Comparative Evaluation of STFT–Random Forest and Fuzzy STFT–SVM Frameworks for Robust Spectrum Sensing Using QPSK I/Q Data

Raman R*, Deepa N Reddy
Department of Electronics and Communication Engineering, BMS Institute of Technology and Management, Bengaluru, Karnataka, India
E:-mail: ramanrajesh1111111@gmail.com
*Corresponding author

Keywords: spectrum sensing, machine learning, STFT, GNU radio, fuzzy systems, QPSK, Signal to noise ratio, Support Vector Machine.

Received: October 2, 2025

The current work exhibits an overview involving the comparison of two different machine learning–based spectrum sensing pipelines which make use of Quadrature Phase-Shift Keying (QPSK) in-phase/quadrature (I/Q) data generated in GNU Radio. The two pipelines share as common the Short-Time Fourier Transform (STFT)–based spectral features along with pseudo-labeling taken from energy detection. Direct handling of raw STFT features by a Random Forest (RF) model is the first pipeline. The second pipeline on the contrary, integrates a fuzzy feature engineering phase where STFT features are altered with neuro-fuzzy processing before being passed to classification with a Support Vector Machine (SVM) technique that is referred to as the Fuzzy STFT–SVM (FuST-SVM) framework. The different methodical tests are carried out when the signal-to-noise ratio (SNR) is low (−10 dB), medium (5 dB), and high (10 dB). The outcome shows that the FuST-SVM pipeline is the one that always has the superiority over the RF-based method that even reaches the highest 92.46% in accuracy measurement through the tested SNR levels from 90.65% to 92.46%. The studies support that the utilization of fuzzy spectral representations in spectrum sensing improves the noise and uncertainty handling in the proposed FuST-SVM framework that it becomes an evenly efficient and dependable solution for wireless environments that are challenging.

Povzetek: Članek primerja dva pristopa strojnega učenja za zaznavanje spektra ter pokaže, da metoda FuST-SVM z uporabo mehkih spektralnih značilk dosega višjo točnost in boljšo odpornost na šum kot pristop z naključnim gozdom.

1 Introduction

With an increasing demand for wireless communication services, the radio frequency (RF) spectrum conventionally allocated has become severely congested. To fight such a shortage, CR technology has been proposed as the new paradigm with an emphasis on dynamic spectrum access as opposed to fixed assignments. Spectrum sensing is a core function behind all CR activities, wherein secondary users detect the presence or absence of primary users (PUs) in a certain frequency band, reliably without causing harmful interference [1]. Such spectrum sensing mechanisms are vital to enhance the utilization of the spectrum, ensure interference-free communication, and promote efficient communication in dynamically operated wireless environments. The merging of electronics and AI in the information society motivates this research, with intelligent spectrum sensing being the major contributor to the adoption of wireless communication systems that are both scalable and efficient [2].

1.1 Background

Since traditional spectrum sensing methods have their inherent drawbacks, let us briefly discuss some of them. Energy detection(ED) is easy to implement but has the drawback of the "SNR wall," meaning it performs poorly in low SNR regimes where primary user signals are deeply buried in noise. Matched filtering requires prior knowledge of the primary user's signal; however, such information is usually not available in practical situations. Cyclostationary feature detection is stronger during low SNR but demands the sacrifice of computational power and observation time. Furthermore, wireless channels are especially dynamic, subject to fading, shadowing, and changing interference levels, which makes it really difficult to achieve robust sensing over a wide SNR range [3]. Thus, with the demand for more spectrum sensing, Machine Learning has emerged recently as a powerful tool thereof. These ML algorithms can learn highly intricate non-linear patterns in observed data and adapt to dynamic

environments, thereby offering a chance of overcoming the limitations of the conventional methods. Using various feature extraction methods and intelligent classifiers, ML methods may be capable of detecting spectrum holes with high accuracy, especially in noisy and uncertain environments [5,6].

1.2 Motivation

The paper puts together a comparative study of two different machine learning pipelines for spectrum sensing. The study works with QPSK I/Q data generated with GNU Radio, while energy detection is used as the pseudo-labeling for supervised learning. Both these pipelines use 8-band STFT features from this data. One pipeline consists of an RF classifier that is directly applied to the STFT features [8-10]. The other pipeline consists of the machine that first learns fuzzy feature engineering over STFT features and subsequently uses them with an SVM for classification [7,11]. By testing under various SNR conditions (low at -10dB, medium at 5dB, and high at 10dB), the performance characteristics indicate each pipeline in view of promising solutions to robust spectrum sensing.

In the realm of ML-aided spectrum sensing, fuzzy logic ensures a rather peculiar advantage. Wireless channel conditions, signal characteristics, and noise levels are all worlds away from the pragmatic certainty that mathematical models usually assume. Fuzzy logic provides one approach to modeling and processing vagueness to permit "soft" decisions that rely on degrees of partial truth as opposed to starkly considered binary logic. By converting crisp input features into degrees of membership of fuzzy sets, the fuzzy-based ML model may be better able to model subtler relationships and arrive at more robust classifications; this is particularly important where the conventional techniques find difficulties posed by the overlaps of signal and noise distributions. The neuro-fuzzy approach, where the fuzzy logic affects the input or structure of the machine learning models, has indeed been fruitful in certain signal processing applications.

1.3 Contributions

The major contributions in this work are summarized below:

- A pragmatic comparison of feature–classifier pipeline combinations for spectrum sensing using realistically generated QPSK I/Q data in GNU Radio
- Introduced the use of 8-band STFT features combined with pseudo-labeling via energy detection as a means to produce efficacious training data.
- Introduced a neuro-fuzzy processing stage on STFT features before classification in an SVM framework (Fuzzy STFT–SVM), highlighting its robustness.
- Detailed analysis of the relative trade-offs in performances, aiming at providing a benchmark study to support future development of intelligent noise-resilient spectrum-sensing models in cognitive radio.

1.4 Organisation

Following are the details of the remaining chapters of the paper: Section 2 is concerned with a review of the literature on ML-based spectrum sensing. Section 3 discusses the proposed method, from data preprocessing to feature extraction methods, fuzzy feature engineering, and ML classifiers. Section 4 presents Implementation. Section 5 presents Results and Discussion. Finally, Section 6 concludes the Paper and points out directions for future work.

2 Related work

The subject of spectrum sensing was one which attracted varied and ample research interest for the improvement of detection accuracy and efficiency. Traditional techniques based on old signal processing paradigms were initially employed: energy detection, matched filter, cyclisation feature detection, and so forth. Energy-detection is simple; however, it reflects very poor performances at low SNR situations, and this is well known as the SNR wall problem [3]. In other words, the matched-filter-based detection necessitates the knowledge of the primary user signal itself, which is more likely not available in dynamic environments; cyclostationary detection provides robustness but requires extensive computations. Thus these drawbacks call for more intelligent sensing paradigms that adapt to given scenarios.

The beginnings of machine learning opened a promising avenue to overcome inherent challenges in traditional spectrum sensing. There have been various studies that have explored the techniques of ML for the detection of occupied and idle spectrum bands [5]. As early as the beginning, classifiers such as SVMs and k-NNs were proven feasible to be used with carefully selected features, which gave even better detection performance [6,9]. From its very nature, ML leaves space for models capable of learning complex patterns from all kinds of wireless environments, thereby bringing their decision-making power upwards.

Feature extraction is one of the essential parts of ML-based spectrum sensing, as the performance of the classifier is directly dependent on feature quality. The researchers have discussed a multitude of feature types with time, frequency, and time-frequency domain references. Moments of statistics, higher-order cumulants, wavelet coefficients, and STFT-derived features have thus been employed in different studies [8,9,14,18]. Wavelet transform-based features, for instance, are used because of their multi-resolution analysis capability that captures transient characteristics of signals. On the other hand,

higher order cumulants differentiate signals while being immune to Gaussian noise [8], [18]. STFT-based features will then provide vast spectral information important to differentiating various signal types [4].

Various machine learning algorithms have been used for spectrum sensing. SVMs are generally practiced because of their generalizing capabilities and being effective in high-dimensional spaces [6], [19]. Random forests combine different decision trees and have proved themselves satisfactorily in terms of better accuracy and avoiding over-fitting [10]. Other algorithms like Naive Bayes (NB), Decision Tree, and varieties of Artificial Neural Networks (ANNs) have been tried, each with its own set of advantages and limitations depending on the dataset and features [12,20,21].

Combining fuzzy logic with machine learning methods, the so-called neuro-fuzzy systems, is an important recent development in addressing the uncertainties of wireless signals. Fuzzy logic allows modeling vagueness and imprecisions so that a decision-making process can be carried out in a more subtle manner as compared to having a more rigid form of binary logic on either side. The literature mentioned the use of fuzzy logic in adaptive thresholding for energy detection and created another set of more robust feature representations for an ML classifier [7,17]. Neuro-fuzzy systems, which combine the learning capability of neural networks with the interpretability of fuzzy systems, were also proposed for spectrum sensing, thus improving reliability, especially in environments with complex and ambiguous signals [11]. The very recent Deep Learning (DL) has emerged as a latest paradigm in spectrum sensing, whereby neural networks of advanced processing like CNNs and RNNs are employed to learn underlying features from raw RF data or spectrograms. These approaches achieve, most of the time, state-of-the-art performance, but the methods require vast training data and a lot of computational power 12]. Existing spectrum sensing approaches suffer from fixed thresholds, noise-sensitive features, and poor robustness under low SNR; while deep models make great computational costs. Traditional ML pipelines such as RF and SVM do not have adaptive preprocessing, whereas pure fuzzy logic methods cannot capture deeper discriminative structures. Table 1 summarizes the related work.

Contemporary fuzzy–ML approaches improve adaptability but often suffer from high computational complexity, manual tuning of fuzzy rules/membership functions, and poor scalability under very low SNR conditions. Moreover, they have a hard time achieving simultaneous high interpretability and maximum classification accuracy. The new FuST-SVM model efficiently removes these barriers by employing fuzzy logic only for determining the adaptive threshold, while the SVM retains robust classification, thus being able to provide better low-SNR performance with decreased complexity and increased reliability.

Unlike deep learning models, the proposed FuST-SVM does not require large labeled datasets or high-end GPUs. It provides quicker training, reduced computational expenses, and improved interpretability, making it appropriate for real-time and resource-constrained spectrum sensing.

FuST-SVM, therefore, fills this gap through combining fuzzy reasoning with SVM crisp classification on STFT features 90–92% at –10 dB SNR, thereby ensuring a lightweight, noise-robust and practically deployable system for reliable spectrum sensing in cognitive radio environments.

Table 1: Summary of related works

Reference	**Methodology**	**Advantages**	**Limitations**	**Reported Performance Metrics on detection accuracy**
Atapattu et al. [3]	Classical energy detection	low computational complexity, no prior knowledge of PU required	Highly sensitive to noise uncertainty, poor performance at low SNR	High Pd at high SNR; detection degrades significantly below –10 dB
Jan & Koo et al. [6]	Feature-based spectrum sensing using SVM	Higher detection accuracy than energy detection; robust to noise variations	Requires feature extraction and training data	Achieved higher Pd (≈90–95%) compared to energy detection in low SNR.
Dibal et al. [8]	wavelet-based feature extraction to detect spectrum edges	Good localization in time and frequency; suitable for wideband sensing	Computationally complex, threshold selection difficult	better detection over energy detection, especially in noisy channels
Dey et al. [11]	Combines neural networks + fuzzy logic with double threshold energy detector	Reduces false alarm and missed detection; adaptive to noise uncertainty	Increased system complexity and training overhead	Higher Pd and lower Pf than conventional ED (≈10–15% Pd improvement)
Wu et al. [15]	ML based cooperative sensing	Improves detection reliability while reducing sensing energy	Needs multiple SUs and coordination overhead	Achieved Pd above 95% with reduced sensing energy consumption

3 Proposed methodology

3.1 Signal generation and dataset preparation

Spectrum sensing by the SU is the detection of the PU activity in a noisy environment. The detection of the licensed user at the SU is modeled as a binary hypothesis test problem, given as

$$\left\{\begin{matrix} H_0: x[n] = v[n] \\ H_1: x[n] = hs[n] + v[n] \end{matrix}\right\} \quad (1)$$

Where, *x [n]* is the received signal by the SU, *h* is the amplitude gain of the channel, *s[n]* is the signal transmitted by PU, *v[n]* is the additive noise at the SU.

The energy measurement *Y* is calculated from *N* samples of the received signal at the CR receiver and compared against λ ,which is the detection threshold to decide on the presence of PU(H_1)or absence of PU (H_0).

$$Y = \sum_{n=1}^{N} |x[n]|^2 \underset{H_0}{\overset{H_1}{\gtrless}} \lambda \quad (2)$$

The transmitted signal by PU is considered to be QPSK modulated and the signal datasets were generated using GNU Radio whereby a dedicated signal flowgraph was designed and configured to simulate the modulation schemes of QPSK. Modulation was realized by constructing circuit-like blocks within GNU Radio Companion (GRC) to provide a fine tuning of signal parameters and waveform generation.

The QPSK signal is mathematically given as:

$$s[n] = I[n]\cos(2\pi f_c n T_s) - Q[n]\sin(2\pi f_c n T_s) \quad (3)$$

where I[n], Q[n]ϵ{-1,+1}, in phase and quadrature components based on 2 bit groups.

A channel model block has been included in each of the GNU Radio flowgraphs to simulate realistic wireless communication conditions. An Additive white Gaussian noise(AWGN) block was not employed for this, but the noise voltage parameter was varied within the channel model to simulate noise levels corresponding to certain SNR values of -10 dB, 5 dB, and 10 dB. These SNR levels were representative of the environments with high, medium, and low interferences, respectively.

This method is an effective emulation of channel impairments of the real world systems, which can be further considered as a better and more pragmatic way to evaluate the performance of the modulation classification model under different noise conditions.

The signals generated form the input dataset which can be further processed to facilitate a controlled, systematic evaluation of the performance of modulation classification under varying noise conditions.

3.2 Preprocessing of signals

To prepare the continuous I/Q data for machine learning, the following preprocessing steps were performed:

- **Segmentation:** Continuous I/Q data streams have been separated into overlapping segments. Individual segments had a fixed Segment Length of 1024 samples with an Overlap Ratio of 50%, generating a step size of 512 samples. The segmentation allows the analysis of signal local characteristics over time [4].
- **Pseudo-Labeling via Energy:** In real-life wireless environments, ground-truth labels truly never exist; therefore, some method must be used to fabricate training labels for supervised learning. The pseudo-labels for each segment were, in this work, generated by energy detection-a very basic method of spectrum sensing [4]. The energy of the -th segment, composed of complex samples as calculated as the sum of the squared magnitudes of its component samples:

$$E_i = \sum_{n=0}^{N-1} |x_i[n]|^2 \quad (4)$$

A global pseudo-threshold (λth) was then set as the mean energy across all M segments in the dataset:

$$\lambda th = \frac{1}{M}\sum_{i=1}^{M} E_i \quad (5)$$

Any segment with energy above this threshold was labeled as 'occupied' (1), and any segment with energy below became labeled as 'free' (0). In this manner, we provide a consistent, if simplified, ideal ground truth for training the classifiers, a common approach in machine learning-based spectrum sensing when explicit ground truth is absent [5].

- Train-Test Split: The entire segment dataset with corresponding pseudo-labels was split into training and testing subsets with 70 and 30% proportions, respectively. A stratified sampling approach was used with random state=42. Now, irrespective of package library used, 'occupied' and 'free' segments must be in the training set with the remaining in the test set, a fair test of the models' ability to generalize.

3.3 Feature extraction technique

Good-quality feature extraction is critical for machine-learning performances. This paper deals with Short-Time Fourier Transform (STFT) based features as they retain spectral information.

- Short-Time Fourier Transform (STFT) Features: The segmented signal was then subjected to an STFT. The STFT for N Per Segment was set to 256, meaning the FFT

window was 256 samples long, and STFT for N Overlap was set to 128, representing a half-window overlapping configuration. The power spectrum for each segment was calculated by taking the mean of the squared magnitudes of the complex STFT coefficients across the time bins. Initially, this provides 256 clear-cut power values for each frequency bin

$$STFTx[n](m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n - mR]e^{-i\omega n} \quad (6)$$

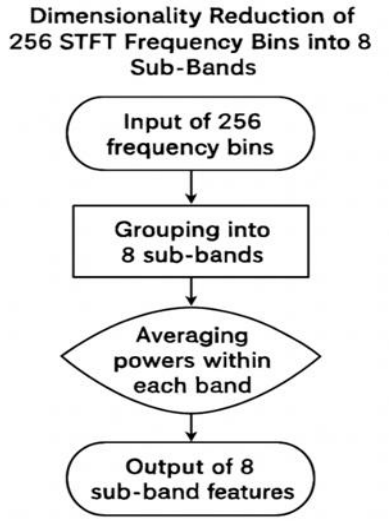

Figure 1: STFT 8-Band reduction

- Dimensionality Reduction (8-Band Features): The 256 crisp STFT features are reduced to 8 features to combat the curse of dimensionality, improve computational efficiency, and build more generalized spectral descriptions. The 256 frequency bins are divided into 8 equally sized sub-bands (each 32 bins wide), and the average power in each sub-band is calculated. These 8 average power values constitute the compact and abstracted crisp STFT features that are used in both classification pipelines. This process is outlined in Figure 1.

3.4 Fuzzy feature engineering

A fuzzy membership function maps a crisp input from the universe of discourse X to a membership value, representing the degree to which the input belongs to a fuzzy set defined as μ_A: X → [0, 1]. Graphically, the universe of discourse is shown on the X-axis and the membership degree on the Y-axis. Among various types, the triangular membership function (TMF) is widely used due to its simplicity and computational efficiency, and is defined by three parameters a, b and c, where a and c determine the base and b denotes the peak (maximum membership) of the triangle [7,17].

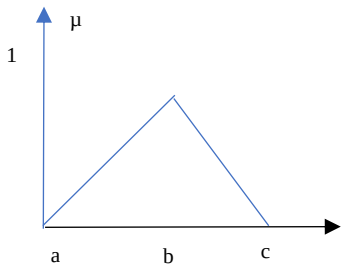

Figure 2: Traingular membership Function

The fuzzy triangular membership function is expressed as

$$\mu(x;a,b,c) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a \le x \le b \\ \frac{c-x}{c-b}, & b \le x \le c \\ 0 & c \le x \end{cases} \quad (7)$$

The eight crisp STFT features were directly fed into the STFT-Random Forest pipeline. For the STFT-Fuzzy-SVM pipeline, these same crisp features were subjected to fuzzification, where fuzzy logic concepts were brought into play to allow for potential uncertainties in the feature values and to offer a richer input to the classifier.

- **Concept of fuzzification**: Fuzzification converts a precise (crisp) numerical input into a set of fuzzy membership degrees. Instead of a feature value being strictly "high" or "low," it can be "partially high" and "partially medium" simultaneously.
- **Fuzzy set definition:** For each of the 8 crisp STFT features, three triangular fuzzy membership functions (FMFs) were defined across the feature's observed range: 'low', 'medium', and 'high'. The universe of discourse (range) for each feature's FMFs was determined from its minimum and maximum values across the entire dataset.
- **Membership degree calculation:** For every crisp feature value from each segment, its degree of membership (between 0 and 1) was calculated for each of the three fuzzy sets. This effectively transforms each single crisp feature into three fuzzy features (the membership degrees). Consequently, the 8 crisp STFT features were expanded into 8 * 3 = 24 fuzzified features. This process was implemented using the scikit-fuzzy Python library.

3.5 Machine learning classifiers

Two distinct machine learning classifiers were employed and compared:

- **Random Forest (RF):** The first of the pipelines, STFT with Random Forest, sundered together Random Forests, which comprise an ensemble learning method that, in the training phase, builds a great number of decision trees and outputs the class which is the mode of the classes (classification) of the individual trees. It is robust, can handle high-dimensional data, and is immune to overfitting. Based on this study, an RF classifier was chosen with n estimators or the number of trees set to 100.

$$\hat{y}[n] = mode\big(h_1(x[n]),\ h_2(x[n]),\ \ldots,\ h_T(x[n])\big) \quad (8)$$

Total number of trees = 100

Support Vector Machine (SVM): The second pipeline (STFT with Fuzzy and SVM) includes the Support Vector Machine, which is known to be an excellent supervised learning classification tool[6]. It creates an optimal separating hyperplane in the high dimensional space to segregate different classes. Because of its nonlinear nature, the radial basis function RBF kernel was chosen, with the parameter C (regularization) set to 10 and gamma (kernel coefficient) set to scale. The hyperparameters of the Random Forest and Support Vector Machine classifiers were selected through empirical tuning using cross-validation. For the Random Forest classifier, the number of trees was set to n_estimators=100, as higher values did not yield significant performance improvements while increasing computational cost. For the SVM classifier, a radial basis function (RBF) kernel was employed, with the regularization parameter C=10 and kernel coefficient set to "scale". These values were chosen based on 5-fold cross-validation on the training dataset to achieve a balance between classification accuracy and generalization performance.

$$f[n] = w^T x[n] + b\hat{y}[n] = \begin{cases} +1 & if\ f[n] \geq 0 \\ -1 & if\ f[n] < 0 \end{cases} \quad (9)$$

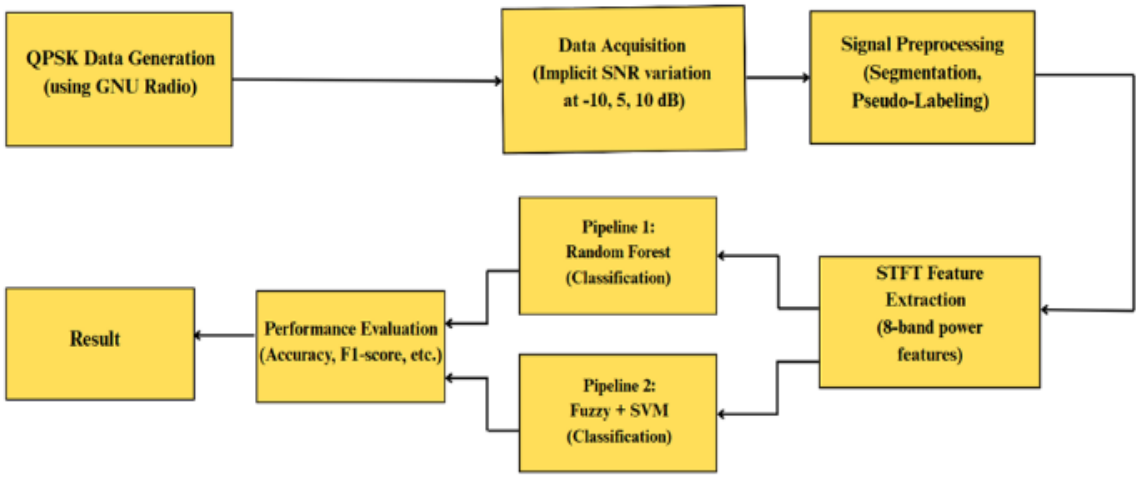

Figure 3: Process flow diagram

4 Results, discussion and performance evaluation

The flow of the project is detailed in Figure 3 and subsequently detailed in the following sections.

4.1 Signal generation using GNU radio

The very first step of this project involved the development and realization of modulation flowgraphs in GRC. The user interface of GNU Radio has great functionality that provides a means of constructing signal processing pipelines by interlinking various kinds of pre-built blocks. For this project, the separate flowgraphs were developed for QPSK modulation scheme. The QPSK signals were generated using GNU Radio with a sampling rate of 1 MSamples/s. The continuous I/Q data stream was segmented into frames of 1024 complex samples with a 50% overlap, resulting in a step size of 512 samples. For each SNR level (−10 dB, 5 dB, and 10 dB), approximately 1500 segments were generated, of which 70% were used for training and 30% for testing. Pulse shaping was implemented using a root-raised cosine (RRC) filter with a roll-off factor of 0.35 to emulate realistic communication conditions.

Every such flowgraph simulates the transmission of modulated signals in which required components such as signal sources, modulation blocks, channel noise blocks, and file sinks interoperate in a realistic signal transmission chain, modulating the signals and adding variable amounts of noise to simulate varying SNR conditions (-10 dB, 5 dB and 10 dB). The File Sink block was used to store the output signals, which were extracted and processed into a dataset for the classification task. These flowgraphs were the mainstay

of signal generation, providing modulated waveforms for further feature extraction and classification through machine learning models.

4.2 Dataset generation and classifier implementation

The signals modulated from GNU Radio were used to generate a dataset for this project. The modulation scheme QPSK has a dedicated flowgraph with Noise Source block, setting the noise voltage parameter adjusted to simulate three different SNR levels, which brought about the different low to moderate signal quality environments. The flow graph for signal generation using QPSK is given in Figure 4.

The nodes in File Sink blocks of each flowgraph captured the modulated output signal with noise. Output is saved in .dat files. These .dat files are the basis of the dataset created and were subsequently subjected to down sampling so as to reduce redundancy and improve efficiency during processing.

4.3 Classification

The complete sensing framework would be composed of data preprocessing, feature extraction, fuzzy feature engineering, training of a machine learning model, and performance evaluation in Python 3.x. All development and execution environments utilized standard libraries of scientific computing, lending reproducibility and efficiency.

Software environment: The key libraries used here were NumPy, which performs the uppermost high-level numerical operations, Pandas that arrives at data loading and manipulation of CSV files, SciPy, which offers signal processing functionalities like the Short-Time Fourier Transform (STFT).

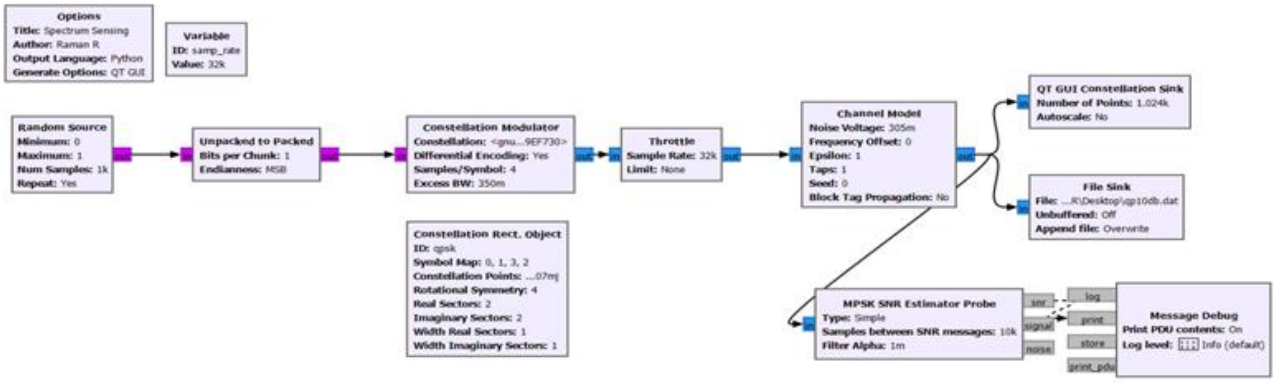

Figure 4: GNU Radio flow graph for QPSK signal generation

- **Machine learning framework:** Scikit-learn library covers generally all aspects of machine learning, including splitting the data (train_test_split), instantiating classifiers such as Random Forest Classifier, SVC), training the models (.fit()), predicting (.predict() and .predict_proba()), and calculating performance metrics (accuracy score, classification report, confusion matrix, Receiver Operating Characteristic (ROC) curve, Area under the curve(AUC)).

- After STFT computation, the spectral features were organized into a structured dataset in CSV format. Each row of the dataset corresponds to one signal segment, while each column represents an extracted feature. For the STFT-based pipeline, each sample is represented by an 8-dimensional feature vector corresponding to the average energy of eight frequency sub-bands. In the FuST-SVM pipeline, each crisp STFT feature is transformed into three fuzzy membership values, resulting in a 24-dimensional feature vector per sample. An additional column is used to store the pseudo-label indicating spectrum occupancy.

- **Pipeline execution flow:** For ensuring the correctness of comparison among the two different spectrum sensing pipelines, the following structure was created and executed in Python:

  1. Common Preprocessing: The QPSK I/Q data was loaded, segmented, and pseudo-labelled using energy detection. The segment indices underwent a consistent stratified train-test split (random_state=42).

  2. Common Feature Extraction: The set of 8-band features was extracted once for every segment.

  3. Pipeline 1 Execution: These 8-band features of STFT would give into Random Forest Classifier directly for its learning and

testing.

4. Pipeline 2 Execution: Initially, the 8-band STFT features underwent fuzzification by the fuzzy feature engineering module (8 crisp features were converted into 24 fuzzified ones); afterward, these fuzzified features were used for training and testing in the SVC classifier.

- **Reproducibility:** In order to compare different methods fairly, a fixed random_state was maintained for all randomized parts (train_test_split, Random Forest Classifier, and SVC initialization), thus producing consistent and reproducible experimental results throughout several runs. Performance was plotted using Matplotlib and Seaborn.

4.4 Tools and environment

The entire development and implementation of the signal type modulation classification system was carried out using the below utilities and software platforms.

A. Python environment

Python is used mainly for signal as well as data processing, feature extraction, implementation and evaluation of machine learning models. It has been used largely with its flexibility and large range of scientific libraries expected to make it an appropriate choice for this project. The version 3.10 was used in this work.

B. GNU Radio

The major area of symptom and modulation simulation of QPSK had to design flowgraph and modular block called signal source, modulator, noise source, and sink to make modularity possible-for the resulting signal at different SNR levels. GNU Radio Version: 3.10 was used in this study.

C. Operating system

Windows 10 served as the main platform that generated signals, prepared datasets, and trained models. It was a reliable platform for both GNU Radio and Python-based toolchains.

All experiments were conducted on a system equipped with an Intel Core i7 CPU, 16 GB RAM, and running Windows 10 operating system. No GPU acceleration was used, as all models were implemented using classical machine learning techniques. Signal processing and machine learning workflows were developed in Python 3.10 using standard scientific libraries, including NumPy, SciPy, scikit-learn, and scikit-fuzzy. GNU Radio version 3.10 was used for signal generation.

A short pseudocode used in the pipeline is detailed below:

---

1. Generate QPSK I/Q data using GNU Radio at specified SNR

2. Segment I/Q data into fixed-length overlapping frames

3. Compute energy of each segment and assign pseudo-labels

4. Apply STFT to each segment

5. Reduce STFT spectrum into 8 frequency sub-band features

6. (FuST-SVM only) Apply fuzzy membership functions to STFT features

7. Split dataset into training and testing sets

8. Train Random Forest or SVM classifier

9. Evaluate performance using accuracy, AUC, and confusion matrix

10. Spectrum occupancy decision (occupied / free)

---

5 Results and performance analysis

The classification results of the proposed model were examined at three different SNR levels, namely: -10 dB, 5 dB and 10 dB. The table below summarizes the performances of three ML classifiers, Random Forest and SVM, trained on features extracted solely using the Short-Time Fourier Transform (STFT) method and Fuzzified STFT.

5.1 Performance analysis

A structured evaluation makes a good comparison of classifier-feature combinations and indeed gives a practical point of view for choosing the most proper configuration for given use cases. In order to evaluate this study the following parameters are considered in the performance evaluation. The performance analysis of this study was carried out considering the following parameters

A. **Noise Robustness:** It checks how much accuracy each classifier retains over different SNR levels. A robust model shows a meager fall in performance, even in low SNR (-10 dB).

B. **Computational Viability**: The time and

resources required for feature extraction, training and testing are considered. This becomes vital when deployment of such systems comes into real-time environments as software-defined radios or embedded systems.

C. **Metrics:** The metrics used in performance evaluation is detailed in Table 2.

The Area Under the ROC Curve (AUC) gives additional support to the fact that the FuST-SVM pipeline has a better discriminative capability, especially at low-SNR conditions. The AUC values given in Table 3. The values that are higher all the time show that the spectrum states of occupied and idle are separated better even when the differences in classification accuracy are small in terms of numbers.

Table 3: AUC values across various SNR conditions

SNR (dB)	**STFT+Random Forest**	**STFT+Fuzzy +SVM**
-10	0.95	0.98
5	0.97	0.99
10	0.98	0.98

Table 2: Metrics used for performance evaluation

Metric	**Equation**	**Interpretation in the context of Signal Detection**
Precision	$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$	How often the system is correct when it says the spectrum is occupied.
Recall	$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$	How well the system detects a primary user when it is actually present.
F1-Score	$\mathbf{F1\ score} = \frac{\mathbf{2TP}}{\mathbf{2TP} + \mathbf{FP} + \mathbf{FN}}$	Overall detection quality considering both misses and false alarms.
Accuracy	$\mathbf{Accuracy} = \frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP}}$	How often the system makes the correct decision overall.

Where FN defines False Negative, TP refers True Positive, FP denotes False Positive, and TN refers True Negative.

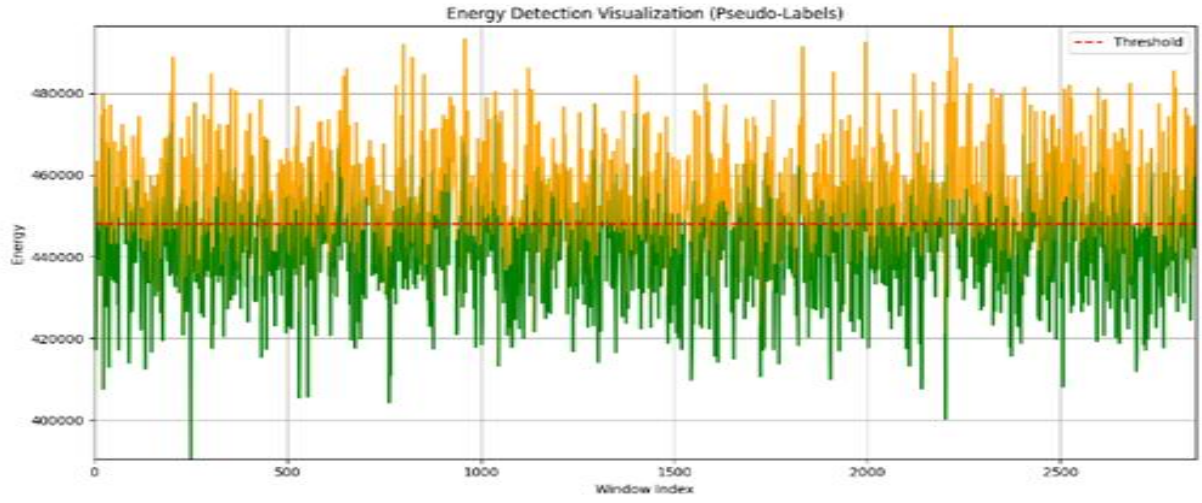

Figure 5(a): Energy distribution and pseudo-labelling threshold for spectrum sensing for SNR of –10dB

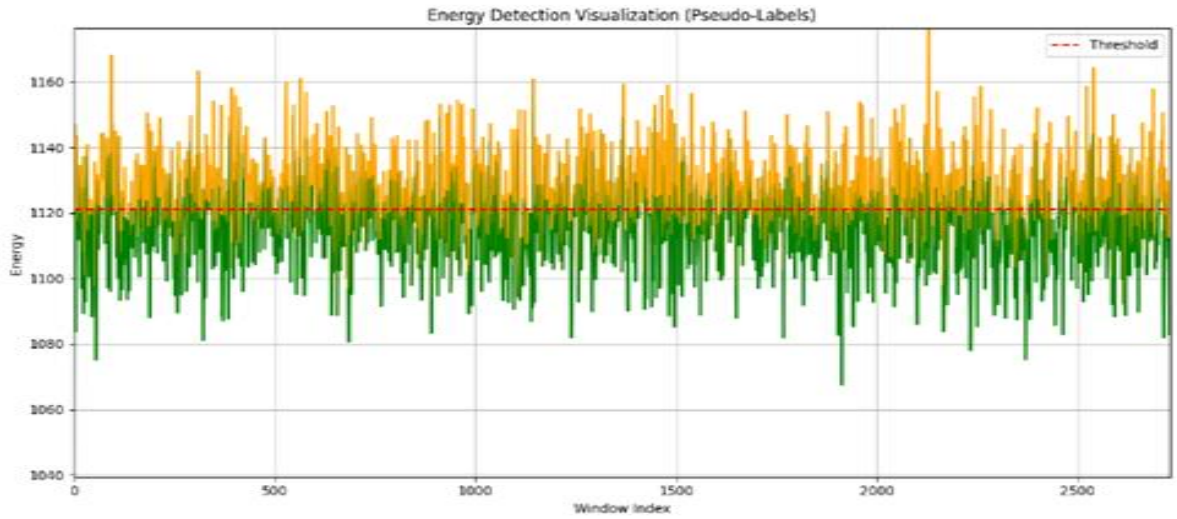

Figure 5(b): Energy distribution and pseudo-labelling threshold for spectrum sensing for SNR of 10dB

The energy distribution and Pseudo labelling threshold for spectrum sensing is shown in Figure 5 (a) and (b). The confusion matrix and ROC curve for STFT with Random Forest and STFT with Fuzzy and SVM pipelines for SNR of -10dB and 1-dB are given in Figure 6 and 7 respectively.

The accuracy at -10dB SNR with such added noise is 87.97% for STFT-Random Forest. According to the confusion matrix (Figure 6a), the system correctly classified 379 examples in the free category and 374 in the occupied category while it counted 55 false positives and 48 false negatives. On the other hand, in the ROC curve (Figure 7a), the pipeline registered an AUC of 0.95, indicating extraordinary discrimination.

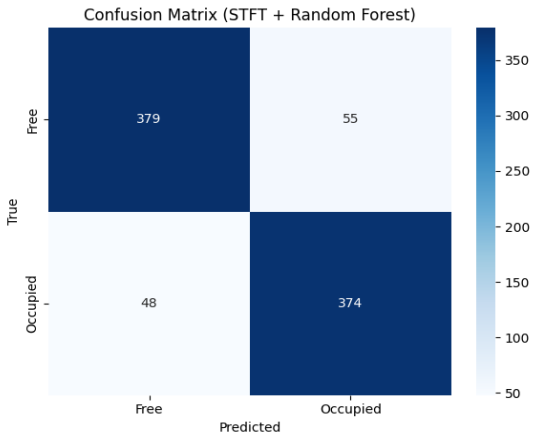

Figure 6(a): Confusion Matrix for STFT + Random Forest at -10dB SNR.

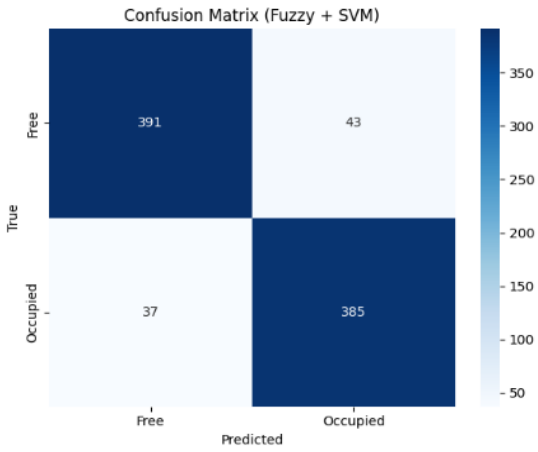

Figure 6(b): Confusion Matrix for STFT + Fuzzy + SVM at -10dB SNR.

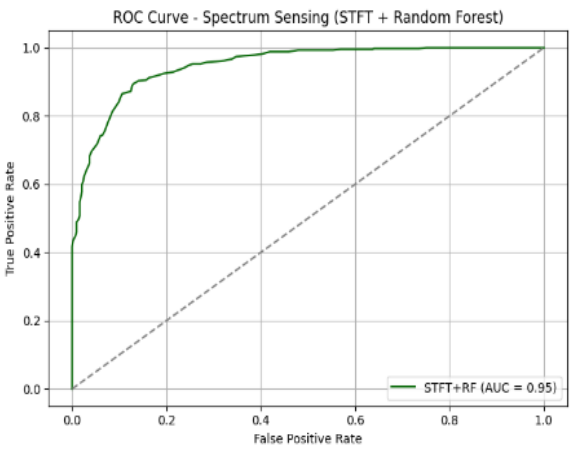

Figure 7(a): ROC Curve for STFT + Random Forest at -10dB SNR.

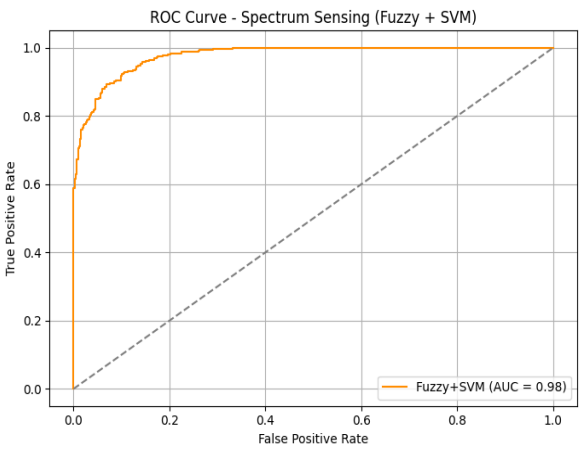

Figure 7(b): ROC Curve for STFT + Fuzzy + SVM at -10dB SNR.

Then, however, at the same noise level of -10dB SNR, STFT-Fuzzy-SVM beats all other methods with an accuracy of 90.65%. The confusion matrix (Figure 6b) revealed that there were 391 'free' and 385 'occupied' correct classifications. It has fewer errors: 43 false positives and 37 false negatives, against these of STFT-Random Forest. The ROC curve of STFT-Fuzzy-SVM (Figure 7b) with an AUC of 0.98 confirms this increased discriminating ability between spectrum states, even under very high noise conditions.

The extreme importance of spectrum sensing at varying SNRs is illustrated in Table 4. Both models kept high accuracy, well above 87%, and high F1-scores above 0.88 for all SNRs tested. Never did this pipeline with Fuzzy and SVM drop below that of Random Forest in the entire range. At -10dB of SNR, it reached 90.65% accuracy (F1: 0.91) against the 87.97% (F1: 0.88) of STFT with Random Forest; peak performance arrived at 5dB SNR, at 92.46% (F1: 0.92), which was well above the 91.24% (F1: 0.91) of STFT with Random Forest, and then remained marginally above with 91.22% (F1: 0.91) at 10dB SNR in comparison to 90.85% (F1: 0.91). This sustained edge, paired with balanced precision and recall scores through every SNR, kicked in the enhanced robustness and discernibility of the fuzzified spectral characteristics with Support Vector Machines for trustworthy spectrum sensing.

Table 4: Accuracy, Precision, Recall, and F1-Score Comparison of Naive Bayes, Random Forest, and SVM

SNR	Pipeline	Accuracy (%)	F1-score (weighted avg)	Precision	Recall
-10dB	STFT + Random Forest	87.97	0.88	0.88	0.88
	STFT + Fuzzy + SVM	90.65	0.91	0.91	0.91
5dB	STFT + Random Forest	91.24	0.91	0.91	0.91
	STFT + Fuzzy + SVM	92.46	0.92	0.92	0.92
10dB	STFT + Random Forest	90.85	0.91	0.91	0.91
	STFT + Fuzzy + SVM	91.22	0.91	0.91	0.91

The summary of the time, and Memory Comparison of Classifiers at Varying SNRs using the models used is given in Table 5.

At -10dB, FuST-SVM required about 6.32 seconds, while the STFT with Random Forest processing lasted only 2.64 seconds. The additional time arises in FuST-SVM due to extra computation involved in fuzzy feature engineering steps (conversion of 8 crisp features into 24 fuzzified features) and somewhat greater training complexity of Support Vector Machines with the RBF kernel compared to Random Forests. Memory-wise, both pipelines typically consumed very little.

While the STFT with Fuzzy and SVM recorded higher memory usages at 5dB and 10dB (0.50 MB) compared to the STFT with Random Forest (0.17 MB at -10dB and 10dB), the latter instead saw a strange spike of 4.76 MB at 5dB. Overall, though, the estimates for memory consumption remain respectable for the two approaches. Thus, this analysis shows that the better classification accuracy offered by the STFT with Fuzzy and SVM pipeline is accompanied by moderately increased computational time, a fairly profitable trade-off in most cases for improved.

Table 5 : Time and memory comparison of STFT-based classification pipelines at varying SNRs

SNR	Classifier	Time (s)	Memory (MB)
-10 dB	STFT + Random Forest	2.6394	0.17
	STFT + Fuzzy + SVM	6.3213	0.17
5 dB	STFT + Random Forest	2.8569	4.76
	STFT + Fuzzy + SVM	6.0999	0.50
10 dB	STFT + Random Forest	3.5302	0.17
	STFT + Fuzzy + SVM	6.8905	0.50

Table 6: Comparative analysis of the proposed method with existing spectrum sensing techniques

Study	**Method**	**Features**	**SNR Range**	**Low-SNR Performance (≈ −10 dB)**	**Key Remark**
Atapattu et al., [3]	Energy Detection	Signal energy	−20 to 10 dB	Poor Pd due to noise uncertainty (SNR wall)	Cannot separate noise and weak PU signals
Jan & Koo, et al.,[6]	SVM classifier	Statistical features	−15 to 10 dB	Moderate Pd, degrades below −8 dB	Limited robustness without spectral features
Geng et al., [14]	Deep learning	Learned spectral features	−20 to 10 dB	High Pd at low SNR	High accuracy but high training and computation cost
FuST-SVM (Present Work)	Fuzzy STFT + SVM/RF	8-band STFT + fuzzified features	−10, 5, 10 dB	Pd ≈ 0.85 @ −10 dB; 92.46% @ 5 dB	Deep-learning-level robustness with low complexity

Table 6 contextualizes the performance of the FuST-SVM pipeline proposed herein against some other spectrum sensing studies based on machine-learning algorithms in the literature. The "Present Work" being FuST-SVM, shows a robust and competitive performance profile across the tested SNR range of -10 to 10 dB, attaining an accuracy of 92.46% at 5 dB SNR using GNU Radio generated QPSK I/Q data and 8-band STFT and fuzzified STFT features. The other works presents a glimpse of the variety of feature extraction techniques (e.g., spectral, statistical, wavelet, energy), classifiers (CNN, SVM), and dataset types (simulated, real).

In order to evaluate the statistical solidness of the reported outcomes, the classification trials were carried out again on many randomized train-test splits, while keeping the same stratification ratio. The accuracy figures reported are the mean accuracy over all runs, with the standard deviations observed being in a narrow range (±0.8% to ±1.3%) for all SNR conditions. This shows that the both pipelines' performance is not only stable but also not too much influenced by random data partitioning.

The differences in precision that are very small, are mainly seen as a result of the different placements of borderline samples close to the decision boundary, especially in cases of low SNR where the signal and noise characteristics are very much mixed up. Even so, the FuST-SVM pipeline has always been ahead of the STFT–Random Forest method in all SNR levels at the same time.

The advantage of FuST-SVM over Random Forest is that it has better accuracy that is not very noticeable (it is approximately 1–2%); however, this is a practically significant case in live applications of spectrum sensing. In cognitive radio systems, the detection accuracy even at a small scale can lead to a substantial decrease in false alarms and missed detections, thus making the system work better in terms of utilization of the spectrum and at the same time reducing the chances of causing interference to the primary users.

Notably, the FuST SVM seems to perform quite consistently across the entire broad SNR range, showing good generalization even at those really poor SNR conditions, which is a great thing considering it could do all this with just very efficient feature extraction and the power of combining fuzzification and SVM. From this, one can place FuST-SVM as one of the finest and most practical frameworks in the entire landscape of ML-based spectrum sensing approaches.

6 Discussion

The experimental results, which are depicted very clearly, show that the suggested pipeline of Fuzzy STFT–SVM (FuST-SVM) always outperforms the STFT–Random Forest (RF) pipeline across all the SNR conditions that have been tested. The most visible improvement can be seen with the noiseless signal condition (−10 dB) where the task of sensing the spectrum is very difficult because of the overwhelming noise and the closer distribution of the signal with that of the noise The heightened resilience of the FuST-SVM system is one of the results of the fuzzy feature engineering stage which is passed at the point of the STFT-derived spectral features. The transfer of the crisp STFT sub-band energies into the fuzzy membership degrees (low, medium, and high) is the main part of the fuzzification process which allows the representation of ambiguous spectral patterns rather than the setting of hard decision boundaries. This representation, which is not very strong, is especially useful at low SNRs since the conventional feature values are considerably altered by noise. Hence, the SVM classifier is able to draw a better decision boundary that is more discriminatory of occupied and idle spectrum states under uncertainty. On the other hand, the STFT–RF pipeline works with the crisp spectral features. The Random Forest classifier is powerful enough to keep noise away and nonlinearities at the same time. Thus, their performance decreases even more noticeably at very low SNR levels because the feature distributions that are becoming less separable. This justifies the observed accuracy gap between the two pipelines at −10 dB, where FuST-SVM is still having the advantage of a reliable detection. The FuST-SVM framework as a whole provides a nice compromise between accuracy and computational cost when one looks at the state of the art in deep learning-based spectrum sensing like convolutional neural networks (CNNs). The methods based on CNNs, even if they sometimes reach higher peak accuracies, are actually requiring a lot of data to be labeled, a lot of time to train and a lot of resources to compute the training. Conversely, the FuST-SVM technique is very able to work with the 8-band STFT features that are compact, with a limited amount of training data and with computational overhead that is significantly lower, thus making it very suitable for real-time and resource-constrained cognitive radio systems. The main drawback of the entire FuST-SVM pipeline is the extra computational burden brought about by the fuzzification procedure, which results in the expanding of each crisp feature into several fuzzy membership values. However, the time and memory analysis of the experiments shows that this overhead is still moderate and by no means unacceptable if it is compared against the consistent accuracy gains that have been realized across all SNR regimes. Furthermore, the application of three fuzzy sets for each feature allows for a good compromise between representational richness and computational efficiency, as it was found that increasing the number of fuzzy sets tends to yield smaller performance gains. In a nutshell, the whole debate verifies that the amalgamation of fuzzy logic with STFT-based spectral features and SVM classification is an efficient way to increase noise immunity without sacrificing the practicality of deployment. Therefore, the FuST-SVM framework is positioned as a very reliable spectrum sensing solution in highly dynamic and noisy wireless environments.

7 Conclusion and future enhancements

This paper presented a complete comparative study of two diverse machine learning pipelines for spectrum sensing under varying Signal-to-Noise ratio (SNR) conditions using QPSK I/Q data generated by GNU Radio. The objective was to assess between the use of STFT features with a conventional RF classifier and those obtained with the FuST-SVM pipeline. With our experimental findings, the FuST-SVM pipeline has proven to be consistently better than the alternatives in low (-10dB), medium (5dB), and high (10 dB) SNR regimes. This pipeline attained an accuracy of 90.65% accuracy at low SNR; rising to the highest of 92.46% accuracies at SNR 5 dB and falling only marginally to 91.22% accuracies at higher SNR of 10 dB. Besides being fairly robust, the pipeline involving STFT plus Random Forest consistently recorded accuracies with a slight margin of inferiority over the tested levels of SNR. The remarkable performance of the FuST-SVM pipeline truly highlighted the synergy of its components during the experimentation. The 8-band STFT features adequately represented the spectral characteristics of the signals.

Fuzzy feature engineering then added the value to this stage; particularly, it helped the SVM classifier better understand ambiguous patterns and deal with uncertainty arising from noisy wireless environments in the real world. The resulting combination with SVM, known for its discriminative power and generalization capability, presents a good, practical solution for spectrum sensing. This research adds to the field of intelligent spectrum sensing by providing a validated and high-performing machine learning framework. Given the consistent behavior of the FuST-SVM pipeline under various noise scenarios, it could be considered for implementation in future cognitive radio systems where it could assure better spectrum utilization and free from interference.

References

[1] S. Haykin, "Cognitive radio: brain-empowered wireless communications," IEEE Journal on Selected Areas in Communications, vol. 23, no. 2, pp. 201–220, Feb. 2005. https://doi.org/10.1109/JSAC.2004.839380

[2] Gams, Matjaž, and Tine Kolenik. 2021. "Relations between Electronics, Artificial Intelligence and Information Society through Information Society Rules" Electronics 10, no. 4: 514. https://doi.org/10.3390/electronics10040514

[3] Atapattu, Saman, Chintha Tellambura, and Hai Jiang. Energy detection for spectrum sensing in cognitive radio. Vol. 6. New York, NY, USA: Springer, 2014.

[4] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications. Pearson Education, 4th ed., 2007.

[5] Khalek, Nada Abdel, Deemah H. Tashman, and Walaa Hamouda. "Advances in machine learning-driven cognitive radio for wireless networks: A survey." IEEE Communications Surveys & Tutorials 26, no. 2, 1201-1237, 2023. https://doi.org/10.1109/COMST.2023.3345796

[6] Jan, Sana Ullah, and In Soo Koo. "Performance analysis of support vector machine-based classifier for spectrum sensing in cognitive radio networks." In 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 385-3854. IEEE, 2018. https://doi.org/10.1109/CyberC.2018.00075

[7] Ahuja, Bhawna, and Gurjit Kaur. "Design of an improved spectrum sensing technique using dynamic double thresholds for cognitive radio networks." Wireless Personal Communications 97, no. 1,821-844, 2017. https://doi.org/10.1007/s40010-019-00595-7

[8] Dibal, Peter Yusuf, Elizabeth N. Onwuka, James Agajo, and Caroline Omoanatse Alenoghena. "Application of wavelet transform in spectrum sensing for cognitive radio: A survey." Physical Communication, 28, 45-57, 2018. https://doi.org/10.1016/j.phycom.2018.03.004

[9] Tailor, Anshul, Mohit Dua, and Pankaj Verma. "Automatic classification of multi-carrier modulation signal using STFT spectrogram and deep CNN." *Physica Scripta* 99, no. 7 , 076009,2024. https://doi.org/10.3390/rs16234550

[10] Pandian, Poornima, Chithra Selvaraj, N. Bhalaji, KG Arun Depak, and S. Saikrishnan. "Machine learning based spectrum prediction in cognitive radio networks." In 2023 International Conference on Networking and Communications (ICNWC), pp. 1-6. IEEE,2023. https://doi.org/10.1109/ICNWC57852.2023.10127512

[11] Dey, Barnali, Ashraf Hossain, Valentina E. Balas, and R. N. Bera. "Improved Energy Detector for Spectrum Sensing Using Neuro-Fuzzy Double Threshold Technique." Studies in Informatics and Control 26, no. 3 , 335-342, 2017

[12] Syed, Sadaf Nazneen, Pavlos I. Lazaridis, Faheem A. Khan, Qasim Zeeshan Ahmed, Maryam Hafeez, Antoni Ivanov, Vladimir Poulkov, and Zaharias D. Zaharis. "Deep neural networks for spectrum sensing: A review." IEEE access 11, 89591-89615,2023. https://doi.org/10.1109/ACCESS.2023.3305388

[13] Arjoune, Youness, and Naima Kaabouch. "On spectrum sensing, a machine learning method for cognitive radio systems." In 2019 IEEE International Conference on Electro Information Technology (EIT), pp. 333-338. IEEE, 2019., https://doi.org/10.1109/EIT.2019.8834099

[14] Geng, Yue, Jingyi Huang, Jianxin Yang, and Sen Zhang. "Spectrum sensing for cognitive radio based on feature extraction and deep learning." In Journal of Physics: Conference Series, vol. 2261, no. 1, p. 012016. IOP Publishing, 2022. DOI 10.1088/1742-6596/2261/1/012016

[15] Wu, Qingying, Benjamin K. Ng, and Chan-Tong Lam. "Energy-efficient cooperative spectrum sensing using machine learning algorithm." Sensors 22, no. 21, 8230. 2022. https://doi.org/10.3390/s22218230

[16] Zhang, Yixuan, and Zhongqiang Luo. "A review of research on spectrum sensing based on deep learning." *Electronics* 12, no. 21 (2023): 4514. https://doi.org/10.3390/electronics12214514

[17] El Omari, Khalil, Aziz Dkiouak, Baghouri Mostafa, and Saad Chakkor. "Comparing fuzzy logic methods for performing spectrum sensing in cognitive radio." In IET Conference Proceedings CP859, vol. 2023, no. 44, pp. 276-281. Stevenage, UK: The Institution of Engineering and Technology, 2023. https://doi.org/10.1049/icp.2024.0938

[18] Wang, Danyang, Ning Zhang, Zan Li, Feifei Gao, and Xuemin Shen. "Leveraging high order cumulants for spectrum sensing and power recognition in cognitive radio networks." IEEE Transactions on Wireless Communications 17, no. 2 (2017): 1298-1310. https://doi.org/10.1109/TWC.2017.2777488

[19] Jan, Sana Ullah, and In Soo Koo. "Performance

analysis of support vector machine-based classifier for spectrum sensing in cognitive radio networks." In 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 385-3854. IEEE, 2018. https://doi.org/10.1109/CyberC.2018.00075

[20] Abusubaih, Murad A., and Sundous Khamayseh. "Performance of machine learning-based techniques for spectrum sensing in mobile cognitive radio networks." IEEE Access 10, 1410-1418, 2021. https://doi.org/10.1109/ACCESS.2021.3138888

[21] Reddy DN, Priyanka R, Sanjana S, Santrupti MB, Sadiya S. Machine learning algorithms for detection: A survey and classification. Turkish Journal of Computer and Mathematics Education. 2021;12(10):3468-74. https://www.turcomat.org/index.php/turkbilmat/article/view/11214