Quality and Reliability Engineering International

WILEY

RESEARCH ARTICLE

An Efficient Automatic Modulation Classification Approach Using Improved Migration Algorithm-Based Ensemble Machine Learning Network

P. G. Varna Kumar Reddy | M. Meena

Department of Electronics and Communication Engineering, Vels Institute of Science, Technology & Advanced Studies, Chennai, Tamil Nadu, India

Correspondence: P. G. Varna Kumar Reddy (varunkumarreddypg.phd1@gmail.com)

Received: 5 March 2025 | **Revised:** 26 November 2025 | **Accepted:** 2 January 2026

Keywords: automatic modulation classification | fuzzy ranking | fitness-based update in migration algorithm | optimal features | optimized ensemble network | performance metrics

ABSTRACT

In the realm of intelligent radio communications, Automatic Modulation Classification (AMC) plays an important role in various applications such as identifying transmitters, allocating spectrum resources and facilitating industrial automation. Both in contemporary military engagements and in civilian electromagnetic regulation, AMC is pivotal for ensuring effective communication signal management on the Internet. It provides the scientific foundation and confidence needed for smart signal reception and processing. Typically, AMC involves extracting and utilizing certain characteristics of the received signal for classification purposes. The efficiency of AMC can be greatly improved by selecting the appropriate features. However, in non-cooperative communication scenarios, the presence of noise in the received signal can pose challenges for existing AMC algorithms, making it difficult to achieve a balance among classification accuracy and model complexity. To solve the problem, a novel Optimized Ensemble Network (OENet) is proposed in this paper. Initially, the required data is gathered from the standard data resource. After that, the features from the input data are optimally selected utilizing the Fitness-based Update in Migration Algorithm (FUMA). Subsequently, the resultant optimal feature is subjected to the classification phase. Here, the OENet is used to classify the class with the help of Naive Bayes (NB), Multilayer Perception (MLP), Capsule Network (CapsuleNet) and Adaboost. These classification outcomes are further averaged by adopting the Fuzzy ranking method. The parameters from the OENet model are optimally tuned using improved FUMA optimization to enhance the performance. Through the simulation experiment, it is determined that the performance of the designed OENet model surpasses that of conventional models. The results obtained indicate superior performance of the OENet model contrasted to existing approaches.

1 | Introduction

Effective communication systems rely on the principle that a message can only be fully understood if both the sender and receiver are privy to its communication parameters, such as the type and features of the selected modulation method, tailored to the conditions of the communication medium [1]. As wireless communication technology progresses, the electromagnetic environment becomes highly dynamic and complex [2]. This evolution leads to a significant increase in the data volume within modulated messages and an expansion in the variety and intricacy of modulation techniques [3]. In situations where specific details such as channel state information and transmitter characteristics are not pre-established, Automatic Modulation Classification (AMC) comes into play to determine the system's modulation schemes without prior knowledge [4]. This marks a key development in the field of intelligent signal processing, finding extensive application across various sectors in recent times [5]. AMC, when integrated with intelligent receivers, can enhance transmission efficiency by reducing the overhead of additional protocols [6]. Research in AMC technologies has given rise to numerous algorithms, which can generally be categorized into two main types: Feature-Based (FB) statistical pattern recognition techniques and Likelihood-Based (LB) decision-theoretic methods. Over the past decade, the FB approach has gained preference among researchers due to its low computational demands and consistent performance [7].

The deployment of Wireless Sensor-Actuator Networks (WSANs) in factory automation has led to the development of intelligent wireless receivers using AMC [8]. These advanced receivers are adept at navigating the challenges of continuous operation and the dynamic conditions typical of wireless environments [9]. The effectiveness of AMC, particularly the FB approach, hinges on the accurate differentiation of signal characteristics, which is crucial for achieving precise modulation classification. Traditional methods of modulation classification, which often rely on manually gathered expert features, struggle to categorize the vast array of emerging communication signals [10]. As a result, their classification accuracy frequently falls short of expectations. In both military and civilian contexts, there is a significant demand for high-precision, data-driven modulator classification techniques [11]. These techniques are crucial for ensuring reliable communication under various conditions. Despite this need, accurately detecting signals at low Signal-to-Noise Ratios (SNRs) remains a challenging task [12].

Deep learning, standing out as the most advanced among machine learning algorithms, has found widespread application in AMC due to its superiority in automating feature extraction and learning over traditional methods that rely on manually engineered features [13]. In the realm of Automatic Modulation Recognition (AMR), feature extraction serves as a pivotal preprocessing step, laying the groundwork for effective classification [14]. Deep learning approaches, leveraging their innate ability to autonomously identify features, are increasingly combined with traditional machine learning and various classifier strategies to enhance AMR performance. To ensure accurate and reliable AMR outcomes, there's a continuous reliance on both machine learning and optimization techniques [15]. Convolution Neural Networks (CNNs), known for their robust capability in extracting spatial features, are especially favored for analyzing image-like data representations within the context of AMR [16]. An advanced AMR framework typically employs a Deep Convolution Neural Network (DCNN) paired with a Long Short-Term Memory (LSTM) network. This combination effectively captures both the spatial and temporal features of modulation signals, significantly improving the system's ability to recognize and classify different modulation types, even in challenging conditions [17].

The proposed AMC approach includes some key contributions, detailed as follows.

1. To design the AMC framework by combining cutting-edge optimization methods and deep learning technologies. This framework is designed to enable smooth communication between devices.
2. To optimally select features from the input data using the recommended Fitness-based Update in Migration Algorithm (FUMA). This mechanism aims to improve the accuracy of the outcomes while enhancing the performance of the network.
3. To develop the FUMA model, which emulates the performance of the existing MA to facilitate the optimal feature selection and improve the classification model within the system. This helps to maximize the relief score, accuracy and precision, thereby enhancing its overall performance.
4. To utilize the Optimized Ensemble Network (OENet) for classifying the modulation, this involves the mechanisms of Naive Bayes (NB), Multilayer Perception (MLP), Capsule Network (CapsuleNet) and Adaboost classifiers, and further refining the classification results using a fuzzy ranking method. The parameters of OENet are optimally tuned using the recommended FUMA approach.
5. To conduct comprehensive simulation experiments and evaluate the performance of the OENet model. This includes comparing its effectiveness in terms of accuracy, precision, recall and other relevant metrics against traditional models to establish its superiority in solving the target problem.

The proposed AMC architecture is structured into seven main sections. Section 2 details the conventional operations involved in modulation classification. Section 3 outlines the various approaches to AMC and the methodologies for data collection. In Section 4, the FUMA strategy is discussed to identify the optimal feature selection for modulation classification. Section 5 presents the results of modulation classification achieved through an enhanced ensemble network coupled with fuzzy ranking. Section 6 gives the results and provides a detailed analysis. Finally, Section 7 concludes the AMC process as proposed.

2 | Literature Survey

2.1 | Related Works

A brand-new technique for spectrum estimation called the Dyadic Aggregated Autoregressive Model (DASAR) was developed by Pinto et al. [18] in 2020. This model described how a modulated message's spectrum moved. When unfamiliar modulation methods were present and only size-restricted data were available to train algorithms for classification, the DASAR improved AMC. It became essential to develop useful knowledge-descriptive features to achieve effective automated categorization. By dividing a signal into consecutive dyadic sections, each represented as an aggregate of single-frequency autoregressive procedures, DASAR developed a multi-level spectral description. Thus, the multi-level breakdown was able to capture time-dependent spectra, and the mathematical framework guaranteed a robust representation at each segment level. As a feature extraction approach, DASAR had the potential to yield valuable learning features for signals with intricate spectra.

Zhang et al. [19] have proposed a tree-shaped multi-layer smooth support vector machine classifier based on a feature selection algorithm in 2020 to identify eleven different types of digital modulating classes. The method utilized a hybrid identification approach based on two novel features alongside additional traditional features. The outcomes of the experiment demonstrated that the algorithm was capable of classifying modulation signals with a low SNR.

An attention-based single-layer LSTM paradigm was proposed by Chen et al. [20] in 2020. The signal was embedded in the first section of this framework, allowing the input to accurately and fully include modulating data. The attention modules then received the hidden state output from the LSTM, which was weighted to assist the LSTM framework in capturing the temporal features of modulated inputs. The aforementioned model performed better in categorization and converged more quickly than a model. The last technique suggested was called RE-TTA, or random erasing-based time for testing augmentation. To increase the precision of the categorization, the test data were randomly erased multiple times, and the outcomes of the categorization were thoroughly assessed.

In 2020, a novel Activation Maximization (AM)-based filter-level pruning method was proposed by Lin et al. [21], targeting the removal of the least significant convolution filters. In the RadioML2016.10a dataset, the CNN that underwent pruning with the AM approach achieved classification accuracy equal to or surpassing that of other network pruning methods.

Ansari et al. [22] have implemented a model in 2022 and compared it using Decision Tree (DT), Radial Basis Function (RBF), MLP, Adaptive Neuro-Fuzzy Inference System (ANFIS) and NB methods. Furthermore, each model's optimal parameters were determined through the application of Genetic Algorithm (GA). In this work, numerous experiments were conducted to ascertain the effectiveness of various methods in detecting modulated signals with widely used digital variations.

In 2021, a Deep Cascading Network Architecture (DCNA) was proposed by Weng et al. [23]. The DCNA comprised an SNR Estimation Network (SEN) and a Modulation Recognition Cluster Network (MRCN). The SEN was tasked with determining the SNR levels for data, while the MRCN, composed of multiple networks of sub-networks, aimed to enhance modulation identification across various SNR settings. Additionally, a label-smoothing technique was introduced to facilitate the integration of SEN and MRCN. To cater to the diverse data requirements of the DCNA, an extra data-segmenting technique was also incorporated. It's important to note that the DCNA did not depend on a specific network architecture and could be expanded to incorporate a variety of deep learning algorithms with further advancements.

Liu et al. [24] have analyzed low-SNR conditions and developed the radio's fuze AMR technique in 2023. To minimize the background noise of the received radio fuze Intermediate Frequency (IF) message, an adaptive denoising technique based on data rearranging and the Two-Dimensional (2D) Fast Fourier Transform (FFT) was initially employed. Support Vector Machines (SVMs) were then used for categorization after the textural features of the denoised IF signal rearrangement data matrix had been recovered from the statistics signal vector of Grey-Level Cooccurrence Matrices (GLCMs).

A unique Convolution Adaptive Noise Reduction (CANR) network was proposed by Bai et al. [25] in 2023, consisting of two distinct modules: Adaptive Noise Reduction (ANR) and Convolution Feature Extraction (CFE). The spatiotemporal traits within the time period were captured by the ANR and CFE modules after denoising the combined input. According to simulations conducted on benchmark datasets, the suggested network required the fewest parameters for training and achieved the highest recognition precision under comparable conditions.

2.2 | Problem Statement

AMC is significant for identifying and demodulating a telecommunication signal. AMC achieved specific attention towards the various applications. AMC is introduced in the electronic welfare system as a source of data for recognizing and disrupting threats. Other civilian applications like frequency management, administration of traffic networks, software-defined radio and monitoring transmitter. The sudden spread and progress of high-speed wireless communication networks instigate the role of AMC in spectrum management and allow the regulatory organization to identify the misuse of spectrum ranges by improper communication channels. Various studies have been developed to investigate different features and recognize the type of modulation, but still, there are some limitations in their performance.

1. Generally, various features of received signals are extracted and utilized for AMC. The unnecessary features in the received signal lead to inaccurate classification. Selecting the appropriate feature plays an important role in improving the performance of AMC. Therefore, the developed approach uses an optimal feature selection model to extract the relevant features for classification.
2. The existing method for AMC poses difficulties in deploying on resource-constrained devices. Therefore, the suggested approach implements the adaptive ensemble network for attaining a high signal to noise ratio.
3. Moreover, in the electromagnetic environment, the SNR is generally low, which causes complexity in accurate classification. To solve this complexity, an adaptive ensemble and fuzzy ranking are introduced, which maximizes the classification accuracy.
4. One of the crucial tasks for the existing AMC model is selecting the appropriate features. Implementing the system with suitable features helps to recognize and separate the multiple modulations. Hence, the designed model implements the optimal feature selection model, which enhances the classification performance and operating speed of the model.
5. The traditional methods still have some difficulties in AMC, as they cannot use the large quantity of communication data. Therefore, the proposed model utilizes the adaptive ensemble network to handle a huge dataset and obtain the result with higher accuracy.

Table 1 gives the features and drawbacks of the conventional AMC models.

TABLE 1 | Features and challenges of the conventional deep learning-based AMC models.

Author [citation]	Methodology	Features	Challenges
Ansari et al. [22]	GA	• It increases the efficiency of the AMC. • It helps to tune the parameters for improving the network performance.	• It requires high computational resources.
Pinto et al. [18]	DASAR	• It gives the necessary learning features related to the signal. • It captures the dynamic spectral signal while retaining the significant spectral component.	• It does not analyze the local conditional probabilities.
Zhang et al. [19]	FS_DT-SSVM	It recognizes a greater number of digital modulation signals. It requires a minimum SNR.	It is not suitable for huge data signals.
Chen et al. [20]	LSTM	• It analyses the temporal feature of the modulated signals • It has a high convergence speed.	• It is not suitable for small sequence data processing.
Lin et al. [21]	Neural Network Pruning	• It increases the accuracy after pruning. • It removes unnecessary data.	• It is computationally more expensive.
Liu et al. [24]	Adaptive denoising	• It helps to minimize the noise of the intercepted frequency signal. • It attains the maximum correlation coefficient.	• It is hard to obtain all the features of the image.
Bai et al. [25]	CANR	• It attains the highest recognition accuracy. • It obtains more dimensional features.	• It takes more time and requires expertise.
Weng et al. [23]	MRCN	• It aggregates the modulation recognition. • It helps to recognize the signal's modulation type.	• It requires a higher amplitude frequency.

3 | Elucidation of AMC: Various Types and Data Collection

3.1 | Divergent Modulation Types Used in the System

AMC is a method employed in communication systems to autonomously determine the type of modulation of input signals without pre-existing data [23]. Various modulation techniques such as 64-Quadrature Amplitude Modulation (64QAM), 16-Quadrature Amplitude Modulation (16QAM), Quadrature Phase Shift Keying (QPSK), 8-Phase Shift Keying (8PSK), and Binary Phase Shift Keying (BPSK) often appear in digital communications.

BPSK: BPSK is a digital modulation scheme, where binary data is modulated onto a carrier signal by shifting its phase by 180 degrees (π radians) for different symbols. It uses two phases, typically 0 and π radians, to represent binary 0 and 1, respectively.

QPSK: QPSK is a digital modulation scheme that uses four different phases of a carrier signal (0, $\pi/2$, π, and $3\pi/2$ radians) to represent two bits per symbol. It divides the symbol time into four intervals, with each interval representing a unique combination of phase shifts.

8PSK: 8PSK is a digital modulation strategy that employs eight different phases of a carrier signal to encode three bits per symbol. It divides the symbol time into eight intervals, with each interval representing a distinct phase shift.

16QAM: 16QAM is a digital modulation strategy that encodes four bits per symbol by varying both the phase and amplitude of the carrier signal. It uses 16 different combinations of amplitude and phase shifts.

64QAM: 64QAM is a digital modulation scheme that encodes six bits per symbol by modulating both the amplitude and phase of the carrier signal. It employs 64 different combinations of amplitude and phase shifts.

In AMC, the task typically involves to analyze the received signal's features, such as its amplitude, phase and frequency components to determine which modulation type is used to transmit the signal.

3.2 | Elaborative View of Proposed AMC Model

AMC is a process used in communication systems to automatically identify the modulation technique of received signals. This approach is crucial for applications across wireless communications, signal intelligence and cognitive radio. AMC aims to analyze the input signals to accurately identify their modulation

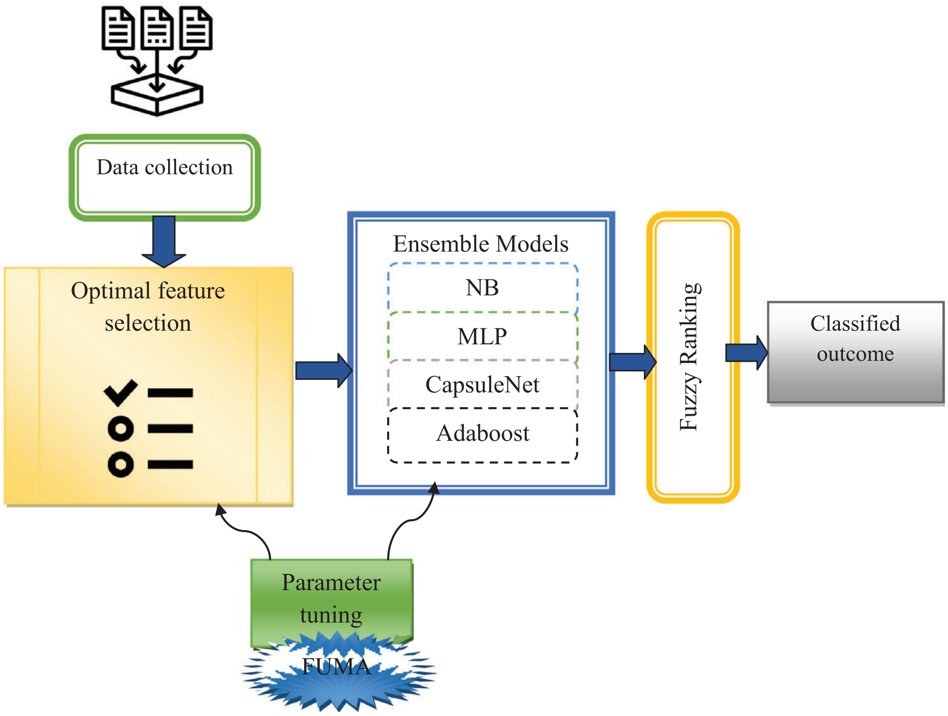

FIGURE 1 | Proposed view of the designed deep learning-based AMC model.

types, such as BPSK, QPSK, 16QAM and 64QAM, without prior knowledge about these signals. AMC equips communication systems with capability to adapt to change the signal conditions and modulation strategies, improving flexibility and operational efficiency in variable environments. However, AMC algoriths can be computationally intensive, demanding significant processing power and resources that may not be readily available on all devices, especially those with limited processing capabilities. The accuracy of AMC can also be compromised by variable signal conditions, including fading, shadowing and interference, leading to potential misclassification of modulation types. Additionally, relying heavily on AMC for critical communications can introduce security vulnerabilities, as adversaries might employ sophisticated jamming or spoofing techniques to mislead the classification system. The recommended approach aims to mitigate these challenges, thereby bolstering system performance and reliability. Figure 1 shows the proposed view of the designed deep learning based AMC model.

In the rapid evolving landscape of communication systems, the efficient management and classification of modulation schemes play a crucial role in ensuring seamless communication among devices. To address this, a novel framework named AMC is proposed, leveraging cutting-edge optimization methods and deep learning technologies. AMC aims to improve the accuracy and performance of modulation classification while enabling smooth communication across diverse devices. The FUMA algorithm is employed to optimally select features from the dataset. FUMA improves the accuracy of outcomes and network performance by efficiently identifying relevant features. This process facilitates the subsequent stages of modulation classification. The modulation classification with OENet involves integrating outcomes from various classifiers like NB, MLP, CapsuleNet and Adaboost. These classifiers are further refined using a fuzzy ranking method to improve classification accuracy. OENet provides a robust framework for accurately identifying modulation schemes. The OENet's parameters are optimally tuned using the proposed FUMA approach. Finally, comprehensive simulation experiments are conducted to determine the performance of the OENet. The effectiveness of OENet is compared with traditional approaches in terms of precision, accuracy, recall, and other relevant metrics. These evaluations aim to prove the superiority of OENet in effectively solving the targeted modulation classification problem.

TABLE 2 | Modulation types in the DeepSig Dataset.

Modulation types	Modulations
Digital	QPSK, CPFSK, QAM16, GFSK, PAM4, QAM64, 8PSK and BPSK
Analog	AM-DSB, WBFM, and AM-SSB

3.3 | Data Acquisition

Dataset (DeepSig Dataset: RadioML 2016.10A): For the operational AMC system, data is collected through https://www.kaggle.com/datasets/nolasthitnotomorrow/radioml2016-deepsigcom. This synthetic dataset encompasses 11 modulation types, which are given in Table 2. The data for this collection can be produced using GNU Radio by varying the SNR levels across -10 dB, -5 dB, 0 dB, 5 dB, 10 dB, 15 dB, and 20 dB.

The acquired data is mentioned as $W_s, \quad where \ \ s = 1, 2, 3, \ldots, S$, and the amount of gathered data is specified as S.

4 | Optimal Feature Selection With Fitness-Based Update in Migration Algorithm (MA) for Classifying the Modulation

4.1 | Fitness-Based Update in MA

The FUMA strategy enhances the foundational principles of the MA as detailed in [26] by fine-tuning classifier parameters and feature selection to increase the accuracy of the classification model. MA's versatility in addressing a broad spectrum of optimization challenges makes it an invaluable resource for researchers and practitioners across diverse sectors. It aims to mimic the natural migration patterns of animals to comprehensively explore the search space, thereby improving its capability to avoid local optima and seek out global solutions, essential for resolving complex optimization problems. However, the performance of MA can be significantly influenced by its parameter settings. Identifying the optimal configuration often requires an extensive experimentation, which can be quite labor-intensive. Compared to more developed optimization techniques, there is a relative lack of theoretical insight into the convergence behaviors and properties of MA, complicating predictions about its effectiveness on novel problem types. In response to these issues, the FUMA approach was developed to improve the algorithm's efficiency. This method is formally described by Equation (1).

$$R\eta = \frac{\frac{B_{fit}^{1.5}}{W_{fit}}}{N_p} \tag{1}$$

In the proposed structure of the FUMA, a random number is denoted by $R\eta$. The variables associated with the highest and lowest fitness values are represented by B_{fit} and W_{fit}, respectively; furthermore, N_p signifies the size of the population. The adaptation of the algorithm is shown in Equation (6). Subsequent to this framework, the mathematical foundation of the FUMA approach is elaborated as below.

Initialization: By leveraging the search capacity of the population in the task-based repetition, the classical MA can tackle optimization problems. In the classical MA, each member of the population is viewed as part of a solution to the problem, constructed using a vector. Equation (2) presents the matrix of the population, while Equation (3) discusses the initialization of the initial position within the population's structure.

$$S = \begin{bmatrix} S_1 \\ \vdots \\ S_i \\ \vdots \\ S_N \end{bmatrix}_{N\times m} = \begin{bmatrix} S_{1,1} & \cdots & 1S_{1,j} & \cdots & S_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ S_{i,1} & \cdots & S_{i,j} & \cdots & S_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ S_{N,1} & \cdots & S_{N,j} & \cdots & S_{N,m} \end{bmatrix}_{N\times m} \tag{2}$$

$$S_i : s_{i,j} = \delta_j + R\eta.\left(\chi_j - \delta_j\right), \; i = 1,2,\ldots,N, \; j = 1,2,\ldots,m \tag{3}$$

Building on the previously discussed concepts, the population matrix S in the traditional MA is also detailed, with the total number of selected variables is displayed as m. Additionally, the term N is utilized, and the overall count of candidates is specified as S_i. The random integer $R\eta$ approach is employed, and the upper χ and lower δ bounds of the decision factor are delineated as well.

For each option within the demographics pertaining to the problem, objective functions are computed. Equation (4) outlines the expected values of the objective function utilizing a vector.

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_i \\ \vdots \\ Y_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} Y(E_1) \\ \vdots \\ Y(E_i) \\ \vdots \\ Y(E_N) \end{bmatrix}_{N\times 1} \tag{4}$$

In this context, the factors Y include the objective function vector, which represents both the objective function and the target function Y_i of the i^{th} component solution.

Exploration phase: Selecting a migration destination from the array of available options is essential. During travel, individuals select their destinations according to their specific conditions and move accordingly. In the traditional MA process, this method is employed in the initial phase of updating the population. The destination of each individual is defined by Equation (5), while the new position of each candidate is calculated using Equation (6), factoring in their movement towards the migration destination. Moreover, Equation (7) determines the position of a given candidate when their objective function is considered.

$$RT_i = \{S_k, Y_k < Y_i \; and \; k \in \{1,2,\ldots,N\}\}, \; where \; i = 1,2,\ldots,N, \tag{5}$$

$$s_{i,j}^{P1} = s_{i,j} + k_{i,j}.\left(UI_{i,j} - R\eta_{i,j}.L_{i,j}\right), \tag{6}$$

$$S_i = \begin{cases} S_i^{P1}, & Y_i^{P1} \le Y_i, \\ S_i, & else, \end{cases} \tag{7}$$

Here, the candidate of destination for the i^{th} population is given as RT_i, and the term UI_i stands as the migration destination for the i^{th} population. The term S_k presents the matrix's of S with k^{th} row. Then, the term $s_{i,j}^{P1}$ is the upgraded term calculated for the i^{th} population candidate. Finally, the random factor is pointed as $R\eta$.

The conventional method operates under the assumption that the random vector falls within the range of 0 to 1. The choice of parameters for this traditional approach plays a crucial role in its performance. Identifying the optimal parameter configuration often requires extensive trial and error; however, even with careful parameter tuning, the MA may experience rapid convergence, particularly if there is insufficient maintenance of diversity among agents. To address these challenges, a new formula is presented, as depicted in the Equation (1).

Exploitation phase: An individual focusing on adjusting to the prevailing conditions upon entering and migrating to a new society and environment embodies the exploitation phase

Algorithm 1 | Developed FUMA.

```
Begin
Initialize the constraints and variables
Execution attributes
Initialize the population matrix and objective function
While j > J_max
  For i = 1  to  N_p
  Upgrade the random value using Equation (1)
    Generate a new random integer using Equation (2)
    Determine the member destination for migration operations using Equation (5)
    Select the migration destination
    Update the new position for the exploration phase using Equation (6) and Equation (7)
    Update the new position for the exploitation phase using Equation (8) and Equation (9)
    Save best solution
  End for
End while
Return optimal solution
```

in the traditional MA. Subsequently, a random location adjacent to the same candidates is generated utilizing Equation (8). If this designed location is evaluated using the level of the objective function, the position of the prospective candidate is altered in accordance with Equation (9).

$$s_{i,j}^{P2} = s_{i,j} + (1 - 2R\eta)\,.\,\frac{\chi_j - \delta_j}{D} \tag{8}$$

$$S_i = \begin{cases} S_i^{P2}, & Y_i^{P2} \le Y_i, \\ S_i, & else, \end{cases} \tag{9}$$

In this section, the new position of the i^{th} population member in j^{th} dimension is denoted as $s_{i,j}^{P2}$, and the current location predicted for each i^{th} candidate according to the second phase is presented as s_i^{P2}. The corresponding objective function of the second phase is denoted as Y_i^{P2}, and the previous objective function is Y_i. Following this, the current iteration counter is indicated as D. The previous position is specified as $s_{i,j}$. The newly updated random integer is given as $R\eta$. The upper and lower regions are given as $\chi_j \quad and \quad \delta_j$. The i^{th} candidate's old position is taken as S_i.

The initial iteration of the MA concludes with the update of each candidate in the pool, reflecting the initial and final stages of the traditional MA process. Subsequently, the update process is reiterated until the MA reaches completion. With each run, the best candidate within the general population undergoes improvement. Algorithm 1 gives the pseudo-code for the FUMA as introduced. Figure 2 shows the recommended FUMA approach's flow chart.

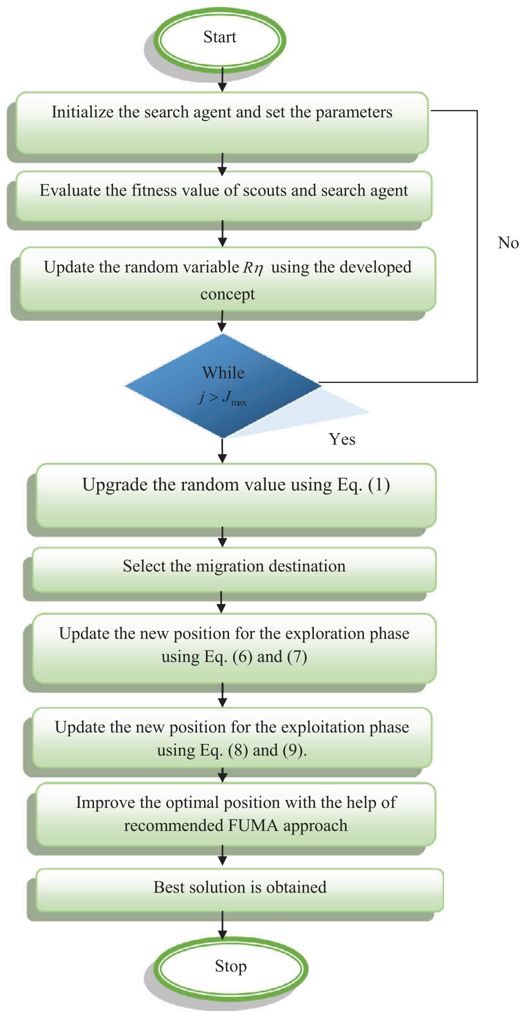

FIGURE 2 | Proposed FUMA approach's flow chart.

4.2 | Optimal Feature Selection

Optimal feature selection involves identifying and selecting the key features from a dataset that significantly influence the predicted feature or outcome of interest. This approach is vital in the development of models because it improves model performance by minimizing overfitting, increasing accuracy and reducing training time. The aim is to select a subset of features that best predict a particular outcome, while eliminating superfluous, irrelevant or noisy data that can diminish the efficiency of the model. In the recommended model, the optimal selection of features from the input data W_s is accomplished using the FUMA

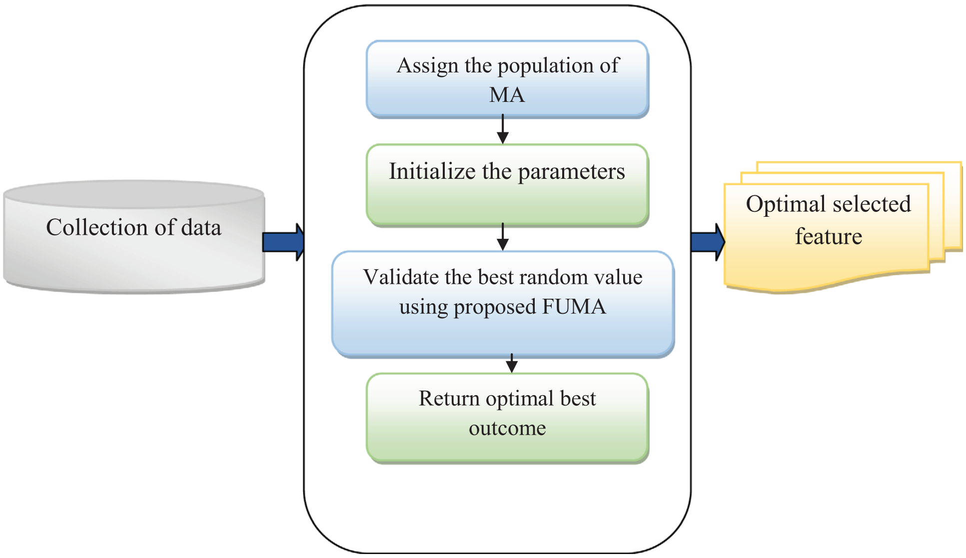

FIGURE 3 | View of optimal feature selection using the recommended algorithm.

approach. Figure 3 shows the view of optimal feature selection using the recommended algorithm.

4.3 | Objective Function

When models incorporate all available features without discerning selection, it may suffer from decreased accuracy due to the inclusion of extraneous or redundant features. This scenario often leads to overfitting, where the model shows high performance on the training dataset but underperforms on new, unseen data. Moreover, models burdened with an excess of features tend to become overly complex. This complexity can lead to increased training time, higher computational demands and obstacles in the evaluation and interpretation of the model's behavior. Navigating through datasets teeming with numerous variables can also cause data overload, complicating the identification of truly impactful features for prediction or classification tasks. To mitigate these issues, an enhancement of feature selection from raw data was undertaken using the FUMA approach. The effectiveness and objective function ob of the proposed system are mathematically articulated in Equation (10).

$$ob = \arg \max_{\{O^{Feature}\}} [RS] \tag{10}$$

The optimal selected feature is indicated as $O^{Feature}$ and selected in the range of [1 - no of features from raw data]. Additionally, the FUMA method is employed to increase its relief score RS. This relief score is an indicator of how effectively the features can distinguish between different classes within the data sources. It evaluates the capacity of an individual feature to differentiate between items of distinct classes and items within the same class, as described in Equation (11).

$$RF = d\left(o_{aa}^{feature} - cs\right) - d\left(o_{aa}^{feature} - cd\right) \tag{11}$$

Here, the difference of the feature is denoted as $d(o_{aa}^{feature})$ and the closest object of the different class is specified as cd. Also, the closest object of the same class is indicated as cs.

5 | OENet and Fuzzy Ranking-Based Outcome for Modulation Classification

5.1 | Networks Used in Ensemble Process

The OENet is a classification framework that leverages multiple base classifiers to make predictions on features extracted from a dataset. In this context, the OENet utilizes a combination of diverse classifiers include NB, MLP, CapsuleNet and Adaboost, to collectively classify the features. These models are deeply explained below.

- **NB**: The NB [27] classifier represents the most fundamental approach to statistical Bayesian learning, earning its designation as 'naive' because it presupposes that each feature independently influences the outcome, despite their mutual correlations. In this model, the selected feature $O^{Feature}$ serves as the input. The application of the NB classification technique, grounded in the Bayesian Theorem, proves especially advantageous for handling input data with high dimensions. At the core of Bayesian classification lies Bayes' theorem, formulated as follows: Consider A as a data sample with an unknown class label, and C represents a hypothesis that A belongs to a specific class B. By applying Bayes' theorem, one can calculate the posterior probability using Equation (12).

$$P(B|A) = \frac{P(A|B).P(B)}{P(A)}.O^{Feature} \tag{12}$$

From the above equation, the term $P(B|A)$ represents the target class probability. The term $P(B)$ is known as the probability of the prior class. The variable $P(A|B)$ refers to the probability of the predictor given the class and $P(A)$ denotes the prior probability of the predictor of the class. Finally, the NB-based classified outcome is obtained.

MLP: The MLP [28] is a type of feed-forward Artificial Neural Network (ANN) consisting of interconnected nodes or neurons that process data to yield a specific output. Within an MLP, the arrangement of neurons is structured into three distinct layers. The first layer consists of input neurons, which are processed with receiving the input data and forward it to subsequent levels. The number of neurons in this layer typically matches the number of input features. In this model, the selected feature $O^{Feature}$ serves as the input. Following this, the hidden layer, which comprises neurons that apply mathematical functions to transform the data, comes into play. The complexity of the problem being addressed dictates the number of hidden layers within an MLP. Finally, the transformed data reaches the output layer, where the final result is produced. The number of neurons in this output layer is determined by the format of desired outcome.

The learning algorithm of the MLP employs the Euclidean norm to minimize the error measure defined over the training set (a_i, f_i) for $i = 1, 2 \ldots N$ and mathematically modelled in Equation (13).

$$P(E) = \frac{1}{2}\sum_{i=1}^{N} ||h(a_i, g) - f_i||^2 O^{Feature} \tag{13}$$

here, the adjustment factor is denoted as g and the total data point is N. The optimized feature is indicated as $O^{Feature}$. Gradient-based methods stand out as some of the most effective strategies for optimization. For medium-sized networks, the Levenberg-Marquardt algorithm proves to be the most efficient, whereas conjugate gradients are favored for larger networks. Generally, gradient algorithms employ a consistent process for incremental weight adjustments step by step. This is modelled in Equation (14).

$$g(p+1) = g(p) + \chi g(p) \tag{14}$$

In the equation mentioned above, the learning rate is represented by χ, the adjustment factor by g and the direction in which reduction occurs during the p^{th} step is denoted by $g(p)$. The method for determining the value of $g(p)$ differs among various learning techniques.

Specifically, the Levenberg-Marquardt method utilizes the least squares form of the learning problem and is shown mathematically as Equation (15).

$$P(E) = -V(p)^{-1} v(p) \tag{15}$$

The conjugate gradient technique, suitable for extensive networks, utilizes the following equations to ascertain the direction $g(p)$ and shown in Equation (16).

$$P(E) = -V(p)^{-1} Y v(p-1) \tag{16}$$

In this context, the Polak–Ribiere formula is commonly employed to compute the conjugate factor Y and shown mathematically as Equation (17).

$$P(p) = \frac{V(p)^{-m} v(p-1)}{v(p-1)^{m} v(p-1)} \tag{17}$$

The learning factor Y in the weight update equation (16) should be adjusted by the user. Typically, an adaptive approach is employed, which accounts for the actual progress of the error function towards reduction. Finally, the MLP-based classified outcome is obtained.

CapsuleNet: The capsule network [29] comprises three distinct hidden layer types: the primary capsule layer, the digit capsule layer, and the convolution layer. In this model, the selected feature $O^{Feature}$ serves as the input. The convolution layer utilizes a 9×9 filter size, with 256 output channels and a stride of 1, facilitated by the ReLU activation function, to extract features from images and create a 20×20 feature map. This feature map is further processed by the primary capsule layer, where scalar feature representations are converted into vectors. This is achieved through 32 sets of 6×6 capsules, each encoding data in 8-dimensional vectors. The output from the digit capsule layer, however, manifests as 16-dimensional vectors for each capsule. The dynamic routing-by-agreement mechanism, starting from the capsules in the primary capsule layer, is employed to identify the relevant capsules, ensuring the network focuses on the hierarchical relationships captured by the capsules.

Transform matrices are denoted as $K_{ij} \in E^{l*m}$ and are applied layer by layer to the lower-level inputs, creating a prediction vector notated as $S_{ij} \in T^{l*m}$ from the input $S_{i|j} \in T^{l*m}$. This process encapsulates crucial spatial relationships between lower-level and higher-level features. The parameter l specifies the number of neurons in the target capsule and is mathematically formulated in Equation (18).

$$\hat{S}_{i|j} = E_{ij} s_i . O^{Feature} \tag{18}$$

The back propagation process focuses on learning the weight matrix E_{ij}. The optimized feature is indicated as $O^{Feature}$. Subsequently, the weighted sum of all gathered prediction vectors is determined utilizing Equation (19).

$$Y_i = \sum I_{ij} s_{ij} \tag{19}$$

where I_{ij} represents the connection coefficient determined by the routing dynamics and s_{ij} denotes the entirety of inputs to the upper capsule and shown in Equation (20).

$$\sum I_{ij} = 1 \tag{20}$$

The output for the higher-level capsule j is generated by applying a non-linear squashing function. This function ensures that

Adaboost: The essence of boosting methods lies in constructing a highly accurate classifier by combining several weak classifiers, which are basic and moderately accurate models. These weak classifiers are trained sequentially, with each one focusing primarily on the examples that proved most challenging for its predecessors. In this model, the selected feature $O^{Feature}$ serves as the input. This approach is exemplified in the context of AMC using the Adaboost [30] algorithm, where a random forest serves as the weak learner. The training data set $K = \{(a_1, b_1)\ldots\ldots(a_m, b_m)\}.O^{Feature}$ serves as the input for the boosting process, where each instance a_i is a feature vector from the domain a, and each label b_i is a class label from the label space b associated with b_i.

Input: $K = \{(a_1, b_1)\ldots\ldots(a_m, b_m)\}.O^{Feature}$ represents a collection of m labelled samples, each with a label from the set b.
Learn through a systematic algorithmic approach.
A constant E

[1] Initialize weights for all i : $E_i^{(1)}$= initialize the weights.

[2] For all i, compute the normalized weight: $R_i^d = \frac{E_i^{(l)}}{\sum E_i^{(l)}}$

[3] T_l ≔ Learn (U, V_l)- call the learning function with normalized weights.

[4] If $\delta_1 > 1/2, then$:

[5] Set $Y = y - 1$

[6] Update the weight adjustment factor $\chi_1 = \frac{Q_i}{1-Q_i}$

[7] For all i compute the new weight

[8] Output of the final hypothesis: $E_{final}(y) = [\sum_{y=1}^{Y} \log(\frac{1}{\chi_1})E_{final}(y)]$.

This algorithm outlines a process for boosting, where weights are initially set and iteratively adjusted to minimize errors in successive learning rounds, ultimately producing a strong classifier from a series of weak ones. Finally Adaboost-based classified outcome is obtained.

longer vectors are scaled down close to unit length K_j and shorter vectors are reduced to near-zero magnitude, all while preserving the direction of the vector and mathematically displayed in Equation (21).

$$O_i = |\frac{|K_j||^2 K_j}{1 + \left\|K_j\right\|^2 \left\|K_j\right\|} \tag{21}$$

The surtax function produces the coupling coefficient I_{ij} as described below, Equation (22).

$$I_{ij} = \exp(t_{ij})//\exp(t_{il}) \tag{22}$$

In other words, the system is expected to better capture hierarchical relationships, an essential aspect in a dataset comprising images with distinct segments. Finally capsuleNet-based classified outcome is obtained (Algorithm 2).

5.2 | Fuzzy Ranking Approach

Fuzzy rating with an ensemble model involves assigning ratings to items or events based on the estimated probabilities or scores generated by multiple classifiers within a collective framework [21]. This approach eschews precise rankings for each item in favor of assigning a degree of belonging to each element for every possible rank, thereby navigating through the ambiguities or uncertainties inherent in ranking processes. By integrating fuzzy logic with ensemble learning, fuzzy classification can adeptly manage complex ranking scenarios characterized by uncertainty or imprecise data, providing outcomes that are both more resilient and interpretable compared to traditional ranking methodologies.

In the proposed model, the optimal features $O^{Feature}$ are fed into an ensemble of four distinct classifier-based outcomes: MLP-based, NB-based, CapsuleNet based and Adaboost-based classified outcomes. These inputs are processed through a fuzzy ranking method to yield the classified output. This approach leverages the strengths of various classification techniques, enhancing accuracy and reliability through the integration of multiple predictive signals. The fuzzy ranking method applies fuzzy logic principles to rank and combine the outcomes from the different classifiers, resulting in a more refined and accurate classification output.

5.3 | Proposed OENet for Classification

The OENet represents a state-of-the-art approach for AMC, a critical task in the realm of digital communications. This innovative framework integrates the power of ensemble learning with optimization techniques to accurately identify modulation schemes from complex signal environments. OENet employs a meticulously selected mix of various classification models such as MLP, NB, CapsuleNet and Adaboost, among others. Each model brings its own distinctive advantages to the task of analyzing signals.

MLPs can overfit when the algorithm is too complicated for the provided dataset. To get ideal efficiency, must carefully tune in terms of features and architecture. The number of hidden neurons in CapsuleNet is determined by the architecture and design decisions. Limitations include a requirement for large processing resources and inadequate research when compared to more established designs, such as CNNs and MLPs. AdaBoost does not use secret neurons like neural networks do. Instead, it focuses on modifying the amount of weight of training cases to highlight

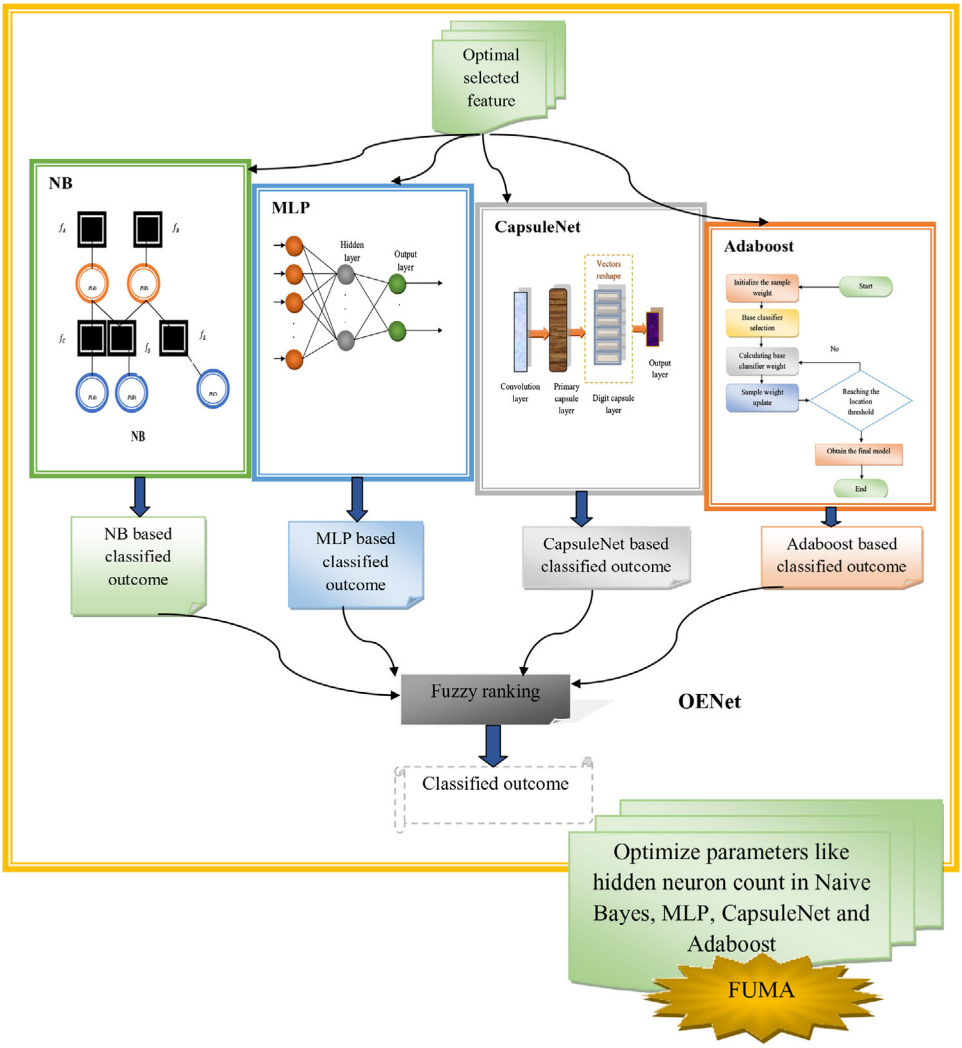

FIGURE 4 | Proposed view of OENet for AMC.

the most difficult-to-classify occurrences. AdaBoost's limitations include sensitivity to dirty data and outliers, plus potential overfitting when the weaker classifiers are too complicated. To reduce such mistakes, this work optimizes the specific parameters like hidden neuron count in NB, MLP, CapsuleNet and Adaboost using the suggested FUMA. The aforementioned optimization tries to increase accuracy and precision. The suggested system's objective function is given in Equation (23).

$$ob = \underset{\{HH_{NB}, HH_{MLP}, HH_{CN}, HH_{AD}\}}{\arg\max} [Ac + pp] \tag{23}$$

From the above equation, the variable $HH_{NB}, HH_{MLP}, HH_{CN}$ and HH_{AD} is the hidden neuron count in NB, MLP, CapsuleNet and Adaboost. Also, the intervals of these models are [5 - 255]. The term Ac and pp defines the accuracy and precision. These metrics are shown in Equations (24) and (25).

Accuracy:

$$Ac = \frac{ff + hh}{ff + hh + bb + dd} \tag{24}$$

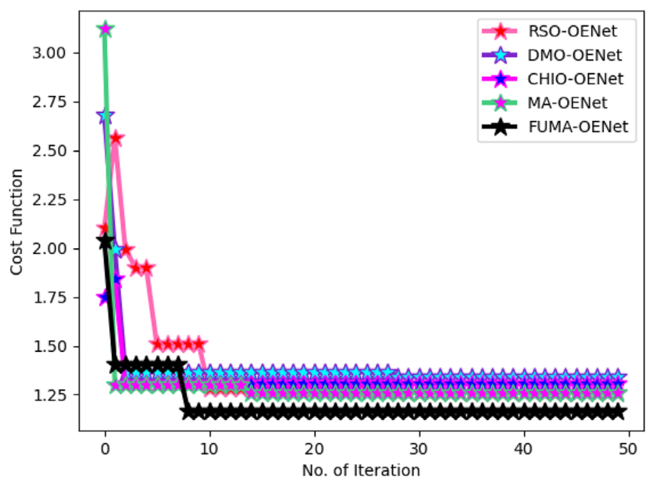

FIGURE 5 | Convergence analysis of the designed FUMA-OENet for AMC over the conventional algorithms.

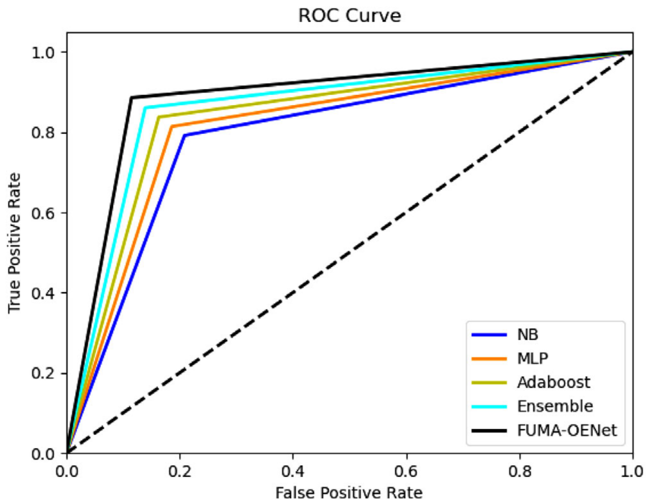

FIGURE 6 | ROC assessment of the recommended FUMA-OENet-based AMC over the traditional classifiers.

Precision:

$$pp = \frac{ff}{ff + hh} \tag{25}$$

here, the "true negative and true positive terms" are noted as ff and hh. The "false negative and false positive terms" are noted as dd and bb.

Role of ensemble learning in improving the proposed method: The performance of the designed OENet approach is highly improved via the adoption of ensemble learning, which integrates the predictive capabilities of distinct classifiers to attain highly reliable and generalized outcomes. In this work, the ensemble combines four distinct learners: NB, MLP, CapsuleNet and Adaboost, each contributing unique decision-making strengths based on their significant learning principles. Though individual classifiers may perform effectively under specific conditions, they are susceptible to variance, bias or overfitting when handling complex modulation environments. The ensemble learning mechanism addresses these problems by combining their outcomes by employing a fuzzy-based ranking approach, which guarantees that the final classification decision demonstrates highly accurate and consistent consensus among all techniques. In addition, the ensemble framework synergizes efficiently with the FUA-aided feature optimization, as the selected high-quality features improve the complementary nature and diversity of the classifiers' responses. This cooperative learning approach results in enhanced stability, robustness and accuracy, allowing the designed FUMA-OENet to perform better than the conventional single classifier and non-optimized ensemble models in the AMC tasks.

Figure 4 shows the proposed view of OENet for AMC.

6 | Results and Discussion

6.1 | Simulation Setup

The suggested modulation classification framework was implemented using the Python platform, where the PyCharm software version 3.11 and Anaconda version 3 were utilized. The libraries such as keras, tflearn, operncv-python, prettytable, tensorflow and numpy were utilized. The population size, chromosome length, and maximum number of iterations of the FUMA were 10, 4, and 50, respectively. A comparative validation was undertaken to assess the efficiency of various methodologies, with the results detailed in the below sections. This evaluation encompassed both contemporary methods, such as NB, MLP, Adaboost and Ensemble network, as well as traditional optimization algorithms, including Rat Swarm Optimizer (RSO)-OENet [31], Dwarf Mongoose Optimization (DMO)-OENet [32], Corona virus Herd Immunity Optimizer (CHIO)-OENet [33], Miration Algorithm (MA)- OENet [26].

6.2 | Performance Metrics

The designed AMC model employs various metrics, and it is given below.

a. **F1-score**: $F1 = \frac{2\times dd}{2bb+ff+hh}$

b. **MCC**:$MCC = \frac{ff\times hh - bb\times dd}{\sqrt{(ff+hh)(bb+dd)(ff+hh)(bb+dd)}}$

c. **FDR**: $fdr = \frac{dd}{ff+dd}$

d. **Hit rate**: $HR = \frac{ff}{bb+dd}$

e. **MAP**: Mean Average Precision (MAP) is a measure employed to evaluate the quality of information retrieval devices and, in some contexts, the performance of object detection models in machine learning. It calculates the Average Precision (AP) for each query or class and then computes the mean of these average precisions, hence the name MAP.

f. **NDCG**: Normalized Discounted Cumulative Gain (NDCG) is a metric employed to estimate the efficiency of information retrieval devices, most importantly in scenarios where the relevance of the retrieved items varies in degrees.

6.3 | Cost Function Evaluation

The cost function of the designed FUMA-OENet system is evaluated in Figure 5. The effectiveness of the proposed approach is gauged by directly comparing it with established standalone techniques. According to the findings, the proposed FUMA-OENet methods significantly decreased cost values at 50 iterations, surpassing the RSO-OENet, DMO-OENet, CHIO-OENet and MA-OENet methods by 22.5%, 12%, 16.66%, and 66.33%. These results underscore the cost-effectiveness of the designed strategy in relation to more traditional approaches.

6.4 | ROC Evaluation

The Receiver Operating Characteristic (ROC) evaluation for the designed FUMA-OENet system is presented in Figure 6, which assesses the system's effectiveness in AMC. The ROC evaluation is carried out as a significant analytical component to validate the classification performance of the designed FUMA-OENet

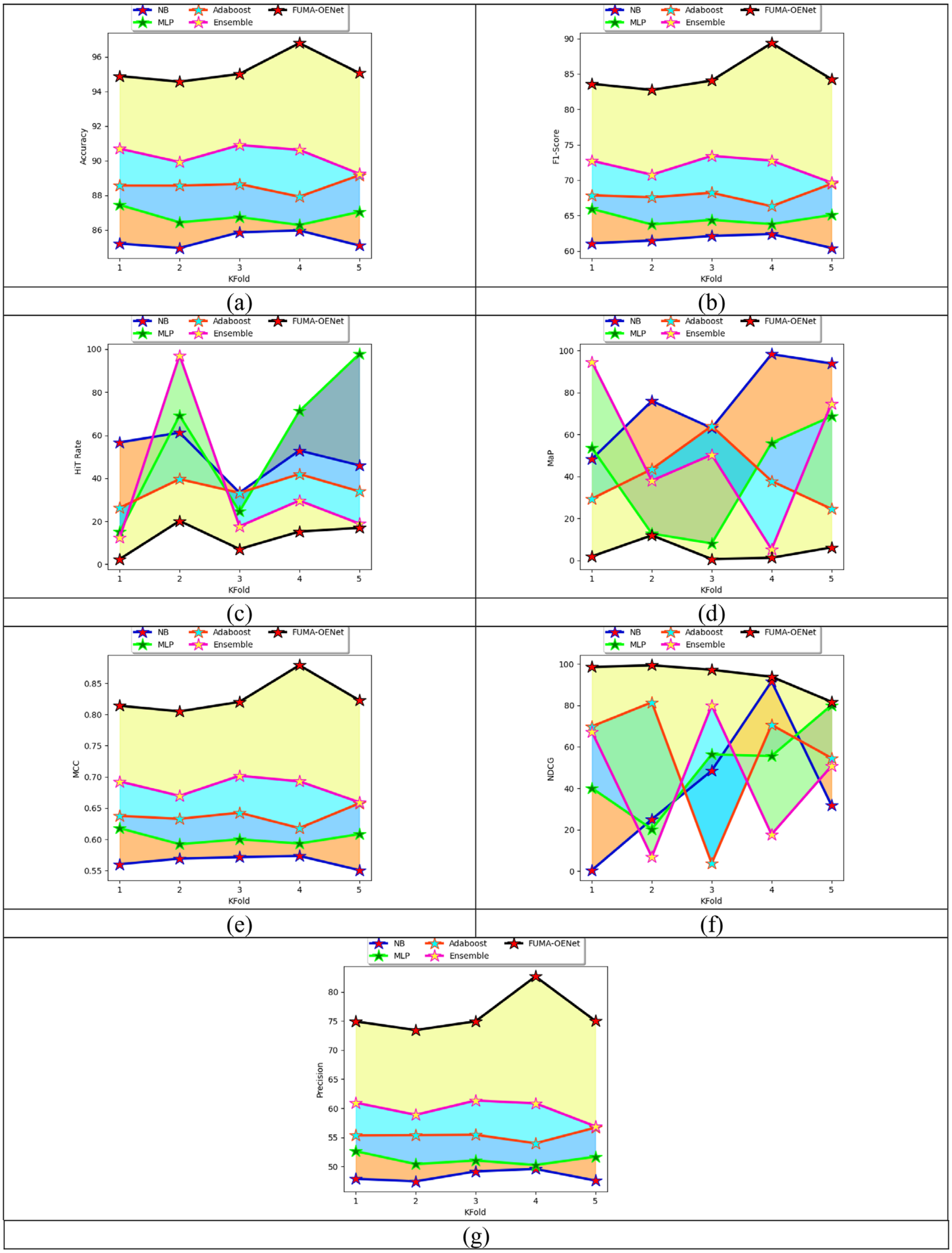

FIGURE 7 | K-fold analysis of the recommended AMC system over the traditional classifier concerning (a) Accuracy, (b) F1-score, (c) Hit rate, (d) MaP, (e) MCC, (f) NDCG and (g) Precision.

approach. The ROC curve demonstrates the relation among the FPR and TPR across distinct classification thresholds, thus providing a detailed view of the technique's discriminative ability. By plotting these parameters, the ROC validation enables the performance of the model to be validated beyond a single decision threshold, specifying how well the technique differentiates among distinct modulation classes. The findings reveal that the proposed FUMA-OENet systems notably reduces the FPR at an FPR threshold of 0.8 compared to the NB, MLP, Adaboost and Ensemble methods by 3%, 23.2%, 44%, and 11%, respectively. This superior ROC performance of the designed FUMA-OENet further estimates the efficiency of the FUMA in optimizing both feature selection and classifier parameters, resulting in a more robust and reliable AMC approach.

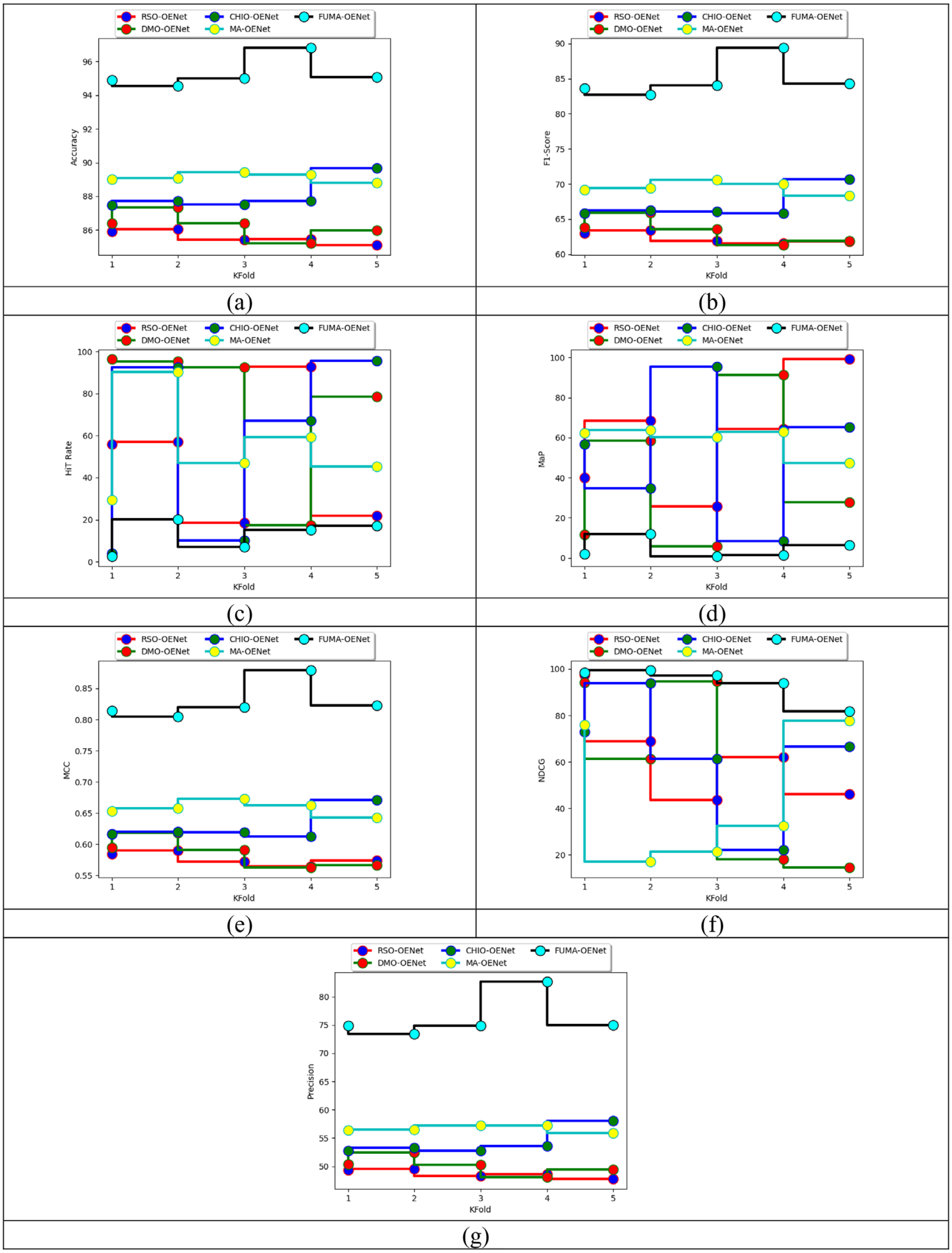

FIGURE 8 | K-fold analysis of the designed AMC system over the traditional algorithm concerning (a) Accuracy, (b) F1-score, (c) Hit rate. (d) MaP, (e) MCC, (f) NDCG and (g) Precision.

6.5 | Comparative Assessment Based on k-Fold Analysis Over Several Classifiers and Algorithms

Figures 7 and 8 provide visual representations of the findings, with Figure 7(a) specifically showcasing the notable accuracy improvements achieved by the proposed method when compared to established models such as NB, MLP, Adaboost and Ensemble. The observed enhancements, quantified as 45.6%, 41.3%, 11.3%, and 65.6% respectively, underscore the superior performance of the designed approach across a range of modulation scenarios. By transcending the limitations of traditional methods and providing a more robust and efficient alternative, the suggested research contributes significantly to the advancement of AMC. The insights gleaned from this study not only deepen our

TABLE 3 | Statistical analysis of the designed algorithm over traditional algorithms.

TERMS	RSO-OENet [31]	DMO-OENet [32]	CHIO-OENet [33]	MA-OENet [26]	FUMA-OENet
Best	1.261148	1.342505	1.303118	1.259573	1.162748
Worst	2.564826	2.678311	1.840045	3.119877	2.041459
Mean	1.375795	1.393947	1.322745	1.30701	1.214242
Median	1.280883	1.365137	1.303118	1.259573	1.162748
Standard deviation	0.260896	0.20421	0.096595	0.259551	0.144947

understanding of modulation classification techniques but also pave the path for improved performance and reliability in live communication systems.

6.6 | Statistical Evaluation Over Algorithms

Recent enhancements in machine learning and signal processing have led to the implementation of numerous AMC algorithms, each strive to improve classification accuracy and efficiency under diverse and challenging conditions. Among these, the FUMA-OENet model has emerged as a promising approach, leveraging innovative techniques to enhance performance metrics significantly. To achieve a comprehensive understanding of the designed FUMA-OENet's performance, the statistical validation is conducted and contrasted over other optimization algorithms such as RSO-OENet, DMO-OENet, CHIO-OENet and MA-OENet. The statistical measures considered include best, worst, mean, standard deviation and median as given in Table 3. These measures offer a comprehensive view of the algorithm's overall efficiency, stability, and reliability across distinct trials. By focusing on the median value, the FUMA-OENet model gains 10.34%, 17.24%, 13.79% and 12.06% over the aforementioned models, respectively. The percentage improvements indicated for the designed FUMA-OENet are calculated based on the median values of each comparative algorithm using the following equation: improvement (%) = ((proposed median value of FUMA-OENet-existing median value of an algorithm/proposed median value of FUMA-OENet)× 100). Applying this formula, ((1.16-1.28)/1.16) × 100 = -10.34 (ignore the negative sign). These calculations guarantee that the designed FUMA-OENet attains substantial improvements across the overall comparative algorithms, emphasizing its superior optimization capability and robustness. The median value is selected as the main performance measure because it gives a robust factor of central tendency that is less sensitive to extreme values or outliers. In optimization-based machine learning approaches, the performance values can vary across iterations because of the stochastic nature of weight initialization and feature selection. As a result, the mean may be influenced by low or high results, whereas the median better reflects the consistent or typical performance of the approach. Hence, concentrating on the median enables a stable and fair comparison between the designed FUMA-OENet and the conventional algorithms. In addition, the smaller values of the best, worst, mean, median and standard deviation achieved by the designed FUMA-OENet specify the algorithm's strong stability under less favourable conditions.

6.7 | Overall Performance Analysis of Proposed Model Over Traditional Models

Tables 4 and 5 provide a detailed comparative analysis of the implemented FUMA-OENet mechanism over conventional optimization algorithms and classifiers. The validation is conducted based on some primary performance measures, which offer a comprehensive validation of the classifier's robustness, reliability and the discriminative ability in the AMC. The results in Table 4 display that the designed FUMA-OENet approach surpasses the existing algorithms, such as RSO-OENet, DMO-OENet, CHIO-OENet, and MA-OENet, across all performance measures. The designed FUMA-OENet achieves the highest accuracy of 94.44%, which specifies improvements of 11.95%, 9.99%, 6.16%, and 2.8% contrasted to RSO-OENet, DMO-OENet, CHIO-OENet, and MA-OENet. The designed FUMA-OENet achieves the highest sensitivity of 95.51%, which specifies improvements of 15.44%, 4.7%, 6.04%, and 6.71% contrasted to RSO-OENet, DMO-OENet, CHIO-OENet, and MA-OENet. These improvements illustrate the superior capability of the FUMA optimization operation to improve the feature selection and the classifier parameter optimization, resulting in highly accurate modulation recognition. Table 5 contrasts the FUMA-OENet with existing classifiers: NB, MLP, Adaboost, and Ensemble to estimate the performance gains of the designed FUMA-OENet. The designed FUMA-OENet attains the highest specificity (94.27%), which specifies the improvements of 10.3%, 8.14%, 8.34% and 2.82% to NB, MLP, Adaboost, and Ensemble. Also, the developed FUMA-OENet achieves 72.68% precision, which specifies the improvements of 36.88%, 30.95%, 31.2% and 11.9% contrasted to NB, MLP, Adaboost, and Ensemble. Hence, the designed FUMA-OENet attains superior results across all performance measures, guaranteeing that the designed FUMA-OENet provides more accurate and consistent predictions, outperforming both the conventional algorithms and classifiers.

7 | Conclusion

In conclusion, the designed FUMA-OENet model demonstrated a significant improvement in AMC by combining an improved MA with an optimized machine learning network. Unlike the conventional AMC that depended on static feature extraction or single technique classification, the designed framework employed an FUMA to optimally select the highly discriminative features from the input. The feature optimization process improved the relevance and quality of the extracted features, which are

TABLE 4 | Overall performance analysis of the recommended automatic AMC model compared over the traditional algorithms.

TERMS	RSO-OENet [31]	DMO-OENet [32]	CHIO-OENet [33]	MA-OENet [26]	FUMA-OENet
Accuracy	83.15697	85.00882	88.62434	91.79894	94.44444
Sensitivity	80.76923	91.02564	89.74359	89.10256	95.51282
Specificity	83.53783	84.04908	88.44581	92.22904	94.27403
Precision	43.90244	47.65101	55.33597	64.65116	72.68293
FPR	16.46217	15.95092	11.55419	7.770961	5.725971
FNR	19.23077	8.974359	10.25641	10.89744	4.487179
NPV	83.53783	84.04908	88.44581	92.22904	94.27403
FDR	56.09756	52.34899	44.66403	35.34884	27.31707
F1-Score	56.88488	62.55507	68.45966	74.93261	82.54848
MCC	0.509457	0.587509	0.646891	0.714684	0.803637

TABLE 5 | Overall performance analysis of the designed AMC model compared over the existing classifiers.

TERMS	NB [27]	MLP [28]	Adaboost [29]	Ensemble [30]	FUMA-OENet
Accuracy	84.21517	86.33157	86.24339	91.88713	94.44444
Sensitivity	82.05128	86.53846	85.25641	93.58974	95.51282
Specificity	84.56033	86.29857	86.40082	91.61554	94.27403
Precision	45.87814	50.18587	50	64.03509	72.68293
FPR	15.43967	13.70143	13.59918	8.384458	5.725971
FNR	17.94872	13.46154	14.74359	6.410256	4.487179
NPV	84.56033	86.29857	86.40082	91.61554	94.27403
FDR	54.12186	49.81413	50	35.96491	27.31707
F1-Score	58.85057	63.52941	63.03318	76.04167	82.54848
MCC	0.532717	0.589793	0.582493	0.732262	0.803637

further employed by an ensemble of classifiers, including NB, MLP, capsuleNet and Adaboost for attaining accurate and robust classification. The fusion of these models via fuzzy-based ranking guarantees balanced decision-making and enhanced generalization across distinct modulation scenarios. By focusing on the accuracy value, the FUMA-OENet model gained 13.52%, 56.36%, 12.99%, and 18.03% over the aforementioned models, respectively: RSO-OENet, DMO-OENet, CHIO-OENet, and MA-OENet. Thus, the integration of FUMA-based optimization not only improves the efficiency of the feature selection but also improves the reliability and the adaptability of the ensemble network, guaranteeing that the designed approach provides more stable, accurate and computationally efficient solutions for the advanced communication systems contrasted to the existing AMC approaches.

Limitations of the Study

Though the designed FUMA-OENet approach provides superior robustness, accuracy and generalization contrasted to existing AMC approaches, specific problems remain that need further analysis. One of the problems is the integration of distinct classifiers, and the FUMA-based optimization task increases the training time and overall computational complexity, which limits the real-time deployment in the resource-constrained environment. Therefore, the future work will be focused on improving the computational efficiency of the FUMA-OENet integration, making the system more suitable for real-time applications and deployment in systems with limited processing capabilities.

Funding

This research did not receive any specific funding.

Data Availability Statement

The data that support the findings of this study are available from the upon reasonable request.

References

1. W.-T. Zhang, D. Cui, and S.-T. Lou, "Training Images Generation for CNN Based Automatic Modulation Classification," *IEEE Access* 9 (2021): 62916–62925, https://doi.org/10.1109/ACCESS.2021.3073845.

2. Y. Wang, L. Guo, Y. Zhao, J. Yang, B. Adebisi, and H. Gacanin, "Distributed Learning for Automatic Modulation Classification in Edge Devices," *IEEE Wireless Communications Letters* 9, no. 12 (2020): 2177–2181, https://doi.org/10.1109/LWC.2020.3016822.

3. S. Huang, R. Dai, J. Huang, et al., "Automatic Modulation Classification Using Gated Recurrent Residual Network," *IEEE Internet of Things Journal* 7, no. 8 (2020): 7795–7807, https://doi.org/10.1109/JIOT.2020.2991052.

4. M. M. Elsagheer and S. M. Ramzy, "A Hybrid Model for Automatic Modulation Classification Based on Residual Neural Networks and Long Short-Term Memory," *Alexandria Engineering Journal* 67 (2023): 117–128, https://doi.org/10.1016/j.aej.2022.08.019.

5. B. Xu, U. A. Bhatti, H. Tang, et al., "Towards Explainability for AI-Based Edge Wireless Signal Automatic Modulation Classification," *Journal of Cloud Computing* 13 (2024): 10, https://doi.org/10.1186/s13677-024-00590-3.

6. L. Guo, R. Gao, Y. Cong, and L. Yang, "Robust Automatic Modulation Classification Under Noise Mismatch," *EURASIP Journal on Advances in Signal Processing* 2023 (2023): 73.

7. Q. Zhou, R. Zhang, F. Zhang, and X. Jing, "An Automatic Modulation Classification Network for IoT Terminal Spectrum Monitoring Under Zero-Sample Situations," *EURASIP Journal on Wireless Communications and Networking* 2022 (2022): 25, https://doi.org/10.1186/s13638-022-02099-2.

8. P. Qi, X. Zhou, S. Zheng, and Z. Li, "Automatic Modulation Classification Based on Deep Residual Networks With Multimodal Information," *IEEE Transactions on Cognitive Communications and Networking* 7, no. 1 (2021): 21–33, https://doi.org/10.1109/TCCN.2020.3023145.

9. H. Zhang, Y. Wang, L. Xu, T. Aaron Gulliver, and C. Cao, "Automatic Modulation Classification Using a Deep Multi-Stream Neural Network," *IEEE Access* 8 (2020): 43888–43897, https://doi.org/10.1109/ACCESS.2020.2971698.

10. K. Liao, Y. Zhao, J. Gu, Y. Zhang, and Y. Zhong, "Sequential Convolutional Recurrent Neural Networks for Fast Automatic Modulation Classification," *IEEE Access* 9 (2021): 27182–27188, https://doi.org/10.1109/ACCESS.2021.3053427.

11. Y. Wang, J. Gui, Y. Yin, et al., "Automatic Modulation Classification for MIMO Systems via Deep Learning and Zero-Forcing Equalization," *IEEE Transactions on Vehicular Technology* 69, no. 5 (2020): 5688–5692, https://doi.org/10.1109/TVT.2020.2981995.

12. S. Chang, S. Huang, R. Zhang, Z. Feng, and L. Liu, "Multitask-Learning-Based Deep Neural Network for Automatic Modulation Classification," *IEEE Internet of Things Journal* 9, no. 3 (2022): 2192–2206, https://doi.org/10.1109/JIOT.2021.3091523.

13. Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, H. Sari, and F. Adachi, "Transfer Learning for Semi-Supervised Automatic Modulation Classification in ZF-MIMO Systems," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 10, no. 2 (2020): 231–239, https://doi.org/10.1109/JETCAS.2020.2992128.

14. D. Zhang, W. Ding, C. Liu, H. Wang, and B. Zhang, "Modulated Autocorrelation Convolution Networks for Automatic Modulation Classification Based on Small Sample Set," *IEEE Access* 8 (2020): 27097–27105, https://doi.org/10.1109/ACCESS.2020.2971586.

15. R. Utrilla, E. Fonseca, A. Araujo, and L. A. Dasilva, "Gated Recurrent Unit Neural Networks for Automatic Modulation Classification With Resource-Constrained End-Devices," *IEEE Access* 8 (2020): 112783–112794, https://doi.org/10.1109/ACCESS.2020.3002770.

16. J. P. Mouton, M. Ferreira, and A. S. J. Helberg, "A Comparison of Clustering Algorithms for Automatic Modulation Classification," *Expert Systems with Applications* 151 (2020): 113317, https://doi.org/10.1016/j.eswa.2020.113317.

17. M. Patel, X. Wang, and S. Mao, "Data Augmentation With Conditional GAN for Automatic Modulation Classification", WiseML '20: Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, July 2020, 31–36, https://doi.org/10.1145/3395352.3402622.

18. M. A. Pinto-Orellana and H. L. Hammer, "Dyadic Aggregated Autoregressive Model (DASAR) for Automatic Modulation Classification," *IEEE Access* 8 (2020): 156096–156103, https://doi.org/10.1109/ACCESS.2020.3019243.

19. X. Zhang, J. Sun, and X. Zhang, "Automatic Modulation Classification Based on Novel Feature Extraction Algorithms," *IEEE Access* 8 (2020): 16362–16371, https://doi.org/10.1109/ACCESS.2020.2966019.

20. Y. Chen, W. Shao, J. Liu, L. Yu, and Z. Qian, "Automatic Modulation Classification Scheme Based on LSTM With Random Erasing and Attention Mechanism," *IEEE Access* 8 (2020): 154290–154300, https://doi.org/10.1109/ACCESS.2020.3017641.

21. Y. Lin, Y. Tu, and Z. Dou, "An Improved Neural Network Pruning Technology for Automatic Modulation Classification in Edge Devices," *IEEE Transactions on Vehicular Technology* 69, no. 5 (2020): 5703–5706, https://doi.org/10.1109/TVT.2020.2983143.

22. S. Ansari, K. A. Alnajjar, M. Saad, S. Abdallah, and A. A. El-Moursy, "Automatic Digital Modulation Recognition Based on Genetic-Algorithm-Optimized Machine Learning Models," *IEEE Access* 10 (2022): 50265–50277, https://doi.org/10.1109/ACCESS.2022.3171909.

23. L. Weng, Y. He, J. Peng, J. Zheng, and X. Li, "Deep Cascading Network Architecture for Robust Automatic Modulation Classification," *Neurocomputing* 455 (2021): 308–324, https://doi.org/10.1016/j.neucom.2021.05.010.

24. Y. Liu, X. Yan, Q. Liu, T. An, and J. Dai, "Automatic Modulation Recognition of Radio Fuzes Using a DR2D-Based Adaptive Denoising Method and Textural Feature Extraction," Defence Technology, Available Online 24 December 2023.

25. H. Bai, M. Huang, and J. Yang, "An Efficient Automatic Modulation Classification Method Based on the Convolution Adaptive Noise Reduction Network," *ICT Express* 9, no. 5 (2023): 834–840, https://doi.org/10.1016/j.icte.2022.11.001.

26. P. Trojovský and M. Dehghani, "Migration Algorithm: A New Human-Based Metaheuristic Approach for Solving Optimization Problems," *Computer Modeling in Engineering & Sciences* 137, no. 2 (2023): 1695–1730.

27. S.-B. Kim, K.-S. Han, H.-C. Rim, and S. H. Myaeng, "Some Effective Techniques for Naive Bayes Text Classification," *IEEE Transactions on Knowledge and Data Engineering* 18, no. 11 (2006): 1457–1466.

28. S. Mitra, R. K. De, and S. K. Pal, "Knowledge-Based Fuzzy MLP for Classification and Rule Generation," *IEEE Transactions on Neural Networks* 8, no. 6 (1997): 1338–1350, https://doi.org/10.1109/72.641457.

29. P. Pandey and R. Prabhakar, "An Analysis of Machine Learning Techniques (J48 & AdaBoost)-for Classification," 2016 1st India International Conference on Information Processing (IICIP), Delhi, India, 2016, 1–6.

30. A. Rojarath, W. Songpan, and C. Pong-inwong, "Improved Ensemble Learning for Classification Techniques Based on Majority Voting," 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2016, https://doi.org/10.1109/ICSESS.2016.7883026.

31. G. Dhiman, M. Garg, A. Nagar, V. Kumar, and M. Dehghani, "A Novel Algorithm for Global Optimization: Rat Swarm Optimizer," *Journal of Ambient Intelligence and Humanized Computing* 12 (2021): 8457–8482, https://doi.org/10.1007/s12652-020-02580-0.

32. J. O. Agushaka, A. E. Ezugwu, and L. Abualigah, "Dwarf Mongoose Optimization Algorithm," *Computer Methods in Applied Mechanics and Engineering* 391 (2022): 114570, https://doi.org/10.1016/j.cma.2022.114570.

33. A. S. Mahboob, H. S. Shahhoseini, M. R. Ostadi Moghaddam, and S. Yousefi, "A Coronavirus Herd Immunity Optimizer for Intrusion Detection System," 2021 29th Iranian Conference on Electrical Engineering (ICEE), Tehran, Islamic Republic of Iran, 2021, https://doi.org/10.1109/ICEE52715.2021.9544165.