SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V13I3P104

Volume 13 Issue 3, 38-50, March 2026

Original Article

Automatic Modulation Classification Using Time-Frequency Features in a GNU Radio-Based Framework

Raman R[1*], Deepa N Reddy[2], S.Vishali[3], Rama Krishna Prasadh H[4]

[1,2,3,4]*Department of Electronics and Communication, BMS Institute of Technology and Management, Bengaluru, Karnataka, India*

[1]*Corresponding Author : ramanrajesh1111111@gmail.com*

Received: 06 December 2025 Revised: 06 January 2026 Accepted: 05 February 2026 Published: 31 March 2026

Abstract - *Automatic Modulation Classification (AMC) is one of the most crucial functions in the advanced wireless communication systems. Mainly, it is used in applications such as cognitive radio, dynamic spectrum access, and interference monitoring, where the knowledge of the transmitted signals is lacking. In the current work, we scrutinize an AMC framework with BPSK, QPSK, and 16-QAM as candidates for modulation under diverse Signal to Noise Ratio (SNR) conditions in detail. A synthetic dataset is produced by means of self-made GNU Radio flowgraphs with the addition of white Gaussian noise in order to mimic real-world wireless transmission situations. Time-frequency features are retrieved with the Short-Time Fourier Transform (STFT) and discrete wavelet transform, and three classical machine learning algorithms, Naive Bayes, Random Forest, and Support Vector Machine (SVM), are applied to classification. The framework is rated in terms of both classification accuracy and computational efficiency at varying SNR levels. The outcomes show that SVM coupled with wavelet features reaches the peak classification accuracy of 97.0% at 10 dB SNR, while the Naive Bayes classifier is the fastest in training time, and hence it can be deployed in real-time and resource-scarce applications. The findings indicate that combining lightweight machine learning methods with suitable time-frequency features can be an effective and efficient alternative to deep learning AMC frameworks in terms of computational cost.*

Keywords - *Modulation Classification, Wavelet Transform, Short-Term Fourier transform, Machine learning, Signal to Noise Ratio.*

1. Introduction

Automatic Modulation Classification (AMC) is one of the important properties of smart wireless receivers, which can differentiate the modulation types, even if the transmission conditions are not predetermined. This kind of feature is so vital in cognitive radio, electronic warfare, dynamic spectrum access, and interference detection that it allows the receivers to operate correctly even under uncertain and noisy channel conditions [1]. The same receiver can adapt its demodulation and decoding techniques based on the exact modulation type, thereby improving overall communication efficiency and robustness.

The majority of the traditional AMC approaches are based on likelihood-based and decision-theoretic techniques. Although these techniques assure the best performance possible in ideal situations, their efficiency drops drastically in day-to-day scenarios where channel impairments and noise are present. Besides, the exceptionally high precision required to estimate signal and channel parameters makes their application in practical systems very difficult. Difficulties mentioned above have made feature-based and machine learning–based AMC methods the primary focus of research due to their resilience and adaptability in complicated environments [5, 6]. On the contrary, feature-based AMC strategies involve picking out the most distinguishing features from the incoming signals and then classifying the signals with classifiers according to the modulation type. Time-frequency representation is one of the various feature extraction methods that has led to great success, specifically because it can illustrate both the temporal and spectral characteristics of non-stationary communication signals [2, 3, 12]. The Short-Time Fourier Transform (STFT) provides a localized frequency representation through fixed windowing, whereas wavelet transformations provide a localized frequency representation through fixed windowing.

In the past few years, the use of deep learning–based AMC frameworks with Convolutional Neural Network (CNN) on spectrograms or scalograms has led to very high classification accuracy. Still, wide-ranging labeled datasets, long training, and powerful computing resources are often prerequisites for these methods, thus limiting their use in on-the-fly or low-end scenarios. While deep learning models are at their spinning peak, classical machine learning models have not let go of the market and still remain an option for their interpretability, low computational complexity, and easy implementation in resource-starved environments [5, 6].

This paper is inspired by these pros and cons and proposes a comparative AMC framework where STFT- and wavelet-based time–frequency features are combined with classical machine learning classifiers. The present work uses similar time–frequency representations to those used in deep learning-oriented studies. However, it emphasizes the lightweight classification models such as Naive Bayes, Random Forest, and Support Vector Machine to explore the trade-off between classification accuracy and computational efficiency. A synthetic dataset that closely resembles actual wireless transmission chains is produced using GNU Radio to facilitate controlled experiments at various SNR levels. Most deep learning–based AMC studies utilize large benchmark datasets such as RadioML, which need extensive computational power to train their Convolutional Neural Networks (CNN). The existing research lacks a proper assessment of traditional machine learning classifiers because it fails to analyze modern time-frequency features in controlled software-defined radio environments while measuring computational expenses, which require extensive computational resources to run Convolutional Neural Networks (CNN).

The research field has not conducted comprehensive studies that assess how traditional machine learning classifiers perform when they use contemporary time-frequency analysis methods in controlled environments that simulate software-defined radio signal generation. Existing AMC studies fail to address the essential aspect of analyzing computational expenses. The study needs to determine whether lightweight classical machine learning models can achieve classification accuracy that matches their performance when they use discriminative time-frequency features derived from STFT and wavelet transforms. The study introduces a new research framework that enables researchers to create synthetic datasets using the reproducible GNU Radio method. The study conducts an extensive evaluation of STFT and wavelet features across various traditional classifiers, which include Naive Bayes, Random Forest, and SVM, while assessing the computational expenses, which change with different SNR levels.

The main aim of this research is to evaluate the performance of various feature–classifier combinations in different noise conditions and to discern the configurations that cater to the real-time and resource-constrained scenarios. Summed up, the contributions of this research are:

- An AMC system that combines STFT and wavelet-based feature extraction, along with different classical machine learning classifiers, has been created and assessed.
- The BPSK, QPSK, and 16-QAM signals at various SNR levels are generated using GNU Radio flowgraphs, resulting in a realistic synthetic dataset.
- The trade-offs among the various machine learning methods are revealed through the detailed comparison of classification accuracy, robustness, and computational cost.
- It is possible to see from the results that the traditional machine learning models can deliver high-performance competitive with the state-of-the-art methods if used along with suitable time-frequency features, and at the same time, they have low computational complexity.

This paper is organized in the following manner: Section 2 provides the relevant literature and associated methods, Section 3 outlines the proposed methodology and Signal set generation techniques. The experimental results are reflected in section 4 and include the analysis of performance for both the current methods and the suggested ones; eventually, the entire study is concluded in section 5.

2. Research Contributions

The significant contributions in this work are summarized as follows:

- The research created a repeatable synthetic dataset that uses GNU Radio to generate BPSK, QPSK, and 16-QAM signals at specific SNR levels.
- The study evaluated STFT and wavelet time–frequency features using three basic machine learning classifiers to assess their performance.
- The study used a grid search to find optimal SVM hyperparameters across various SNR conditions.
- The research measured computational costs by evaluating training duration and memory usage.

The study compares its results with modern deep learning–based AMC systems.

3. Related Work

3.1. Categorization of AMC Techniques

Existing automatic modulation classification techniques can be broadly categorized into three groups: (i) likelihood, (ii) feature-based machine learning, and (iii) deep learning approaches. AMC has been identified as an important area in wireless communications systems, leading to the study of various methods. In order to cope with model complexity, current research has been concentrating on lightweight and hybrid models, together with data augmentation, attention mechanisms, and semi-supervised learning tests. A light-weight neural network augmented with hybrid data to simultaneously maintain accuracy and performance is proposed in [2]. In contrast, fractional S-transform features in combination with a semi-supervised scheme are proposed in [3] to escalate the performance initially constrained by the presence of little labeled data. Moreover, the use of zero-shot learning methods to detect new types of modulations that were not seen before, based on the properties learned from the Signal, has been thoroughly studied [4].

3.2. Likelihood-Based Approaches

The first techniques used for AMC depended mainly on likelihood-based and decision-theoretic approaches requiring prior knowledge of primary parameters and channel conditions. Though these methods yield an optimum result when ideal conditions are assumed, their noise sensitivity and dependency on the channel deteriorations limit their application in real wireless environments [5, 6].

3.3. Feature-Based Machine Learning Approaches

In order to improve the methods, feature-based AMC techniques were created, which dealt with identifying and isolating the characteristics of the received signals. The use of statistical features, higher-order cumulants, and cyclostationary properties has been prominent in modulation recognition. Still, many of these features show performance degradation when the Signal-To-Noise Ratio (SNR) is low and, most of the time, need meticulous manual design [7, 10].

Time-frequency analysis has been among the most significant techniques in the area of automatic modulation classification due to the fact that it can take non-stationary signals and provide their temporal and spectral characteristics. The Short-Time Fourier Transform (STFT) has been applied to generate spectrogram-based features, yielding classification performance superior to that of time-domain or frequency-domain representations alone [6]. However, the fixed window size of STFT engages in a trade-off between the time and the frequency resolution, which can, under the quickly changing signal conditions, put a limitation on its effectiveness.

3.4. Deep Learning-Based Approaches

Deep learning has become a significant force in the use of CNNs for automatic modulation classification, on the assumption that spectrograms or scalograms are image-like representations for the input. As a consequence of reliance on Neural Network techniques, questions of classification accuracy have been thoroughly addressed, with the best possible solution achieved across an extensive range of modulation scenarios and channel conditions [9, 14]. The benefits of wavelet-based feature extraction have been demonstrated through their provision of multiresolution time-frequency analysis, which improves performance. Traditional methods combined with wavelet transformations and deep learning models further improved the performance of the classifier, especially at moderate and high SNR levels [17]. Nevertheless, the high accuracy that these methods provide comes with a need for large labeled datasets, a long training time, and considerable computational power, thereby limiting their use in real-time or constrained-resource applications. A comprehensive review of the machine learning algorithms for AMC is given in [20]. An approach to AMC using a threshold denoiser technique with CNN is detailed in [21].

The recent studies in [22] have used STFT-based representations along with deep learning architectures, but the methodology and objective of the present work are fundamentally different. The present study, on the contrary, is centered around machine learning classifiers that are classical in nature and rely on wavelet and STFT features, thus emphasizing interpretability, computational efficiency, and real-time feasibility more than ever. While deep learning methods prefer high accuracy with associated training cost and hardware requirements, the proposed framework aims to achieve similar performance using lightweight classifiers, thereby making it a candidate for resource-constrained and embedded communication systems.

3.5. Identified Research Gap

Even though the frameworks for AMC based on deep learning are the most talked about in the recent literature, classical machine learning techniques have not yet relinquished their advantages in the areas of interpretability, computational efficiency, and ease of deployment.

Nevertheless, a smaller number of studies have been conducted to systematically evaluate classical classifiers by applying modern time-frequency features in controlled, reproducible settings. Deep learning–based AMC frameworks largely dominate the most recent literature, but classical machine learning methods still provide important benefits regarding interpretability, computational efficiency, and deployment. Comparatively fewer studies, however, have thoroughly evaluated classical classifiers using modern time–frequency features under controlled, reproducible conditions. Additionally, the comparison of traditional machine learning models against deep learning–oriented approaches using synthetic datasets generated through software-defined radio platforms has been minimal.

The current study, motivated by these research gaps, presents a comparative evaluation of STFT- and wavelet-based feature extraction combined with classical machine learning classifiers. This work will use synthetic datasets generated by GNU Radio and will look at classification performance and computational cost across different SNR levels to provide a reproducible and lightweight alternative to current deep learning–centric AMC frameworks.

4. Proposed Methodology

4.1. Signal Generation and Dataset Preparation

This part of the paper presents a detailed account of the methodology applied for automatic modulation classification, which includes synthetic signal generation, preprocessing, feature extraction, dimensionality reduction, and machine learning-based classification. One of the primary aims of this research is to measure the performance of the classical classifiers when using time-frequency features in different signal-to-noise ratio conditions. The synthetic training set was created using custom flowgraphs designed in the GNU Radio Companion (GRC) environment. These flowgraphs were set up to generate signals that were similarly affected by the real wireless communication conditions, hence, the source of the synthetic training set. Three digital modulation types were included in the training set: BPSK (Binary Phase-Shift Keying), QPSK (Quadrature Phase-Shift Keying), and 16-QAM (16-Quadrature Amplitude Modulation).

For each modulation type, 1000 signal samples were generated, yielding a total of 3,000. The SNR levels of −10 dB, 5 dB, and 10 dB were selected to check the system's performance in the presence of noise. The SNR levels correspond to low, medium, and high noise environments, respectively. The carrier frequency was fixed at 2 kHz, whereas the sampling frequency was set at 32 kHz. The Signal was then saved in .dat format using file sink blocks, and further processing and analysis were done offline. The dataset was split in a manner that enabled a fair evaluation. For each modulation type and SNR level, 300 samples were used for training and validation, and 100 for testing. This division ensured that no part of the dataset used for testing overlapped with the model training phase.

Table 1. Synthetic dataset configuration parameters

Parameter	Description
Dataset type	Synthetic (GNU Radio generated)
Modulation schemes	BPSK, QPSK, 16-QAM
Samples per modulation	1000
Total samples	3000
SNR levels	−10 dB, 5 dB, 10 dB
Channel model	AWGN
Sampling rate	32 kHz
Carrier frequency	2 kHz
Downsampled rate	4 kHz
Feature extraction	STFT, Wavelet (db4, level-5)
Dataset format	.dat files

The BPSK signal in discrete time is mathematically given as:

$$s[n] = A\, m[n]\, cos(2\pi f_c n T_s) \quad (1)$$

Where m[n] ϵ{-1,+1}, A is the amplitude, f_c is the normalized carrier frequency, T_s is the sampling interval, and n is the discrete time index.

The QPSK signal is mathematically given as:

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

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

The 16-QAM modulation 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,±3}, 16 possible combinations for each symbol.

A channel model block was included in each GNU Radio flowgraph to simulate realistic wireless communication conditions. In the channel model block, since noise had to be inserted into the channel, an AWGN source was included, with the noise voltage parameter set at various values dependent upon the required SNRs of -10 dB, 5 dB, and 10 dB. These SNR levels exhibited the characteristics of the environments with high, medium, and low interferences in that order. This technique is a very efficient simulation of the channel imperfections of the actual systems, which can be regarded as a more realistic and convenient method for performance evaluation of the modulation classification model in different noise scenarios.

The signals created from the input data set can be used for further processing, thus allowing a controlled, systematic evaluation of the performance of modulation classification under varying noise conditions to take place.

4.2. Preprocessing of Signals

The signals that were generated were downsampled from 32 kHz to 4 kHz to reduce the computational complexity. The downsampling led to an eightfold decrease in the size of the data but did not drastically alter the classification needed frequency and phase information. After the downsampling process, the amplitude normalization technique was used to obtain uniformity in the SNR conditions. This preprocessing step helped to lower the influence of the higher-energy signals, and at the same time, it contributed to the robustness of feature extraction and classification.

4.3. Feature Extraction Technique

The aim was to create an enriched feature space through the STFT and wavelet transform.

- Short-Time Fourier Transform (STFT): A Hamming window of 256 samples was used, allowing only 50% overlap in the application of STFT to each Signal. The procedure leads to the creation of a spectrogram that depicts the frequency content over time for a specific location. The resulting time-frequency matrix was flattened and normalized to create a feature vector of size 129 × 150 for each signal instance. These features reflect the spectral changes that are typical of different modulation schemes.

- Wavelet Transform: For the extraction of multiresolution time–frequency features from the signals, the Discrete Wavelet Transform (DWT) was applied. Daubechies-4 (db4) wavelet was chosen because it is considered to be the best for non-stationary communication signal analysis. The decomposition was done up to level five, which resulted in the generation of approximation and detail coefficients that represent both low- and high-frequency components. The wavelet coefficients thus obtained were stacked up to create a feature vector of length 4096 for each signal sample.

4.4. Feature Analysis and Dimensionality Reduction

- The large size of STFT and wavelet feature vectors had an adverse effect on computational cost by increasing it, and also caused overfitting. To perform dimensionality reduction, PCA was used, and it kept the principal components that contributed 95% to the total variance of the data. Dimensionality reduction, on the one hand, led to better accuracy and, on the other hand, also to less time needed for calculations. These factors were quantified by evaluating classifier performance on the original and PCA-reduced feature sets.

4.5. Machine Learning Classification

The feature extraction process outputs three classical machine learning algorithms' results, one from each of them: Naive Bayes based on the assumption that features are conditionally independent, Random Forest as a robust model utilizing majority vote among several decision trees, and SVM as the one with the most efficient non-linear separations with its Radial Basis Function (RBF) kernel. The entire process was done using the Scikit-learn library, and the same train-test split was applied for all the classifiers to keep the comparison fair.

4.5.1. Naive Bayes Classifier

The Naive Bayes classifier was chosen because of its low overhead during computation and quick learning time, which made it enjoyable for real-time and low-capability-resource applications.

$$P(C_k|x[n]) = \frac{P(C_k)\prod_{i=1}^{N} P\left(x_i[n]\middle|C_k\right)}{P(x[n])} \tag{4}$$

4.5.2. Random Forest Classifier

Having been built with 100 trees, it was still commonly preferred on account of general resistance to over-fitting, and also for its competence when dealing with data of high dimension. It is an ensemble-based algorithm where hundreds of decision trees are built, and the outputs are integrated for the final prediction. It is immune to overfitting and thus enjoys the strength of reducing very high-dimensional data, coupled with noise tolerance, which is typically very useful during modulation classification under varying SNR conditions.

$$\hat{y}[n] = mode\left((h_1 x[n]), h_2(x[n]), \ldots \ldots, h_T(x[n])\right) \tag{5}$$

Total number of trees T=100

4.6. Support Vector Machine (SVM)

The SVM classifier employed a Radial Basis Function (RBF) kernel, which facilitated the efficient modeling of non-linear decision boundaries.

The SVM hyperparameters were optimized through a joint process of grid search and cross-validation, allowing the regularization parameter and kernel width to reach their optimal values. This not only increased the classification accuracy but also maintained the model's generalization ability across various SNR levels.

$$f[n] = \mathrm{w}^T \mathrm{x}[n] + b, \hat{y}[n] = \begin{cases} +1, & \text{if } f[n] > 0 \\ -1, & \text{if } f[n] < 0 \end{cases} \tag{6}$$

$$K\left(x_i, x_j\right) = \exp(-\gamma\left\|x_i - x_j\right\|^2 \tag{7}$$

where γ controls the influence of each sample, and C regulates the trade-off between margin width and misclassification.

4.7. Evaluation Strategy

To guarantee the stability and reliability of the analysis framework, the performance of each classifier combined with a given set of features was evaluated by k-fold cross-validation (5 folds). The benchmarks used for reporting results were accuracy, precision, recall, and F1-score. The analysis involved an ablation study to compare the performance produced by STFT features alone against that of wavelet features alone with each classifier at all SNR levels, shedding light on the contribution of each feature set. The performance was measured at all 3 SNR levels in arriving at proving the robustness and shortcomings of the model.

4.8. Computational Cost Analysis

To provide a reproducible benchmark, the computational cost was measured in terms of training time and memory usage. All experiments were conducted on a machine with the following specifications: an Intel Core i5-10th Gen CPU, 16 GB of RAM, and a Windows 11 OS.

The computational complexity of Naive Bayes is approximately O(n), but Random Forest needs more time to process because its complexity increases to O(n·T) with every additional tree. The training process for SVM needs $O(n^2)$, which results in longer training times. The proposed classical models need less than one second of CPU training, while CNN-based AMC models in the literature take multiple minutes to several hours of GPU training time.

5. Implementation

5.1. Signal and Dataset Generation

The initial step involved generating a signal dataset synthetically through some flowgraphs implemented in GNU Radio Companion (GRC). Separate flowgraphs were created for the three modulation schemes-BPSK, QPSK, and 16-QAM. Each flowgraph emulates a signal transmission chain, containing signal sources, modulation blocks, and a channel block. There was an Additive White Gaussian Noise (AWGN) source inside the channel block, with voltage varying to vary three distinct SNR levels: -10 dB, 5 dB, and 10 dB, respectively. Signals were generated at a rate of 32 kHz and 2 kHz as a carrier frequency. The output of the flowgraph was fed into File Sink blocks that store the data in .dat files. The overall workflow of the proposed automatic modulation classification framework is shown in Figure 1. The first step of the whole process is the creation of signals using GNU Radio for different modulation schemes under different SNR conditions. These signals are then preprocessed, downsampled, and feature extraction is performed using STFT and the discrete wavelet transform. The last steps include the application of classical machine learning classifiers and the evaluation of system performance with the help of standard metrics.

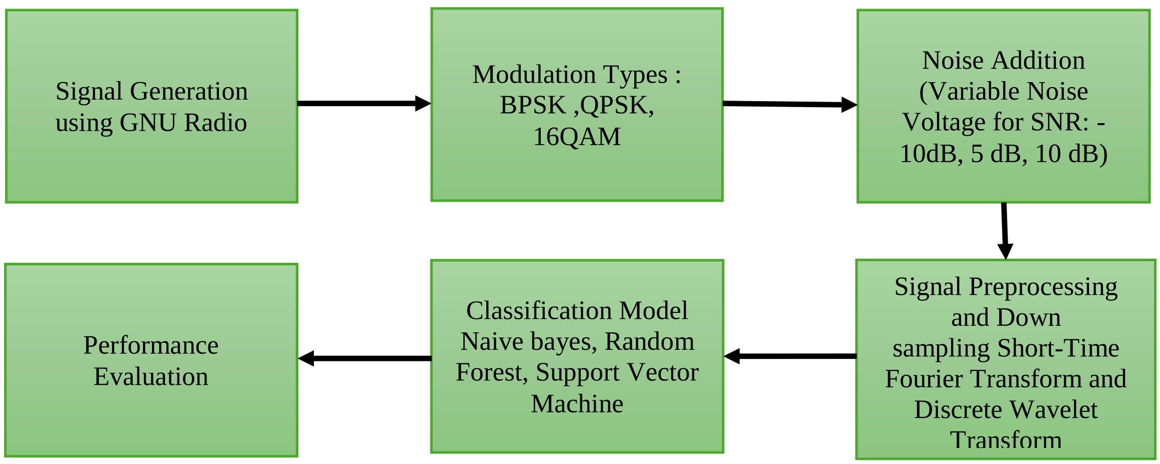

Fig. 1 Process flow diagram

The strong dataset was then created by mixing many of these generated signals. Without downsampling, computational complexity would become an issue; hence, the data was downsampled from such a high sampling rate to a 3,000-sample size, with 1,000 samples belonging to each modulation channel.

Downsampled data were split into a training/validation set (900 samples) and a held-out testing set (100 samples). This enabled a controlled, systematic, and efficient platform for testing our classification model.

5.2. Feature Extraction

5.2.1. Short-Time Fourier Transform (STFT)

In this study, feature extraction was investigated with the STFT. This method is quite a potent technique in the field of temporal-frequency analysis of signals, where one seeks to understand the time evolution of the frequency content of such a signal. The process of using STFT is accomplished by splitting the Signal into windows, overlapping time windows, and applying the Fourier transform. This builds up a two-dimensional space in time frequency at a given solution point, where a point signal marks the amplitude of a specific frequency versus certain times. This enables viewing the signal dynamic frequency behavior in passing/resourcefulness.

$$STFT\,\{x[n]\}(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-mR]e^{-j\omega m} \tag{8}$$

The window function and its size affect the time and frequency resolution of the STFT directly. A Hamming window balances the space-time resolution from spectral leakage in our implementation.

The normalized flattened STFT features were put forth as input features for machine learning classifiers. This allowed the model to effectively learn or differentiate BPSK, QPSK, and 16-QAM modulation types under different SNR levels.

With a focus on STFT, the model manifests both clear visualization and extraction in temporal frequency shifts representative of different modulation techniques. The initial step in the system is signal generation through GNU Radio, where all modulated signals (BPSK, QPSK, and 16-QAM) are generated under different noise conditions, for instance, at SNR levels of -10 dB, 5 dB, and 10 dB. The generated signals are first stored and then subjected to processing for the extraction of characteristics. Short-Time Fourier Transform (STFT) is employed to get a time–frequency representation that mirrors localized spectral variations tied up with different modulation schemes. These spectral properties are still informative in a noisy environment and hence are good for classification purposes. After that, the extracted features are put into practice for the training and testing of machine learning classifiers for modulation recognition.

5.2.2. Wavelet Transform

As an alternative to one-dimensional signal data, the wavelet transform is used as a feature extraction method. It is especially suitable for non-stationary communication signal analysis, due to its capability of providing both time and frequency resolution at the same time. It is the other way around compared to classical Fourier-based methods, where wavelet analysis shows the signal behavior at several different resolutions, allowing the capture of both slow-moving and transient elements. The Discrete Wavelet Transform (DWT) splits the original Signal into approximation and detail coefficients through several levels using a chosen wavelet basis.

Let x[n] be a discrete signal.

At each level, j DWT decomposes the Signal as:

The recursive general equations are:

1. Approximation coefficients at level j:

$$a_j[n] = \sum_k a_{j-1}[k].h[2n-k] \tag{9}$$

2. Detail coefficients at level j:

$$d_j[n] = \sum_k a_{j-1}[k].g[2n-k] \tag{10}$$

Where $a_0[n]=x[n]$ where x[n] is the original input signal, h[k] low-pass filter (scaling function), g[k] is high-pass filter (wavelet function), $a_j[n]$ is approximation at level j, $dj_{[}n]$ is detail at level j, 2n denotes downsampling by 2. The block-level GNU Radio flow graph for BPSK signal generation is illustrated in Figure 2. The flow graph comprises a random source, a BPSK modulator, and a pulse-shaping filter in sequence. A noise simulation model, called AWGN channel, is used for the noise simulation under varying SNR levels, and then the Signal is saved through a file sink for subsequent feature extraction and classification.

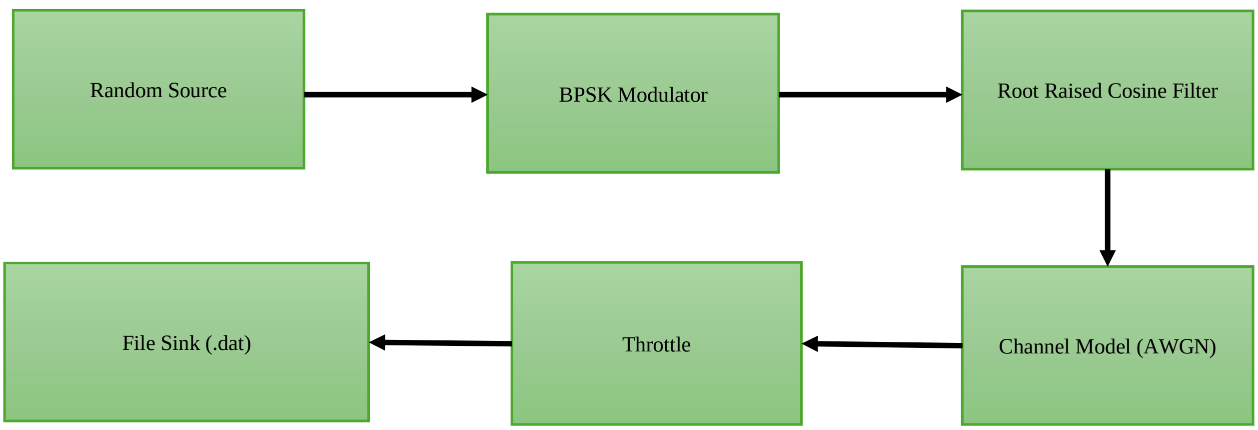

Fig. 2 GNU radio flow graph for BPSK signal generation

The feature extraction performed exclusively with the wavelet method provided the time-frequency representation of an input signal, which successfully captured sharp frequency changes occurring in a signal even in noisy conditions. Wavelet features were extracted in Python using the feature extraction modules within the NumPy and SciPy libraries.

5.3. Classification

The extracted features were used to train and test three different classifiers: Naive Bayes, Random Forest, and SVM. All implementations were done through the Scikit-learn library, and a particular stratified train-test split was followed across all the models, integrating a fair comparison.

The performance classification was carried out using confusion matrices and scores of accuracy for each SNR, known by the types of modulation. The modularity of the pipeline made it easy for any experimentation because by just changing the file paths in the Python script, the signal files can be replaced to evaluate the pipeline in different SNRs and modulations.

5.4. Tools and Environment

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

5.4.1. Python Environment

Python is used mainly for Signal processing, 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, which is expected to make it an appropriate choice for this project. Version 3.10 was used in this work.

5.4.2. GNU Radio

The major area of symptom and modulation simulation BPSK, QPSK, and 16QAM, had to design a 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.

5.4.3. 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.

6. Results and Discussion

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

6.1. 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

6.1.1. 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).

6.1.2. Metrics

Table 2. Metrics used for performance evaluation

Metric	Equation
Precision	$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}}$
Recall	$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP+FN}}$
F1-Score	$\mathbf{F1\ score} = \frac{\mathbf{2TP}}{\mathbf{2TP+FP+FN}}$
Accuracy	$\mathbf{Accuracy} = \frac{\mathbf{TP+TN}}{\mathbf{TP+TN+FP+FN}}$

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

6.1.3. 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, such as software-defined radios or embedded systems.

Table 3 indicates that the Naive Bayes classifier incurs the least computational cost among all the classifiers, as it only requires less than 0.02 s for training and approximately 476 MB of memory. Consequently, it is capable of being employed for real-time applications, though its classification accuracy is still below that of the SVM.

Table 4 summarizes the grid search–based RBF Kernel Hyperparameter Selection for Different SNRs and the best cross-validation accuracy achieved.

From the confusion matrices in Figures 3(a),(b), and (c) for wavelet features at 10 dB SNR, each matrix shows very good diagonal dominance, implying that most of the true modulation types (16QAM, BPSK, QPSK) were correctly predicted. Few off-diagonal elements mean near-zero misclassifications, further corroborating that in a high signal-to-noise environment and using wavelet features, the system attains a highly effective and accurate discrimination. The ROC curves are provided in Figures 4 (a),(b) and (c) for all modulation types with Naive Bayes and Random Forest almost stay at the top left corner of the plot, implying very high true positive rates coupled with very low false positive rates, while the associated AUC (Area Under the Curve) values stand very close to 1.0 (e.g., above 0.99 for most classes), which in essence means that the classifiers separate almost flawlessly between different modulation types when wavelet features were at play.

Table 3. Computational cost comparison of classifiers (wavelet features, 10 dB SNR)

Classifier	Training Time (s)	Memory Usage (MB)
Naive Bayes	0.02	476
Random Forest	0.18	812
SVM	0.41	1024

Wavelet-based features used with the tested classifiers show very good discrimination power even at 10 dB SNR. Table 5 shows that SVM accuracy improves with SNR: 35% at –10 dB, 74% at 5 dB, and 96.7% at 10 dB. Optimal values of C and γ, selected via cross-validation systems, significantly impact performance**.**

The data presented in Table 5 give a summary of the classification performance of the chosen classifiers on the synthetic dataset as regards BPSK, QPSK, and 16-QAM modulation schemes. The best performance among all configurations is SVM with wavelet features, which gives the highest accuracy of 97.0% at 10 dB SNR. It is followed by Naive Bayes and Random Forest classifiers, which are less accurate, using either STFT or wavelet features. The gain in performance comes from the SVM's capability in non-linear decision boundary modeling and the capability of wavelet features in characterizing transient signal qualities.

Table 4. Grid search–based RBF kernel hyperparameter selection for different SNRs

SNR (dB)	Best C	Best gamma	Kernel	Best Cross-Validation Accuracy
-10	1	0.01	Rbf	0.35
5	0.1	scale	Rbf	0.7396
10	0.1	1	Rbf	0.9671

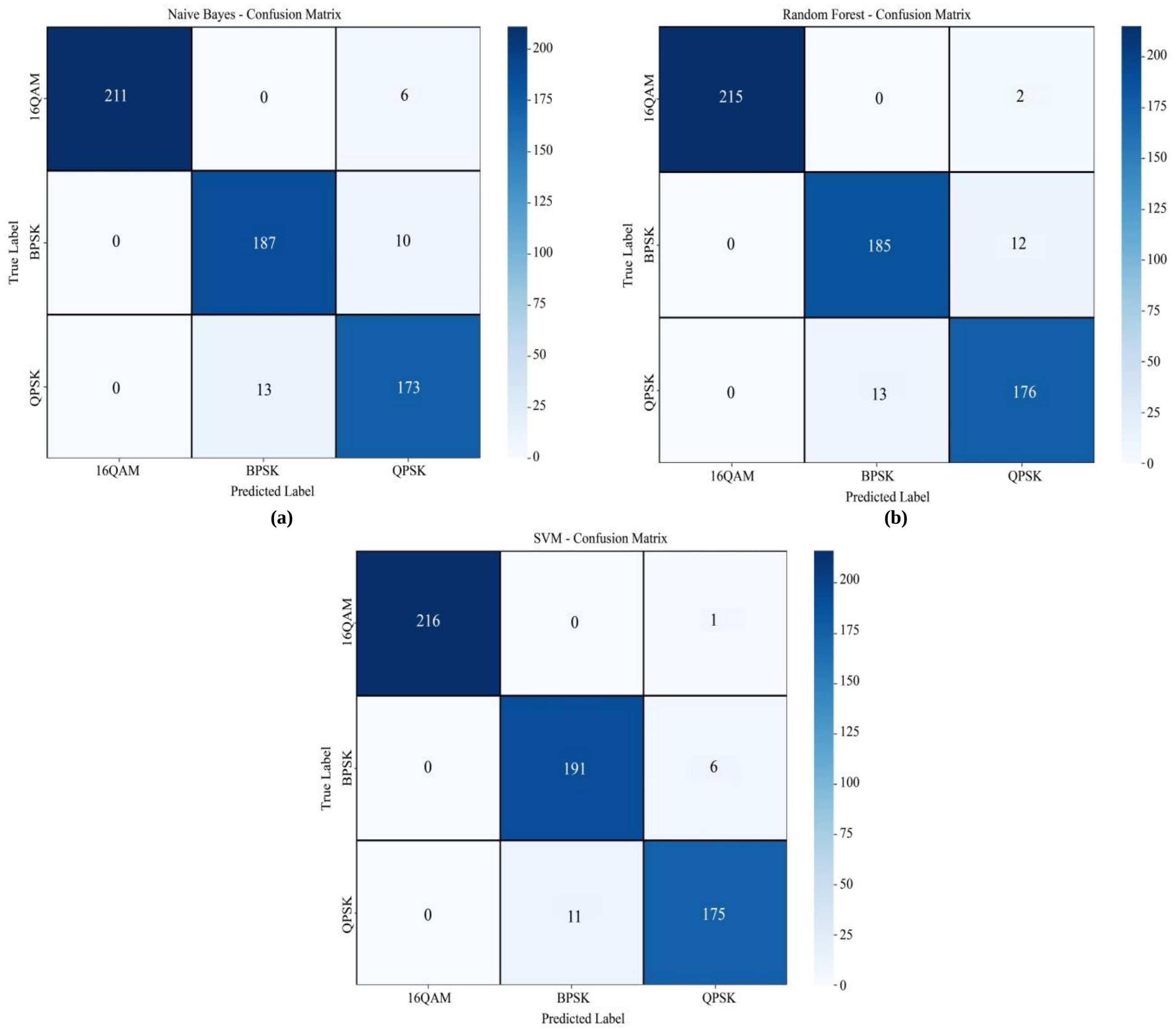

Fig. 3(a) Confusion matrix of Naive Bayes using wavelet features at SNR of 10dB, (b) Confusion matrix of Random Forest using wavelet features at SNR of 10dB, and (c) Confusion matrix of SVM using wavelet features at SNR of 10dB.

Table 5. Ablation study of feature extraction techniques

Feature Extraction	SNR in dB	Method	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
STFT	-10	Naive Bayes	32.67	33.00	32.00	31.00
		Random Forest	33.83	34.00	34.00	34.00
		SVM	31.50	32.00	32.00	32.00
	5	Naive Bayes	69.00	69.00	68.33	68.33
		Random Forest	60.50	60.33	60.33	60.33
		SVM	67.50	67.33	67.00	67.00
	10	Naive Bayes	95.67	95.00	95.00	95.60

		Random Forest	95.17	95.00	95.00	94.60
		SVM	96.17	96.00	96.00	96.00
Wavelet	-10	Naive Bayes	30.83	32.00	31.33	29.00
		Random Forest	33.33	33.66	33.33	33.33
		SVM	32.83	33.66	33.33	32.33
	5	Naive Bayes	76.17	76.00	75.33	75.00
		Random Forest	73.83	73.00	73.33	73.33
		SVM	76.67	76.66	76.33	76.00
	10	Naive Bayes	95.17	95.33	95.00	95.00
		Random FOREST	96.00	96.00	96.00	96.00
		SVM	97.00	97.00	97.00	97.00

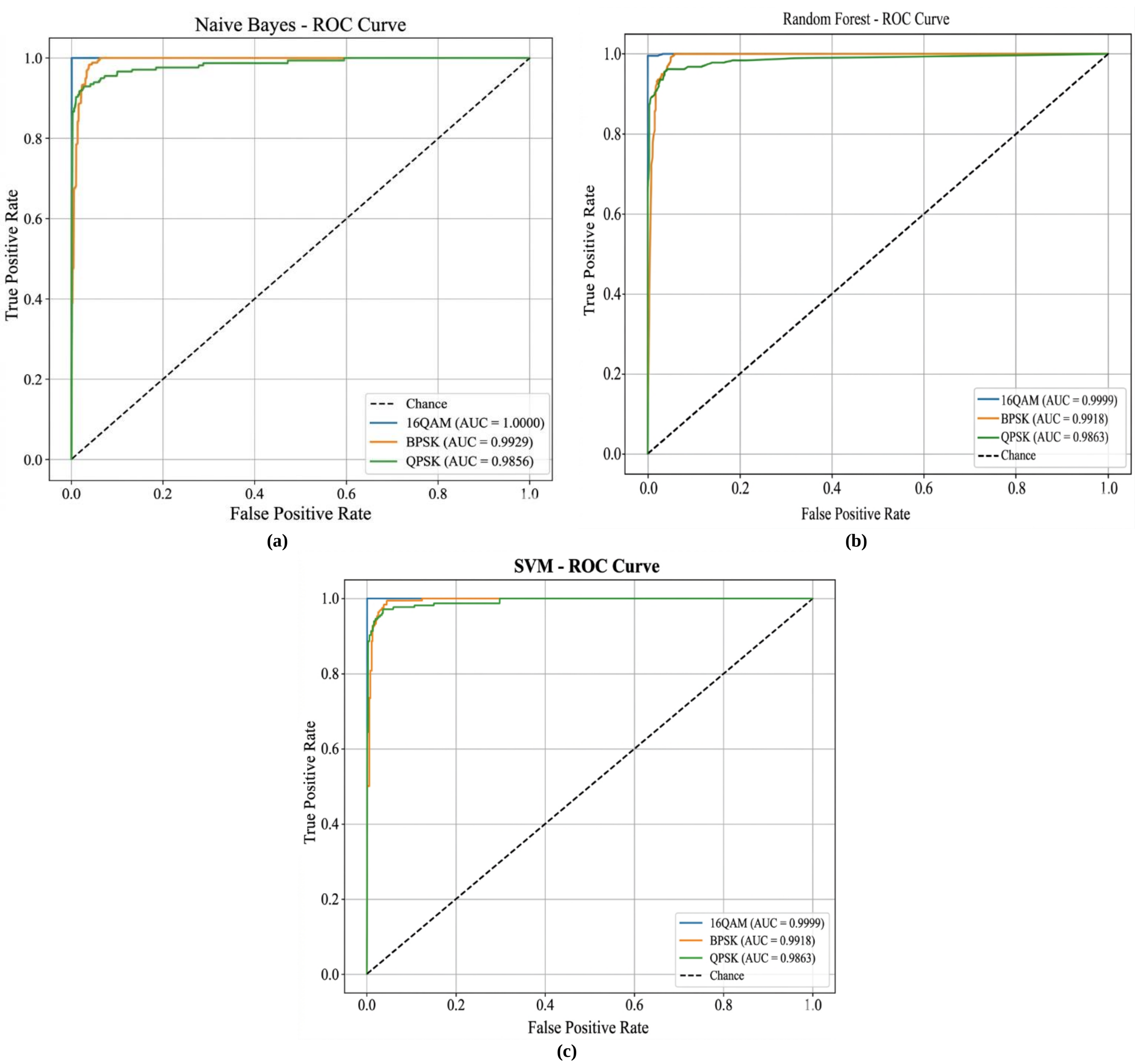

Fig. 4(a) ROC curve of Naive Bayes using wavelet features at SNR of 10dB, (b) ROC curve of Random Forest using wavelet features at SNR of 10dB, and (c) ROC Curve of SVM Using wavelet features at SNR of 10dB.

Table 6. Overview of related work on AMC, highlighting differences in datasets and methods

Study	Dataset	Modulation Pool	Feature Extraction	Classifier	Best Detection Accuracy	Remarks
Solanki et al. [11]	RadioML.2016.10A	10 modulation schemes	Temporal features	CNN-Long short-term memory · Transfer learning	≈78%@ 10dB	Efficient architecture, slightly lower peak accuracy
Zuo et al [22]	RML2016.10A	11 modulation schemes	STFT	DENSE CNN	≈80.0%@ 10 dB	complex multilevel network
To Troong An et al [21]	RML2016.10A	11 modulation schemes	Threshold denoise	CNN	≈91.0%@ 10 dB	Computationally dense
Present Work	Synthetic/ GNU Radio–generated dataset	BPSK, QPSK, 16-QAM	Wavelet, STFT	SVM, RF, Naive Bayes	97.0@10dB (Wavelet+SVM)	High accuracy with reduced computational complexity

Among all the examined classifiers, Random Forest still competes with SVM but takes the second place because of the latter's optimization based on a margin, which leads to superior generalization and robustness. Also, the capture of non-stationary signal characteristics using wavelet-based features is a further contribution to the increased classification performance. The proposed framework gives a clear and interpretable view of modulation classification as per Table 5. The proposed method is unlike end-to-end deep learning techniques such as Convolutional Neural Networks (CNNs) or Convolutional Long Short-Term Deep Neural Networks (CLDNNs), operating via classical signal processing with STFT and wavelet-based feature extraction, then applying the common machine learning classifiers-SVM, Random Forest, and Naive Bayes. A synthetic dataset produced with the help of GNU Radio not only provides practical advantages but also allows for controlled experimentation and the ability to reproduce the results.

The proposed framework registers the highest classification accuracy of 97.0% for an SNR of 10 dB, and it has low computational demand, thereby being applicable for real-time. Table 6 provides a comparison of the proposed work with other related work on AMC, with details of the dataset along with their performance and limitations. A major contribution of this study is the provision of reproducible benchmarks on a custom dataset, distinct from the existing studies that lean on very advanced and time-consuming deep learning architectures.

7. Conclusion

The research offers an elaborate comparative study of an automatic modulation classification system that employs time-frequency feature extraction and classical machine learning classifiers. The findings of the study unveil a crucial balance between the two main aspects, namely, classification accuracy and computational efficiency. The highest accuracy is 97.0% ± 0.5% at an SNR of 10 dB is obtained using wavelet-based features in combination with the SVM classifier, thus this particular arrangement is deemed appropriate when accuracy is the main goal. On the other hand, the Naive Bayes classifier results in the least computational expense, being only around 0.01s for the whole training; hence, it is perfectly fit for applications that are low in resource but high in demand for real-time processing.

All the analyses performed for different Machine Learning Classifiers(MLCs), feature extraction techniques, and SNR conditions confirmed the trade-off, whereby for all MLCs, the good performance dropped below 35% at an SNR level of -10 dB, posing a challenge in low-noise level environments. The study establishes that a conventional yet computationally inexpensive solution to the problem of automatic modulation classifications, thus providing a valuable and reproducible benchmark for communication systems working under resource constraints. Furtherly, this study brings attention to various directions for future work. Though our selected techniques are in fact effective, incorporating noise-resistant feature extraction techniques, for instance, cyclostationary properties, and broadening the scope to embrace a wider range of modulation schemes like 64-QAM or OFDM might improve the results.

The study is constrained to synthetic signals produced under conditions of AWGN (Additive White Gaussian Noise), and three modulation techniques are examined. The degradation of performance at very low signal-to-noise ratios (−10 dB) indicates the necessity for extracting features that can withstand noise better. Moreover, the impact of real-life channels like fading and interference was neglected.

Limitations of the Study

The current research focuses on testing three separate modulation methods using synthetic signals that researchers have created under AWGN channel conditions. The study does not account for real-world interruptions, which include multipath fading and Doppler effects, together with hardware distortions. The dataset size of the study remains moderate in comparison to large public benchmarks.

Future Work

The future work should also emphasize validating the model using a variety of modulation schemes against real-world datasets like RadioML 2016.10A or implementing it in real-time on hardware platforms such as Software-Defined Radios (SDRs). Such steps will also verify the operational feasibility of the framework amid true wireless environments and pave the way further for its deployment in cognitive radio and next-generation wireless communication systems.

References

[1] Min Wu et al., "Automatic Modulation Classification based on Wavelet Analysis and Convolution Neural Network," *Electronics*, vol. 14, no. 19, pp. 1-16, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[2] Fan Wang et al., "Automatic Modulation Classification using Hybrid Data Augmentation and Lightweight Neural Network," *Sensors*, vol. 23, no. 9, pp. 1-16, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Anwr Hasan Yahya Hasan Abohadi et al., "Automatic Modulation Classification using Fractional S-Transform and Semi-Supervised Learning with Confidence-Guided Consistency Regularization," *Knowledge-based Systems*, vol. 336, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[4] Asif Shakil Ahamed et al., "Zero-Shot Short Time Fourier Transform Radar and Communication Signal Classification," *Discover Electronics*, vol. 2, no. 1, pp. 1-17, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[5] Wenwen Li et al., "Wavelet Transform based Modulation Classification for 5G and UAV Communication in Multipath Fading Channel," *Physical Communication*, vol. 34, pp. 272-282, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] Longmei Zhou, Zhuo Sun, and Wenbo Wang, "Learning to Short-Time Fourier Transform in Spectrum Sensing," *Physical Communication*, vol. 25, pp. 420-425, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[7] Qinghe Zheng et al., "Recent Advances in Automatic Modulation Classification Technology: Methods, Results, and Prospects," *International Journal of Intelligent Systems*, vol. 2025, no. 1, pp. 1-28, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[8] Haithem Ben Chikha, Alaa Alaerjan, and Randa Jabeur, "Deep Learning for Enhancing Automatic Classification of M-PSK and M-QAM Waveform Signals Dedicated to Single-Relay Cooperative MIMO 5G Systems," *Scientific Reports*, vol. 15, no. 1, pp. 1-11, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[9] Peng Wu et al., "Automatic Modulation Classification based on Deep Learning for Software-Defined Radio," *Mathematical Problems in Engineering*, vol. 2020, no. 1, pp. 1-13, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] Luke Boegner et al., "Large Scale Radio Frequency Signal Classification," *arXiv preprint*, pp. 1-29, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Surendra Solanki, Banalaxmi Brahma, and Yadvendra Pratap Singh, "Automatic Modulation Classification for CR using Deep Learning," *SN Computer Science*, vol. 5, no. 8, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Ramiro Utrilla, "MIGOU-MOD: A Dataset of Modulated Radio Signals Acquired with MIGOU, A Low-Power IoT Experimental Platform," *Mendeley Data*, vol. 1, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] Fuxin Zhang et al., "Deep Learning based Automatic Modulation Recognition: Models, Datasets, and Challenges," *Digital Signal Processing*, vol. 129, pp. 1-19, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Hamza Ouamna et al., "Deep Learning-Assisted Automatic Modulation Classification using Spectrograms," *Engineering, Technology and Applied Science Research*, vol. 15, no. 1, pp. 19925-19932, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[15] Faheem Ur Rehman, Qamar Abbas, and M. Karam Shehzad, "DL-AMC: Deep Learning for Automatic Modulation Classification," *arXiv preprint*, pp. 1-5, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[16] To Truong An et al., "Efficient Automatic Modulation Classification for Next-Generation Wireless Networks," *IEEE Transactions on Green Communications and Networking*, vol. 10, pp. 249-259, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[17] Xiaoqian Qin et al., "Multilevel Adaptive Wavelet Decomposition Network-based Automatic Modulation Recognition: Exploiting Time-Frequency Multiscale Correlations," *IEEE Transactions on Cognitive Communications and Networking*, vol. 11, no. 5, pp. 3218-3231, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[18] Soyabul Islam, and Mirza Mohd Shahriar Maswood, "Designing Novel Deep Learning based Frameworks for Performance Improvement of Automatic Modulation Classification," *Engineering Applications of Artificial Intelligence*, vol. 152, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[19] Noureddine El-Haryqy et al., "Improved Automatic Modulation Recognition using Deep Learning with Additive Attention," *Results in Engineering*, vol. 26, pp. 1-14, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[20] Bachir Jdid et al., "Machine Learning based Automatic Modulation Recognition for Wireless Communications: A Comprehensive Survey," *IEEE Access*, vol. 9, pp. 57851-57873, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] To Truong An, and Byung Moo Lee, “Robust Automatic Modulation Classification in Low Signal to Noise Ratio,” *IEEE Access*, vol. 11, pp. 7860-7872, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Xiaoya Zuo et al., “An Automatic Modulation Recognition Algorithm based on Time-Frequency Features and Deep Learning with Fading Channels,” *Remote Sensing*, vol. 16, no. 23, pp. 1-20, 2024. [CrossRef] [Google Scholar] [Publisher Link]