBP-based segmentation technique provides a more accurate and adaptive representation of signal dynamics, particularly in regions characterized by sharp transitions or irregular patterns.

C. Approximating a Non-stationary Signal Using Bezier Curve

The input consists of an oscillatory signal represented by the fractional Bezier Bernstein curve with degree d and data points $p0$, $p1$, ..., $pn-1$. A Fractional Bezier curve approximation of the signal that has segmentation and control points at points of inflection is the output. The traditional Bernstein-Bezier polynomial is extended to non-integer degree values by the fractional Bernstein-Bezier polynomial. The fractional Bernstein Bezier polynomial of degree d and order α is defined as:

$$\Rightarrow B_{d,a,i/N} = \sum_{k=0}^{d} \binom{d}{k} \frac{\Gamma(a+1)}{\Gamma(a-k+1)\Gamma(k+a-d)} \left(\frac{i}{N}\right)^k \left(1-\frac{i}{N}\right)^{d-k} \tag{11}$$

d is polynomial degree, α is polynomial order, i is index ranging from 0 to N and N is number of discrete parameter value. The fractional parameter α of the Bezier polynomial in Eq. (1) is between 0 and 1. When α is an integer, the fractional Bernstein-Bezier polynomial simplifies to the classical Bernstein-Bezier polynomial. When working with data that is not consistently spaced or when more precise control over the curve's shape is needed, it is very helpful. The polynomial for non-integer value of α is generalized using the gamma function; with the binomial coefficient $\binom{d}{k}$ added. The contribution of a control point to the polynomial evaluation at the discrete parameter value $\frac{i}{N}$ is represented by each term in the summation. The formula computes the index $\frac{i}{N}$ fractional Bezier Bernstein basis function, which establishes the i^{th} control point's influence on the overall curve. Ultimately, create a fractional Bezier curve that roughly represents the non-

stationary signal by adding together these basis functions for each control point.

In order to use the Fractional Bezier curve to approximate the non-stationary signal, first determine the Bernstein polynomial basis functions. For t in the interval [0, 1], the Bernstein polynomial basis function bi, $d(t)$ is given as:

$$\Rightarrow b_{i,d}(t) = \binom{d}{i} t^i (1-t)^{d-i} \quad (12)$$

where i and t are in the range [0, d] and [0, 1], respectively. By adding up the control points and adjusting for weight using the Bernstein polynomial basis functions, which are as follows: the first Bezier curve $B_d(t)$ is created.

$$\Rightarrow B_d(t) = \sum_{i=0}^{d-1} b_{i,d}(t) \times c_i \quad (13)$$

Points of inflexion are regarded as control points, therefore determining the values of 't' at which the 2nd derivative of $B_d(t)$ changes sign will yield the points of inflexion $x_0, x_1, \ldots, x_{m-1}$.

$$\Rightarrow \frac{d^2 B_d(t)}{dt^2} = \sum_{i=0}^{d-2} \frac{d^2 d_{i,b}(t)}{dt^2} \times c_i \quad (14)$$

Utilizing the acquired points of inflection, establish the control points. In a similar manner, by mapping $c_i = B_d(x_i)$ for i in [0, m−1], the control points for the Bezier curve are determined as the locations of intersection. Following basis function calculation, Using the Fractional Bezier Bernstein polynomial as well as the control points $c_i = p_i$, for $i \in [0, d-1]$, construct the Fractional Bezier curve $B_d(t)$. the Bernstein polynomial with fractional Bezier is given as

$$\Rightarrow B_d(t) = \sum_{i=0}^{d-1} B_{d,a,\frac{i}{N}}(t) \times c_i \quad (15)$$

Lastly, the Fractional Bezier curve may be evaluated as follows to approximate the non-stationary signal at data point p_i:

$$\Rightarrow B_d\left(\frac{i}{\mathrm{N}}\right) = \sum_{i=0}^{d-1} \left[\sum_{k=0}^{d} \binom{d}{k} \frac{\Gamma(a+1)}{\Gamma(a-k+1)\Gamma(k+a-d)} \left(\frac{i}{N}\right)^k \left(1 - \frac{i}{N}\right)^{d-k}\right] \times c_i \quad (16)$$

Determine the error that exists between the points on the Fractional Bezier curve $B_d\left(\frac{i}{\mathrm{N}}\right)$ and the original data points pi. One may quantify the error by use the squared Euclidean distance is given by.

$$\Rightarrow e = \sum_{i=0}^{n-1} \left| p_i - B_d\left(\frac{i}{N}\right)\right|^2 \quad (17)$$

- p_i: Original data point (sample of the non-stationary signal) at index i.
- $B_d\left(\frac{i}{\mathrm{N}}\right)$: Value of the frctional Bézier curve approximation of the signal at normalized position $\left(\frac{i}{\mathrm{N}}\right)$.
- d: Degree of the Bézier curve (order of approximation).
- i: Index of the data/sample point, where i=0, 1, …, n−1i = 0, 1, dots, n-1, i=0,1,…, n−1.
- N : Normalization constant (usually equal to the number of points or *maximum* index), used to map discrete index iii into the unit interval [0, 1].
- c_i : Control point coefficients (weights) for the Bézier curve *representation*. These are estimated during curve fitting.
- $\binom{d}{i} = \frac{d!}{k!(d-k)!}$: *Binomial* coefficient.
- a : Fractional *parameter* (controls the "fractionality" or flexibility of the curve, extending standard Bézier formulation).
- $\Gamma(\cdot)$: Gamma function, generalizing factorial (i.e.,$\Gamma(n) = (n-1)!$).
- n: Total number of *original* data points in the signal.
- e : Total squared Euclidean error (quantifies approximation *accuracy*).

Using a least square fitting method, modify the control points (ci) in the fractional Bezier curve $B_d(t)$ to minimize the error 'e'. Fig. 3 demonstrates how the fractional Bernstein approximation function is used to segment and approximate the synthetic signal $x(t)$. Within certain time periods, the signal varies with three unique multiple components. Eq. (14) is utilized to identify the points of inflection linked to these numerous frequency contents. Shown graphically on the approximate polynomial Bd(t) in Fig. 3. These points of inflection are used as control points for each instance (i/N) in the approximation Bezier curve given by Eq. (16) to get the segmented signal.

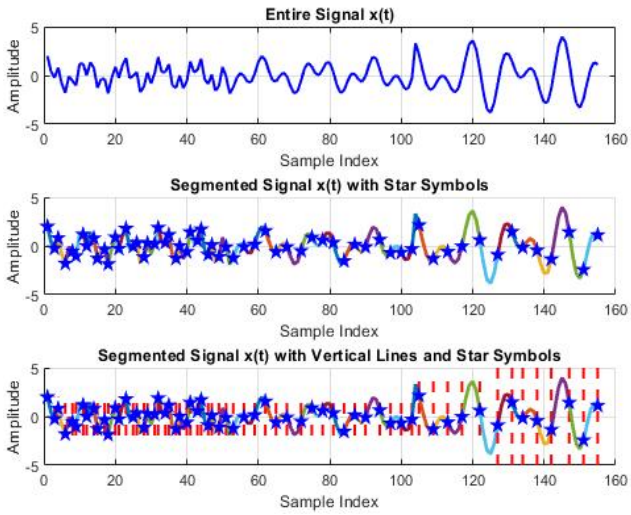

Fig. 3. Segmentation of signal using fractional Bernstein Bezier curve approximation.

D. Fast Fourier Transform (FFT)

Following signal approximation, the Fast Fourier Transform (FFT) is applied to evaluate the frequency-domain characteristics of the signal. Both the Bernstein polynomial framework and the FFT play distinct yet complementary roles in spectrum sensing, particularly in the context of energy detection. The FFT decomposes the time-domain signal into its constituent frequency components, thereby enabling the estimation of power levels across different frequency bands. This information is essential for determining whether a particular frequency band is occupied or vacant. Once the spectral components and their corresponding power levels are obtained through the FFT, Bernstein's inequality can be employed to further enhance the decision-making process. Specifically,

Bernstein's inequality provides probabilistic bounds that allow for the estimation of detection likelihood under noisy conditions. By quantifying the uncertainty associated with signal presence in the frequency spectrum, it contributes to more robust and reliable detection outcomes. Thus, while the FFT serves as a tool for transforming and analyzing the spectral content of signals, Bernstein's inequality supports statistical evaluation by strengthening detection reliability in noise-prone environments. Together, these methods form a complementary framework that improves the accuracy of energy detection-based spectrum sensing.

Let $B_d(t_n)$ be the discrete-time signal representation, where t_n is the n-th sample point and $B_d(t_n)$ is the discrete-time signal. The discrete signal has the following expression:

$$\Rightarrow B_d(t) = \sum_{i=0}^{d-1} B_{d,a,\frac{i}{N}}(t_n) \times c_i \quad (18)$$

where $B_{d,a,i/N}(t_n)$ the Bernstein basis polynomial with fractional Bézier parameters. c_i are the coefficients. The FFT of the signal $B_d(t_n)$ is denoted by $X(k)$, where k is the frequency bin index. The $X(k)$ represent the FFT of the signal $B_d(t_n)$ where k is the frequency bin index.

$$\Rightarrow X(k) = \sum_{n=0}^{N-1} B_d(t_n)\, e^{-j\frac{2k\pi n}{N}} \quad (19)$$

for k=0, 1, 2, …, N-1. Substitute $B_d(t_n)$ into above equation.

$$\Rightarrow X(k) = \sum_{n=0}^{N-1} \left(\sum_{i=0}^{d-1} B_{d,a,\frac{i}{N}}(t_n) \times c_i \right) e^{-j\frac{2k\pi n}{N}} \quad (20)$$

$$\Rightarrow X(k) = \sum_{i=0}^{d-1} * c_i \left(\sum_{n=0}^{N-1} B_{d,a,\frac{i}{N}}(t_n) \right) e^{-j\frac{2k\pi n}{N}} \quad (21)$$

The inner sum can be defined as $X_i(k)$, signifying the input of the *i-th* Bernstein polynomial to the FFT:

$$\Rightarrow X_i(k) = \sum_{n=0}^{N-1} B_{d,a,\frac{i}{N}}(t_n)\, e^{-j\frac{2k\pi n}{N}} \quad (22)$$

Thus, the $X(k)$ can represent as:

$$\Rightarrow X(k) = \sum_{i=0}^{d-1} \times c_i \times X_i(k) \quad (23)$$

The Bernstein Polynomial Approximation purpose is Smooth segmentation of the time-domain signal before FFT to reduce noise-induced fluctuations in energy detection. some of advantages of Bernstein Polynomial Approximation are: 1). Shape-preserving, 2). Low variance, 3). Numerical stability,4. Robust to low SNR. The Challenges in real-world systems are 1). Computational cost increases with polynomial order mmm if implemented naively, 2). Requires tuning mmm for trade-off between smoothing and detail retention, 3). Hardware implementation may need fixed-point approximations for speed.

E. Convolution Neural Network

In low-SNR scenarios, typically encountered in cognitive radio and dynamic spectrum access systems, two major challenges degrade the performance of traditional energy detection, 1). Noise Uncertainty, 2). Loss of Discriminative Features. To address these limitations, a Convolutional Neural Network (CNN) is incorporated in the proposed sensing framework. Convolutional Neural Network (CNN) addresses these limitations by: 1). Feature Learning Beyond Energy. 2). Pre-trained Knowledge for Noise-Invariant Detection. 3). Nonlinear Decision Boundaries.

In energy detection-based spectrum sensing, a Convolutional Neural Network (CNN) may be utilized to accurately determine whether or not a frequency band is occupied after calculating the signal's FFT. This is a suggested CNN architecture designed specifically for this job. In order to minimize dimensionality, pooling layers are added after a few convolutional layers to aid in the extraction of significant features from the FFT output. More intricate patterns are captured with the aid of the deeper convolutional layers. The output layer and fully linked layers ultimately yield the final categorization determination. With the ability to be adjusted depending on the particulars of the dataset, this model is made to handle the frequency-domain representation of the signal for spectrum sensing applications in an effective manner. Using the frequency-domain representation of signals acquired by the Fast Fourier Transform (FFT), a Convolutional Neural Network (CNN) architecture suited for energy detection-based spectrum sensing is proposed in this paper. The size of the FFT result, which is usually shown as a one-dimensional array, serves as the CNN's input. The first two 1D convolutional layers in the architecture include 32 and 64 filters, a kernel size of 3, and ReLU activation functions. A max-pooling layer with a pool size of 2 comes next. This is followed by another max-pooling layer for additional dimensionality reduction and deeper convolutional layers with 128 and 256 filters while keeping the same kernel size and activation function. After these layers' outputs are flattened, they are passed through fully linked layers that use ReLU activation and have 256 and 128 units, respectively. One neuron with a sigmoid activation function makes up the final output layer. Its purpose is to do binary classification, which tells us whether or not a certain frequency band is occupied. The model is assembled utilizing the Adam optimizer with binary cross-entropy as the loss function. Accuracy, precision, and recall metrics are used for assessment. This architecture offers a reliable solution for spectrum sensing in cognitive radio networks since it is especially designed to collect and analyze spectral data quickly.

Approximation segmentation and CNN Method (see Algorithm 1).

Algorithm 1:

Inputs:

- $x(t)$: Received signal over time t
- n: Degree of the Bernstein polynomial
- T: Observation period
- θ: Detection threshold
- NFFT: Number of FFT points
- CNN (·): Pre-trained Convolutional Neural Network model

Output:

- Decision D ∈{0,1}: 0 indicates the spectrum band is vacant, 1 indicates it is occupied.

Step 1: Compute the energy signal $E(t)$ over the observation period TTT.

Step 2: Normalize the time variable t to the interval [0, 1].

$t' = \frac{t - t_{min}}{t_{max} - t_{min}}$

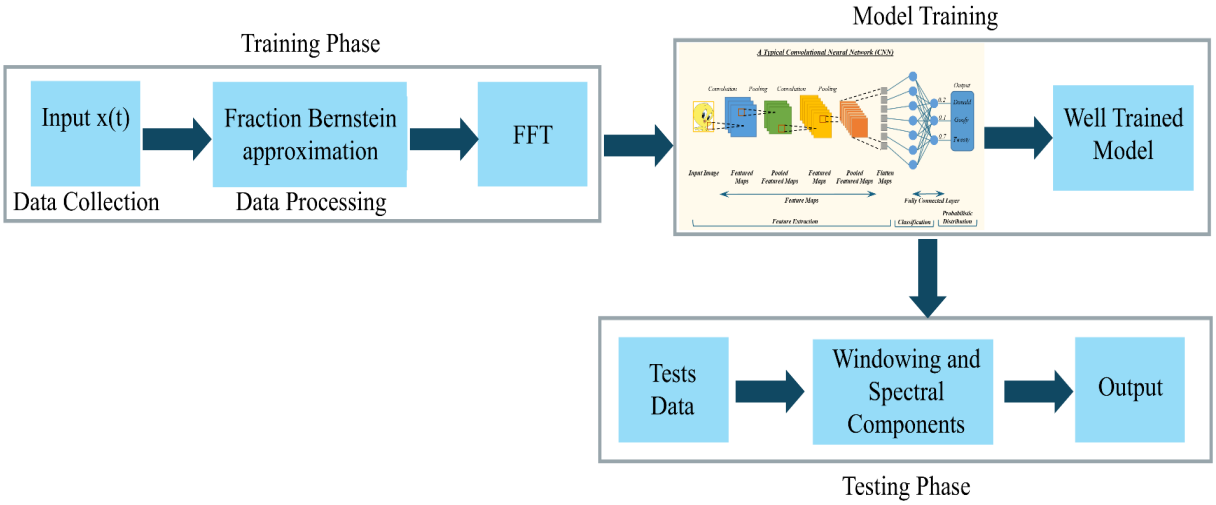

Fig. 4. Block diagram of proposed bernstein.

where t_{max} and t_{min} are the maximum and minimum time values in the observation period.

Step 3: Apply Bernstein polynomial approximation to the energy signal $E(t)$ obtain a smoothed version $B_n(E, t')$.

Step 4: Apply FFT to the Smoothed Energy Signal.

$X_B(f) = FFT(B_n(E, t'), N_{FFT})$.

Step 5: Calculate the Power Spectral Density (PSD):

$E_B(f) = |X_B(f)|^2$ Compute the power spectral density $E_B(f)$ from the magnitude squared of the FFT result.

Step 6: Normalize the approximated power spectral density $E_B(f)$ as required for CNN input.

Step 7: Reshape the normalized power spectral density $E_B(f)$ into the format required by the CNN (e.g., 1D array or 2D matrix).

Step 8: Extract Features Using CNN: Feed the power spectral density $E_B(f)$ into the CNN to extract relevant features and obtain a decision score or probability.

Step 9: Make Classification Decision:$D = \begin{cases} 1 & if\ CNN(E_B(f)) \geq \theta \\ 0 & if\ CNN(E_B(f)) < \theta \end{cases}$

Compare the CNN output to the detection threshold θ to determine if the spectrum band is occupied (1) or vacant (0).

Step 10: Return the Final Decision: Output D, indicating whether the spectrum band is occupied (1) or vacant (0).

F. Cyclostationary Feature Detection for Spectrum Sensing

By taking use of the periodicity present in the modulated signals, cyclisation feature identification is a spectrum sensing approach used in cognitive radio networks to identify the presence of Main Users (PU). This technique is based on the observation that most communication signals, because of their coding, multiplexing, and modulation techniques, display periodic statistical features like mean and autocorrelation, which give rise to cyclisation traits. There is periodic autocorrelation in modulated signals. The expression for the autocorrelation function is:

$$\Rightarrow R_x(t, \tau) = E[x(t)x^*(t - \tau)] \quad (24)$$

where $x^*(t)$ is the complex conjugate of $x(t)$ and τ is the time lag. If $R_x(t, \tau)$ is periodic with period T_0, the signal $x(t)$ is considered to be second-order cyclostationary. This is to be expressed as

$$\Rightarrow R_x(t, \tau) = R_x(t + T_0, \tau) \quad (25)$$

The Fourier transform of the cyclic autocorrelation function with respect to the time lag τ is known as the Spectral Correlation Density (SCD). It displays the correlation between a signal's spectral components at various cyclic frequencies α. The SCD $S_x(f, \alpha)$ is given by:

$$\Rightarrow S_x(f, \alpha) = \int_{-\infty}^{\infty} R_x^{\alpha}(\tau)\, e^{-j2\pi f\tau} d\tau \quad (26)$$

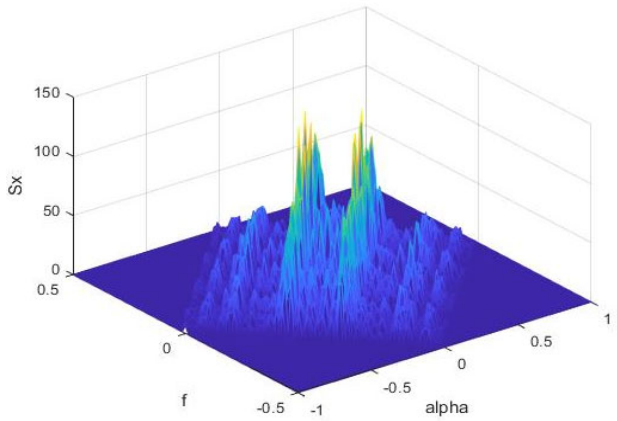

Fig. 5. Spectrum of cyclisation features for primary user signal without noise.

where f is the frequency and α is the cyclic frequency. The signal is represented in two dimensions by the SCD in both the frequency (f) and cyclic frequency (α) domains. To separate a modulated signal from noise, which usually lacks cyclisation characteristics, the SCD shows peaks at particular cyclic frequencies. Fig. 5 and Fig. 6 displays the cyclisation property of the signal sample sequence example.

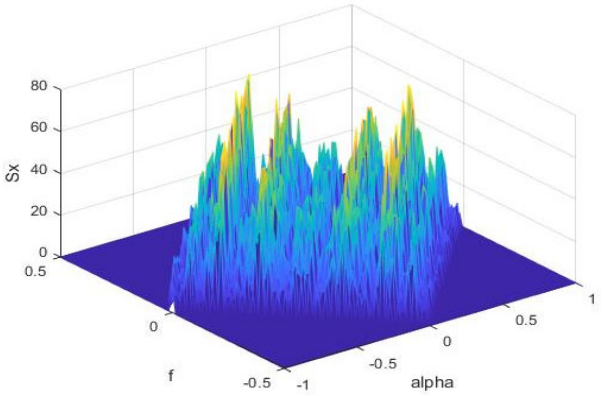

Fig. 6. Spectrum of cyclisation features for primary user signal with noise.

The spectral correlation density is examined by concentrating on the feature values primarily along the line corresponding to $\alpha = 0$ and $f = 0$ in order to effectively convey cyclisation characteristics for spectrum sensing. The spectral correlation is sampled 64 times by setting the spectral frequency (f) and cyclic frequency (α) to zero. This yields two 64 × 1 cyclisation feature vectors, $Cy.S_\alpha$ and $Cy.S_f$. The unique cyclic characteristics of the received signal are represented by these vectors. These cyclisation data are merged with energy and power spectrum features to generate a 64 × 4 feature matrix, which is then used to create a full feature set for spectrum sensing.

G. Segmentation of Signal Using Windowing

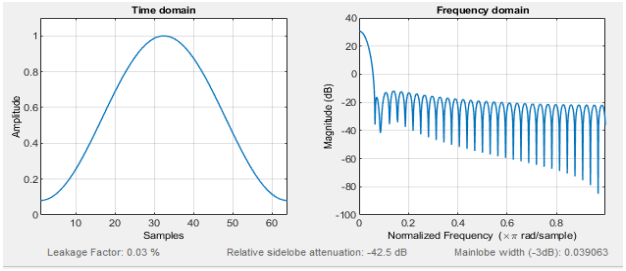

Fig. 7. Hamming window function and its spectrum.

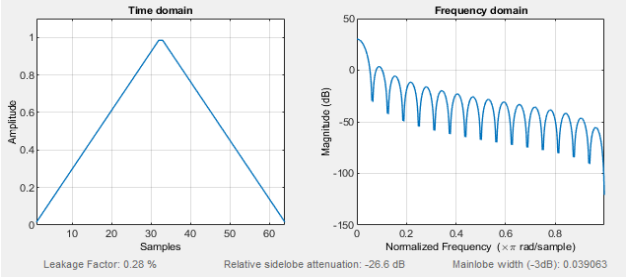

Fig. 8. Triangular window function and its spectrum.

However, because FFT processes the received signal arbitrarily, there are certain concerns with energy leakage surrounding the real signal due to the discontinuity in its extraction. Using windowing techniques is a widely recommended strategy to reduce energy leaks. How much of a suppressive impact it has depends on the particular window function that is used. This approach involves multiplying the received time-domain signal by the assigned window function. Concurrently, the signal that is ready for FFT is presented via the window processing technique. When a signal is segmented using windowing, a window function, such as a rectangular, Hamming, Hanning, or Blackman window, is applied to divide the signal into smaller, overlapping segments. Let the whole continuous-time signal be represented by x(t). By applying a window function w(t) of length N samples to this signal, a windowing approach of signal segmentation is performed. One definition for the segmented signal $x_w(t)$ is:

$$x_w(t)=x(t)\times w(t) \tag{27}$$

Some examples of windowing functions and associated spectra are shown in Figs. 7 and 8.

H. Neural Network Architectures and Training Settings

To promote transparency and reproducibility, the architectures of both the Convolutional Neural Network (CNN) and the Linear Neural Network (Linear Net) are explicitly documented. The structural specifications of each model are summarized in Table I and Table II, respectively, with accompanying explanatory notes provided to clarify their design choices and functional roles within the proposed framework.

TABLE I. CNN ARCHITECTURE

Layer	Parameters	Activation
Input	(1 × 64 × 2)	–
Convolution 1	4 filters, kernel=2×2, stride=1, pad=1	–
Max Pooling 1	Kernel=2×2, stride=1	–
Convolution 2	8 filters, kernel=2×2, stride=2, pad=1	–
Max Pooling 2	Kernel=2×2, stride=1	–
Fully Connected 1	128 neurons	Sigmoid
Fully Connected 2	84 neurons	Sigmoid
Fully Connected 3	48 neurons	Sigmoid
Output Layer	2 neurons (binary classification)	SoftMax

The Convolutional Neural Network (CNN), summarized in Table I, is designed to extract hierarchical spatial features from two-dimensional input signals. The convolutional layers are responsible for capturing localized patterns within the data, while the fully connected layers integrate these features into higher-level representations suitable for decision-making. Nonlinearity is introduced through the use of sigmoid activation functions, enabling the network to model complex relationships within the input. Finally, a SoftMax layer is employed at the output stage to provide probabilistic classification, allowing the input signals to be categorized into two distinct classes.

The Linear Neural Network (LinearNet), summarized in Table II, adopts a conventional feed-forward architecture. In this design, the extracted signal features are first flattened into a one-dimensional vector, which is then processed through successive fully connected layers to learn class boundaries. Although computationally efficient and structurally simple, the absence of convolutional layers limits its capacity to capture complex local patterns within the data. Consequently, LinearNet exhibits reduced representational power compared to the CNN, particularly in tasks requiring hierarchical feature extraction.

TABLE II. LINEARNET ARCHITECTURE

Layer	Parameters	Activation
Input	Flattened (64 × 2) = 128	–
Fully Connected 1	64 neurons	Sigmoid
Fully Connected 2	32 neurons	Sigmoid
Fully Connected 3	16 neurons	Sigmoid
Output Layer	2 neurons	SoftMax

I. Training Configuration (Common to Both Models)

To ensure fairness, both models were trained under similar conditions:

- Optimizer: SGD with momentum (0.9).
- Learning Rate: 0.01, decayed by StepLR (× 0.1 every 20 epochs).
- Batch Size: 16 (LinearNet), 32 (CNN).
- Epochs: 15 for LinearNet, 50 for CNN.
- Loss Function: CrossEntropyLoss (CNN), MSE with one-hot encoding (LinearNet).
- Data Split: Train 60%, Validation 20%, Test 20%.
- These standardized hyperparameters provide a reliable basis for comparing the two architectures.

J. Dataset Split and Parameter Tuning

The dataset was partitioned into three subsets: 60% for training, 20% for validation, and 20% for testing. This division ensured an adequate volume of data for model training while preserving independent validation and test sets for unbiased performance assessment. Hyperparameters, including the learning rate and batch size, were selected empirically based on validation outcomes to optimize model performance. While the CNN required a deeper configuration to achieve higher classification accuracy, the LinearNet demonstrated faster convergence but exhibited performance saturation at comparatively lower accuracy levels.

K. Convergence and Computational Complexity

Table III indicates the comparison the training duration, parameter count, and inference latency of both models while the Convolutional Neural Network (CNN) incurs higher computational demands during the training phase, it demonstrates superior detection accuracy, rendering it more appropriate for applications where reliability is of paramount importance. In contrast, the LinearNet exhibits faster training times and reduced computational overhead, making it a more suitable choice for lightweight or resource-constrained environments.

TABLE III. CONVERGENCE AND COMPLEXITY COMPARISON

Model	Training Time (50 epochs)	Parameters	Inference Latency (per sample)
CNN	−25 seconds (GPU)	−32,000	1.8 ms (GPU), 6.2 ms (CPU)
LinearNet	−10 seconds (GPU)	−10,000	1.1 ms (GPU), 4.8 ms (CPU)

L. Statistical Analysis

To address variability, each experiment was repeated five times with different random seeds. The averaged performance across runs is summarized in Table IV.

TABLE IV. STATISTICAL RESULTS (MEAN ± STANDARD DEVIATION)

Model	Validation Accuracy	Test Accuracy	P_d (±SD)	P_f (±SD)
CNN	91.3% ± 1.2%	90.6% ± 1.5%	0.942 ± 0.01	0.084 ± 0.008
LinearNet	85.7% ± 1.6%	84.9% ± 1.8%	0.902 ± 0.02	0.116 ± 0.010

As presented in Table IV, the Convolutional Neural Network (CNN) consistently outperforms the LinearNet across multiple trials, achieving a higher probability of detection (Pd) and a lower probability of false alarm (Pf). Moreover, the relatively smaller standard deviations highlight the stability and reliability of the CNN, thereby reinforcing its robustness in noisy environments.

V. RESULTS

The dataset preparation for the network's training is involved, two types of band-pass modulated signals—QPSK and 8PSK—with a carrier frequency of 10 kHz are simulated in MATLAB. A raised cosine filter with a roll-off factor of 0.5 is utilized as the pulse shaping filter. The signal travels across a Rayleigh fading channel with many paths. The delays vector is [0.001 s, 0 s]. It is assumed that the noise is Gaussian white noise. The signal is configured to while it travels across the AWGN channel, the SNR drops from -20 to 5. After running the received signal via the fraction Bernstein approximation, the spectral component FFT is found. Spectral components are where the features are taken from. After Bernstein approximation and before FFT, the received signal is sampled 1280 times after going through the AWGN and Rayleigh channels. The spectral components are then put into a feature extractor to create feature matrices. "Signal 0" and "signal 1" are the two types included in the train dataset. Only the feature matrixes pertaining to noise samples are contained in "signal 0," whereas the feature matrixes related to signals, comprising main signal and noise, are contained in "signal 1." There are around 5000 data points in each sort of signal. The test dataset is used to evaluate the performance of the spectrum sensing model. The model's performance may be assessed using two parameters: probability of detection (P_d) and probability of false alarm (P_f). For QPSK and 8PSK signals, the probability of detection (P_d) is computed for each test signal at different SNRs using simply CNN, CNN with windowing, and CNN with fraction Bernstein approximation. The various plots for the aforementioned are displayed in Fig. 9. The CNN+WD is represented as CNN with window segmentation and CNN+B as CNN with fraction Bernstein approximation segmentation. The probability of detection (P_d) is found to be nearly constant across all techniques at high SNRs (−5 dB to 5 dB), with 100% of P_d being attained. At Low SNR (−10 dB), the QPSK received signal with Bernstein approximation segmentation and CNN features gives better P_d nearly about 96% of P_d compared with QPSK with CNN. Overall, it appears that the greatest notable gain in spectrum sensing performance for both QPSK and PSK signals comes from combining CNN with

Bernstein polynomial approximation. It is discovered that the model performs pretty poorly for both signal types when the SNR is low (<15 dB). As shown in Fig. 9 and Fig. 10 shows SNR increases cause Pd to rise and Pf to fall. Pd will be greater than 80% and Pf will be less than 10% when the SNR is greater than −10.

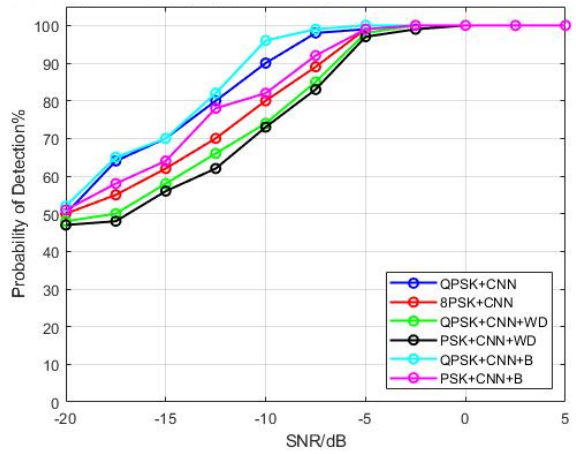

Fig. 9. Probability of detection (P_d) performance comparison between QPSK and 8-PSK in terms of CNN, CNN+ window segment and CNN +Bernstein approximation segmentation.

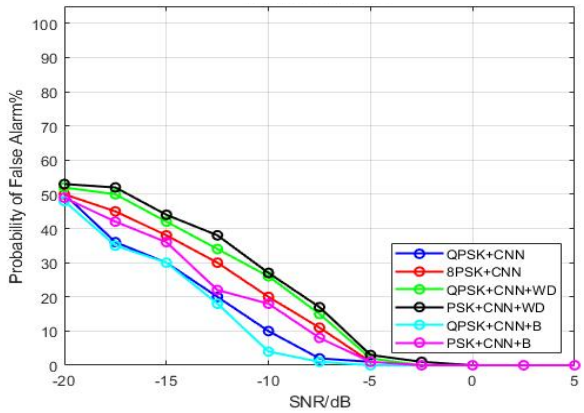

Fig. 10. Probability of False Alarm (P_f) performance comparison between QPSK and 8-PSK in terms of CNN, CNN+ window segment and CNN +Bernstein approximation segmentation.

The probability of detection (P_d) and probability of False Alarm (P_f) when SNR rises is seen in Fig. 11 and Fig. 12. These provide an analysis of the detection performance of several signal types, including cyclisation features, across a range of SNR situations. In comparison to cyclisation features, the state-of-the-art methods—such as CNN in conjunction with the Bernstein polynomial approximation—perform noticeably better at low SNR levels. It shows that for all investigated strategies; the detection probability approaches 100% at SNRs (above −5 dB). The combination of CNN features and QPSK signals with the Bernstein polynomial approximation maintains a higher P_d (~96%) at lower SNRs (−10 dB) than cyclisation features. P_d generally, rises with improved SNR, and better sophisticated techniques such as QPSK signals. Fig. 9 and Fig. 10 illustrate the detection performance in-case of 8PSK signal. The cyclisation curve demonstrates the extremely poor performance in both 8PSK and QPSK signal cases, where the P_d is around 50% between −15 and −10 dB SNR. The curves shown by proposed method (CNN+B) illustrate the more detection performance difference between QPSK and 8PSK modulations. When the system using CNN with Bernstein Approximation segments, the QPSK modulation achieves more than 20 % of P_d compared to the 8PSK modulation between −15 and −10 dB SNR.

Tables V and VI illustrate that at Low SNR Performance (–15 dB) the conventional Cyclisation Feature Detection technique demonstrates a detection rate of 48% at an SNR of –15 dB. While the CNN + Bernstein Approximation further boosts detection performance to 70%, the CNN + Windowing approach improves detection performance to 58%. This illustrates the value of utilizing CNN and the Bernstein approximation, which together capture more discriminative signal characteristics even at extremely low SNR. At SNR Performance (–10 dB) the Cyclisation feature detection rate stays low at 47% at −10 dB SNR. CNN + Windowing, on the other hand, considerably increases to 74%, while CNN + Bernstein Approximation reaches an even greater detection rate of 96%. This noteworthy increase demonstrates how reliable the CNN + Bernstein Approximation method is. At −5 dB the Both CNN-based techniques exhibit extremely high detection rates at an SNR of -5 dB, with CNN + Windowing achieving 98% and CNN + Bernstein Approximation 100%, respectively. While it still trails behind the CNN-based methods, the Cyclisation Features method also becomes better, highlighting the supremacy of deep learning-based approaches under high SNR settings.

TABLE V. DETECTION PERFORMANCE COMPARISON OF VARIOUS METHODS FOR QPSK SIGNAL

SNR in dB	Cyclisation Features	CNN+ windowing	CNN+ Bernstein approximation
−15	48%	58%	70%
−10	47%	74%	96%
−5	86%	98%	100%

The detection performance of QPSK and 8PSK signals was evaluated under Rayleigh fading and AWGN channels using three approaches: a baseline CNN, CNN with window segmentation (CNN+WD), and CNN with Bernstein polynomial approximation–based segmentation (CNN+B). The performance was assessed in terms of probability of detection (P_d) and probability of false alarm (P_f). As shown in Figs. 9 and 10, all methods achieve nearly perfect detection ($P_d \approx 100\%$) for SNR ≥ –5 dB, indicating that CNN-based approaches can reliably discriminate between signal and noise under moderate-to-high SNR conditions. However, significant performance differences emerge at low SNRs. At –10 dB, CNN+B attains approximately 96% P_d for QPSK, outperforming CNN+WD (74%) and cyclisation detection (47%). Even at –15 dB, CNN+B maintains 70% P_d for QPSK, while CNN+WD and cyclisation detection decline to 58% and 48%, respectively. A consistent modulation-dependent trend is observed: QPSK outperforms 8PSK across all

methods under low-SNR conditions. For instance, at –10 dB, CNN+B achieves 96% P_d for QPSK but only 82% for 8PSK, reflecting the reduced noise robustness of 8PSK due to its smaller Euclidean distance between constellation points.

TABLE VI. DETECTION PERFORMANCE COMPARISON OF VARIOUS METHODS FOR 8PSK SIGNAL

SNR in dB	Cyclisation Features	CNN+ windowing	CNN+ Bernstein approximation
−15	48%	56%	64%
−10	47%	73%	82%
−5	86%	97%	99%

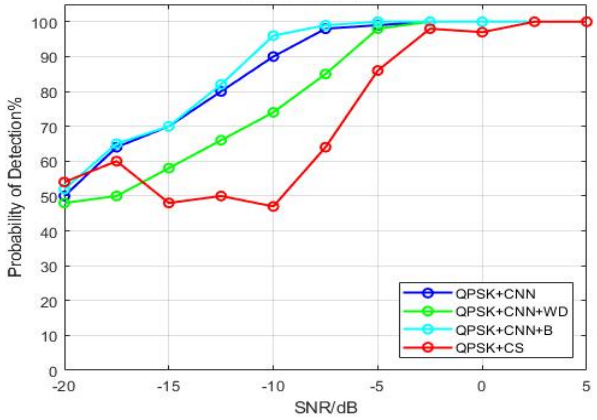

Fig. 11. Probability of detection (P_d) performance comparison between cyclisation feature and CNN, CNN+ window segment, CNN +Bernstein approximation segmentation for QPSK signal case.

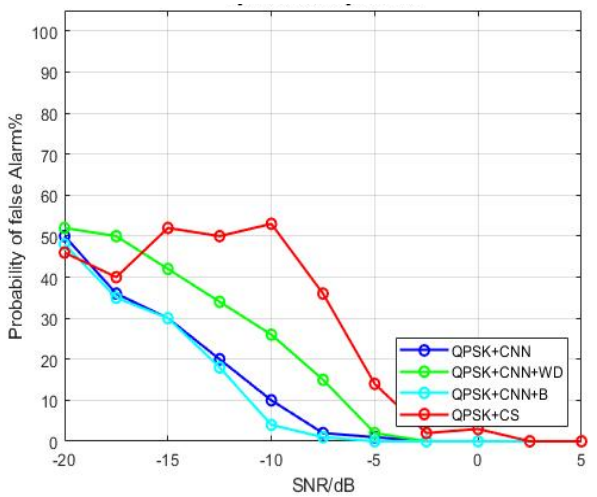

Fig. 12. Probability of False Alarm (P_f) performance comparison between cyclisation feature & CNN, CNN+ window segment, CNN +Bernstein approximation segmentation for QPSK Signal case.

False alarm probability results, presented in Figs. 10−14, show a monotonic decrease with increasing SNR for all methods. CNN+B consistently exhibits the lowest P_f, demonstrating that Bernstein polynomial approximation not only enhances P_d but also suppresses false detections by producing more discriminative spectral features. At SNR ≥ –10 dB, CNN-based methods maintain P_f below 10%, while cyclisation detection records significantly higher false alarms. Figs. 11−14 further illustrate the limitations of cyclisation detection: at –15 to –10 dB, P_d remains at only 47–50% with elevated P_f, as noise masks cyclostationary periodicities. In contrast, CNN+B nearly doubles P_d under the same conditions, highlighting the effectiveness of combining deep learning with Bernstein-based spectral smoothing.

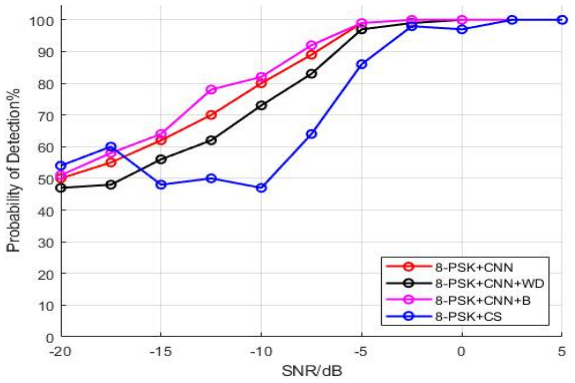

Fig. 13. Probability of detection (P_d) performance comparison between cyclostationary feature & CNN, CNN+ window segment, CNN +Bernstein approximation segmentation for 8-PSK signal case.

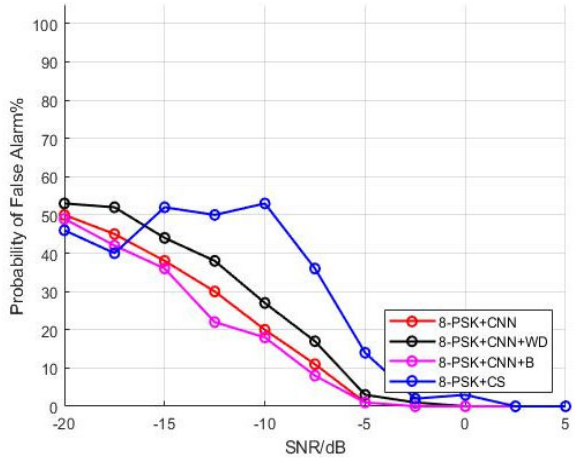

Fig. 14. Probability of False Alarm (P_f) performance comparison between cyclostationary feature & CNN, CNN+ window segment, CNN +Bernstein approximation segmentation for 8PSK Signal case.

Overall, two key performance regimes emerge. In the high-SNR regime (≥ –5 dB), all methods achieve near-perfect detection, with CNN-based models offering lower false alarm rates than cyclisation detection. In the low-SNR regime (≤ –10 dB), CNN+B significantly outperforms both CNN+WD and cyclisation methods, particularly for QPSK, due to its superior spectral approximation and feature discriminability. Furthermore, QPSK consistently demonstrates higher noise tolerance than 8PSK, owing to its wider constellation spacing. Collectively, these results establish CNN+B as a robust and reliable spectrum sensing approach under adverse low-SNR conditions, offering substantial improvements over both conventional cyclisation techniques and baseline CNN models.

VI. CONCLUSION

This work integrates deep learning methods, namely Convolutional Neural Networks (CNNs), with Bernstein polynomial approximation to propose an optimum strategy

for energy detection-based spectrum sensing. The suggested approach makes use of the approximation characteristics of Bernstein polynomials to increase CNN feature extraction, which significantly boosts spectrum sensing detection performance, particularly in low Signal-to-Noise Ratio (SNR) situations. The experimental results show that for both QPSK and 8PSK signals, CNN-based approaches perform better than classic Cyclisation Feature Detection and CNN with windowing techniques at all SNR levels. This is especially true for CNN paired with Bernstein approximation. Better feature extraction and higher detection rates are made possible by the CNN architecture's integration of the Bernstein approximation, particularly in low SNR situations. The experimental findings, supported by the statistical evidence in Table IV, demonstrate that the CNN achieves superior accuracy, higher probability of detection (P_d), and lower probability of false alarm (P_f) compared to the LinearNet This demonstrates how sophisticated deep learning methods may be used to improve cognitive radio networks' spectrum sensing capabilities. In future, investigate advanced CNN architectures, such as residual connections and attention-based mechanisms, to further improve classification performance.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

M. Subbarao was primarily responsible for drafting the main manuscript text, conducting literature research, and organizing the methodology; N. Venkateswara Rao was actively involved in the practical aspects of the research; both authors had approved the final version.

REFERENCES

[1] A. Ali and W. Hamouda, "Advances on spectrum sensing for cognitive radio networks: Theory and applications," *IEEE Commun. Surv. Tutorials*, p. 1, 2016.

[2] N. Kaabouch and W. C. Hu, *Handbook of Research on Software-Defined and Cognitive Radio Technologies for Dynamic Spectrum Management*, IGI Global, 2014.

[3] A. Ranjan, Anurag, and B. Singh, "Design and analysis of spectrum sensing in cognitive radio based on energy detection," in *Proc. International Conference on Signal and Information Processing*, 2016, pp. 1–5.

[4] M. K. Sherbin and V. Sindhu, "Cyclostationary feature detection for spectrum sensing in cognitive radio network," in *Proc. 2019 International Conference on Intelligent Computing and Control Systems*, 2019, pp. 1250−1254.

[5] S. Subramaniam, H. Reyes, and N. Kaabouch, "Spectrum occupancy measurement: An autocorrelation-based scanning technique using USRP," in *Proc. IEEE Wireless and Microwave Technology Conference*, 2015, pp. 1–5.

[6] H. Reyes, S. Subramaniam, N. Kaabouch, and W. C. Hu, "A spectrum sensing technique based on autocorrelation and Euclidean distance and its comparison with energy detection for cognitive radio networks," *Comput. Electr. Eng.*, vol. 52, pp. 319–327, May 2016.

[7] F. Salahdine, H. E. Ghazi, N. Kaabouch, and W. F. Fihri, "Matched filter detection with dynamic threshold for cognitive radio networks," in *Proc. International Conference on Wireless Networks and Mobile Communications*, 2015. pp. 1–6.

[8] X. Zhang, R. Chai, and F. Gao, "Matched filter-based spectrum sensing and power level detection for cognitive radio network," in *Proc. IEEE Global Conference on Signal and Information Processing*, 2014, pp. 1267–1270.

[9] Y. Arjoune, Z. E. Mrabet, H. E. Ghazi, and A. Tamtaoui, "Spectrum sensing: Enhanced energy detection technique based on noise measurement," in *Proc. 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018, pp. 828−834.

[10] T. Sato and M. Umehira, "A new spectrum sensing scheme using overlap FFT filter-bank for dynamic spectrum access," in *Proc. 2011 6th International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM),* 2011, pp. 6−10.

[11] Z. Tian, D. Yu, Y. Bai, S. Lei, and Y. Wang, "Application of spectrum state prediction method based on CNN-LSTM network in communication interference," *IEEE Access*, vol. 11, pp. 93538−93550, 2023.

[12] H. Monga, D. Gautam, and S. Katwal, *Wavelet Transform-Spectrum Sensing, Recent Advances in Wavelet Transforms and Their Applications*, Intech Open, Nov. 23, 2022.

[13] M. Subbarao and N. Rao, "Modified parallel FFT energy detection using machine learning based spectrum sensing," *Microsystem Technologies*, vol. 31, no. 4, 2024.

[14] P. V. Yadav, A. Alimohammad, and F. Harris, "Efficient design and implementation of energy detection-based spectrum sensing," *Circuits, Systems, and Signal Processing*, vol. 38, no. 11, pp. 5187−5211, 2019.

[15] E. V. Vijay and K. Aparna, "Deep learning-CT based spectrum sensing for cognitive radio for proficient data transmission in wireless sensor networks, e-prime," *Advances in Electrical Engineering, Electronics and Energy*, vol. 9, 100659, 2024.

[16] G. G. Kumar, K. S. Sahoo, and P. K. Meher, "50 years of FFT algorithms and applications," *Circuits, Systems, and Signal Processing*, vol. 38, pp. 5665−5698, 2019.

[17] S. Liao and Y. W. Chen, "Clustering algorithms for signal segmentation," *Journal of Signal Processing Systems*, vol. 82, no. 3, pp. 355−365, 2016.

[18] D. C. Rivera and G. Valenza, "Cluster permutation analysis for EEG series based on non-parametric wilcoxon–mann–whitney statistical tests," *Software X*, vol. 19, 2022.

[19] Y. M. Tenorio, A. P. Guerrero, and R. A. Gonzalez, "Novel multiband spectrum sensing method based on wavelets and the higuchi fractal dimension," *Sensors*, vol. 19, no. 1322, 2019.

[20] S. Han and X. Kong, "Analysis and research for tunnel monitoring data using wavelet transform modulus maxima," in *Proc. 2025 10th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 2025, pp. 247−251.

[21] B. Ziolko, "Fuzzy precision and recall measures for audio signals segmentation," *Fuzzy Sets and Systems*, vol. 279, pp. 101−111, 2015.

[22] M. M. Hassan, H. T. Mirzaei, and A. A. Ani, "ECG signal segmentation using PCA and DWT," *Biomedical Signal Processing and Control*, vol. 4, no. 4, pp. 338−346, 2009.

[23] M. H. M. Noor *et al*., "Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer," *Pervasive and Mobile Computing*, vol. 38, pp. 41−59, 2017.

[24] C. Lea, M. J. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156−165.

[25] Y. Guo *et al*., "A review of semantic segmentation using deep neural networks," *International Journal of Multimedia Information Retrieval*, vol. 7, pp. 87−93, 2018.

[26] Y. Yang, L. Li, S. Li, and Q. Wang, "Signal segmentation using convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 3039−3049, 2018.

[27] Y. Y. Li *et al*., "PointCNN: Convolution on x-transformed points," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[28] S. Kim and J. H. Choi, "Convolutional neural network for gear fault diagnosis based on signal segmentation approach," *Structural Health Monitoring*, pp. 1401−1415, 2019.

[29] A. Shahin and M. A. Alsunaidi, "A segmentation method for noisy speech signals based on multiscale product high-order statistics and deep learning," *IEEE Access*, vol. 7, pp. 15349−15363, 2019.

[30] D. Li, X. Wang, Y. Zuo, and Y. Yang, "Signal segmentation via recurrent neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2782−2793, 2019.
[31] P. Fearnhead, "Exact bayesian curve fitting and signal segmentation," *IEEE Transactions on Signal Processing*, vol. 53, no. 6, pp. 2160−2166, 2005.
[32] A. Mahmood and Z. A. Hussein, "Energy detection technique for spectrum sensing in cognitive radio: A survey," *International Journal of Computer Networks and Communications*, vol. 4, no. 5, September 2012.
[33] Y. C. Liang, K. C. Chen, G. Y. Li, and P. Mahonen, "Cognitive radio networking and communications: An overview," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 3386–3407, 2011.
[34] R. Tandra and A. Sahai, "Fundamental limits on detection in low SNR under noise uncertainty," in *Proc. 2005 International Conference on Wireless Networks, Communications and Mobile Computing*, 2005, pp. 464−469.
[35] M. A. Tuko, "Performance evaluation and comparison of different transmitter detection techniques for application in cognitive radio," *International Journal of Networks and Communications*, vol. 5, no. 5, pp. 83–96, 2015.
[36] X. Fang, M. Jin, Q. Guo, and T. Jiang, "CNN-transformer-based cooperative spectrum sensing in cognitive radio networks," *IEEE Wireless Communications Letters*, vol. 14, no. 5, pp. 15763–1580, May 2025.
[37] L. Li, W. Xie, and X. Zhou, "Cooperative spectrum sensing based on LSTM-CNN combination network in cognitive radio system," *IEEE Access*, vol. 11, pp. 876153–87625, 2023.
[38] A. Mehrabian, M. Sabbaghian, and H. Yanikomeroglu, "CNN-based detector for spectrum sensing with general noise models," *IEEE Transactions on Wireless Communications*, vol. 22, no. 2, pp. 1235–1249, Feb. 2023.
[39] S. N. Syed *et al.*, "Deep neural networks for spectrum sensing: A review," *IEEE Access*, vol. 11, pp. 89591–89615, 2023.
[40] Y. Zhang and Z. Luo, "A review of research on spectrum sensing based on deep learning," *Electronics*, vol. 12, no. 21, 4514, 2023.
[41] A. Kumar, N. Gaur, S. Chakravarty, M. H. Alsharif, P. Uthansakul, and M. Uthansakul, "Analysis of spectrum sensing using deep learning algorithms: CNNs and RNNs," *Ain Shams Engineering Journal*, vol. 15, no. 3, 2024.