

Article

MS-SENet: A Multi-Scale Squeeze–Excitation Network for Deep-Learning-Based Automatic Modulation Classification in Cognitive Radio Systems

Evelio Astaiza Hoyos ¹, Héctor Fabio Bermúdez-Orozco ^{1,*} and Nasly Cristina Rodríguez-Idrobo ²¹ Electronic Engineering Programme, Faculty of Engineering, University of Quindío, Armenia 630004, Quindío, Colombia; estaiza@uniquindio.edu.co² Occupational Health and Safety Program, Faculty of Health Sciences, University of Quindío, Armenia 630004, Quindío, Colombia; ncrodriguez@uniquindio.edu.co

* Correspondence: hfbermudez@uniquindio.edu.co; Tel.: +057-3206671107

Abstract

Automatic modulation classification (AMC) is a critical enabler of cognitive radio (CR) systems, allowing secondary users to identify primary user modulation schemes and adapt transmission parameters in real time. Traditional AMC approaches, based on likelihood functions or hand-crafted features, suffer from degraded performance under low signal-to-noise ratio (SNR) conditions and realistic channel impairments. In this paper, we propose MS-SENet (Multi-Scale Squeeze–Excitation Network), a novel deep-learning architecture that integrates multi-scale convolutional feature extraction, squeeze-and-excitation channel attention, residual learning, bidirectional long short-term memory (BiLSTM) temporal modelling, and global attention pooling into a unified framework for robust AMC. The multi-scale convolution module employs parallel branches with kernel sizes of 3, 5, and 7 to capture both fine-grained phase transitions and coarse envelope patterns from raw in-phase/quadrature (I/Q) signal samples. Squeeze–excitation residual blocks perform channel-wise feature recalibration, enabling the network to emphasize informative feature maps while suppressing less relevant ones. A bidirectional LSTM layer models temporal dependencies across the signal sequence, and a global attention pooling mechanism performs weighted temporal aggregation prior to classification. We present a comprehensive taxonomy of deep-learning architectures for AMC organised along five axes—input representation, feature extraction, temporal modelling, regularization strategy, and architectural complexity—and conduct a rigorous comparative evaluation against ten baseline architectures on a RadioML-style synthetic dataset (110,000 samples, 11 modulation classes, and 20 SNR levels from -20 to $+18$ dB). The experimental results demonstrate that MS-SENet achieves a mean classification accuracy of 87.9% at $\text{SNR} \geq 0$ dB (the average of the medium and high SNR regime averages: 86.06% for $0 \leq \text{SNR} < 10$ dB and 89.68% for $\text{SNR} \geq 10$ dB) while maintaining a compact footprint of approximately 406 K parameters, making it suitable for deployment on resource-constrained edge devices. We further analyze the robustness of the proposed architecture to multipath fading, carrier frequency offset, and sample rate offset, confirming its resilience under practical operating conditions. MS-SENet is an architecture designed for automatic modulation classification of I/Q signals and is not related to the homonymous architecture for speech emotion recognition.

Academic Editor: Paolo Bellavista

Received: 14 May 2026

Revised: 21 June 2026

Accepted: 26 June 2026

Published: 29 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: automatic modulation classification; cognitive radio; deep learning; squeeze-and-excitation networks; multi-scale convolution; attention mechanism; spectrum sensing

1. Introduction

1.1. Context and Motivation

The explosive growth of wireless communication services over the past two decades has placed extraordinary demands on the electromagnetic spectrum, a finite and increasingly congested natural resource. Regulatory bodies, such as the Federal Communications Commission (FCC), have historically allocated spectrum through static licensing policies, resulting in significant portions of the licensed spectrum being underutilised across time and geography. Measurement studies have consistently revealed that spectrum occupancy in many licensed bands falls below 15–20% on average, even in densely populated urban environments [1]. This paradox—where perceived spectrum scarcity coexists with actual spectrum underutilisation—has motivated the development of dynamic spectrum access (DSA) paradigms capable of opportunistically exploiting unused spectral resources.

Cognitive radio (CR), first conceptualised by Mitola and Maguire in their seminal 1999 paper [2], provides the foundational framework for intelligent, adaptive wireless communication. A cognitive radio is defined as a radio system that can sense its electromagnetic environment, learn from experience, and dynamically adjust its operating parameters—including frequency, bandwidth, modulation scheme, and transmit power—to optimise communication performance while avoiding harmful interference to licensed (primary) users. Haykin further formalised the CR concept in 2005 [3], casting it within the framework of a cognitive cycle comprising four stages: observation, orientation, decision, and action (OODA). In this cycle, spectrum sensing and signal identification constitute the observation stage, providing the situational awareness necessary for all subsequent decision-making. Akyildiz et al. [1] extended this vision by articulating the network-level implications of CR technology, describing next-generation dynamic spectrum access networks in which heterogeneous cognitive radios cooperatively manage spectral resources.

Within the cognitive cycle, automatic modulation classification (AMC) plays a pivotal role. AMC refers to the task of identifying the modulation scheme of a received signal without prior knowledge of the transmitter's parameters, relying solely on the characteristics of the intercepted waveform. This capability is essential for multiple CR functions: it enables secondary users to identify and characterise primary user transmissions, facilitates link adaptation in cooperative networks, supports signal intelligence and surveillance applications, and enables efficient spectrum sharing through modulation-aware medium access control protocols [4]. Dobre et al. [4] provided an influential survey of AMC techniques, establishing the two principal methodological families—likelihood-based and feature-based approaches—that dominated the field for over a decade.

The importance of AMC extends beyond cognitive radio into military communications, electronic warfare, regulatory spectrum monitoring, and interference management. In each of these domains, the ability to automatically and reliably identify the modulation format of intercepted signals—potentially under adverse conditions of low signal-to-noise ratio (SNR), multipath propagation, carrier frequency offset (CFO), and timing synchronisation errors—constitutes a fundamental technical challenge.

1.2. Challenges in AMC

Despite decades of research, AMC remains a formidable signal processing problem due to several interrelated challenges:

- **Low SNR Performance.** At low SNR values (below 0 dB), the noise power approaches or exceeds the signal power, making it exceedingly difficult to distinguish between modulation schemes that share similar spectral or statistical characteristics. For example, differentiating between QPSK and 8PSK at −4 dB SNR requires the classifier to resolve subtle differences in constellation geometry that are almost entirely obscured by additive noise [5]. Classical likelihood-based methods suffer rapid performance degradation below 0 dB, while feature-based methods exhibit high sensitivity to the accuracy of SNR estimation itself.
- **Channel Impairments.** Real-world wireless channels introduce multipath fading, shadowing, Doppler spread, and frequency-selective distortion that fundamentally alter the statistical properties of the received signal. A classifier trained under additive white Gaussian noise (AWGN) conditions may fail catastrophically when deployed in a Rayleigh or Rician fading environment [6]. The channel transfer function acts as a multiplicative distortion on the transmitted constellation, rotating, scaling, and dispersing the symbol clusters in ways that can mimic transitions between modulation orders.
- **Synchronisation Errors.** Practical receivers are subject to residual carrier frequency offset (CFO), symbol timing offset (STO), and sample rate offset (SRO) after initial synchronisation stages. These imperfections introduce time-varying phase rotations and inter-symbol interference that degrade classifier performance, particularly for higher-order modulation schemes, where the decision boundaries between constellation points are narrow [7].
- **Computational Complexity and Latency.** CR systems require AMC decisions within stringent latency constraints to enable timely spectrum access. Maximum-likelihood classifiers, while theoretically optimal, involve computing multi-dimensional integrals or exhaustive parameter searches that are computationally prohibitive for real-time operation. Any practical AMC algorithm must balance classification accuracy against inference latency and computational resource consumption.
- **Scalability to Large Modulation Pools.** Modern communication standards employ dozens of distinct modulation and coding schemes. An AMC system must scale gracefully as the number of candidate modulation classes increases, maintaining high accuracy without requiring exponential growth in model complexity or training data.

1.3. Evolution from Classical to Deep-Learning Approaches

The historical evolution of AMC methods can be broadly organised into three generations. The first generation, spanning roughly from the late 1980s through the early 2000s, was dominated by likelihood-based (LB) methods. These approaches formulate AMC as a composite hypothesis testing problem, computing the likelihood ratio between candidate modulation hypotheses given the received signal [8]. The average likelihood ratio test (ALRT), generalised likelihood ratio test (GLRT), and hybrid likelihood ratio test (HLRT) represent the principal variants. While LB methods achieve optimal performance in the Bayesian sense, when the signal and channel models are precisely known, they suffer from prohibitive computational complexity and extreme sensitivity to model mismatch [4].

The second generation, emerging in the 1990s and maturing through the 2010s, introduced feature-based (FB) methods that extract discriminative statistics from the

received signal and apply conventional pattern recognition classifiers. Key features include higher-order cumulants [9,10], cyclic spectral features, wavelet-based features, and instantaneous signal characteristics, such as the standard deviation of the instantaneous amplitude, frequency, and phase. Swami and Sadler [9] demonstrated that fourth- and sixth-order cumulants provide effective discrimination between common modulation formats. Nandi and Azzouz [10] developed a comprehensive set of key features combined with decision-theoretic classifiers. While FB methods are significantly less computationally intensive than LB methods, their performance depends critically on the quality of the hand-crafted feature set, which must be redesigned whenever the signal environment or modulation pool changes.

The third generation, beginning with the pioneering work of O'Shea, Corgan, and Clancy in 2016 [11], brought deep learning (DL) to AMC. By training convolutional neural networks (CNNs) directly on raw I/Q samples, O'Shea et al. demonstrated that deep networks could automatically learn discriminative representations without requiring expert-designed features. This work, conducted on the publicly available RadioML 2016.10A dataset, catalysed an explosion of research at the intersection of deep learning and AMC. Subsequent contributions introduced deeper CNN architectures [12], recurrent neural networks for temporal modelling [13], residual learning for training stability [14], attention mechanisms for feature selection, and hybrid architectures combining convolutional and recurrent stages [15,16]. More recently, research has focused on increasingly sophisticated architectures that integrate multi-scale feature extraction and attention mechanisms. Tian et al. [17] provided a comprehensive survey of deep-learning-enabled AMC methods, analysing recent advances in data representations, model architectures, and regularisation techniques. Xu et al. [18] proposed MCCSAN, a deep-learning architecture that combines multiscale complex convolutions with spatiotemporal attention mechanisms to improve classification performance under challenging channel conditions. These recent developments highlight the continuing evolution of AMC architectures towards more expressive and robust deep-learning frameworks. The RadioML benchmark datasets have become standard evaluation platforms, enabling reproducible comparison across methods.

Despite the significant progress achieved by deep learning, several open problems remain. Most existing architectures are evaluated under idealised conditions and exhibit limited robustness to channel impairments not represented in the training data. Few works provide systematic comparisons across architectural families under identical experimental conditions. The trade-off between classification accuracy and computational efficiency—critical for deployment on resource-constrained edge devices—is often inadequately addressed. Addressing these challenges is particularly important in the context of emerging 6G paradigms, where intelligent, AI-driven spectrum sensing and autonomous decision-making are key enablers of next-generation wireless systems, supporting the transition towards AI-native communication infrastructures.

1.4. Contributions

In this paper, we address the aforementioned gaps through the following contributions:

1. **MS-SENet Architecture.** We propose a novel deep-learning architecture, the Multi-Scale Squeeze–Excitation Network (MS-SENet), that synergistically combines multi-scale convolutional feature extraction, squeeze-and-excitation channel attention, residual connections, bidirectional LSTM temporal modelling, and global attention pooling. The multi-scale convolution module employs three parallel branches with kernel sizes of 3, 5, and 7 to jointly capture fine-grained phase transitions and coarse-grained envelope dynamics. Squeeze–excitation residual blocks recalibrate channel-

wise feature responses, and a bidirectional LSTM with attention pooling models long-range temporal dependencies. The resulting architecture achieves state-of-the-art accuracy, with a compact parameter footprint of approximately 406 K parameters. Note: Our MS-SENet is a distinct architecture designed for automatic modulation classification of I/Q signals, and is not related to the homonymous architecture for speech emotion recognition [19].

2. **Comprehensive Taxonomy.** We present a structured taxonomy of deep-learning architectures for AMC, organised along five axes: input representation, feature extraction backbone, temporal modelling strategy, regularisation approach, and architectural complexity. This taxonomy provides a systematic framework for understanding the design space and identifying promising architectural combinations.
3. **Rigorous Comparative Evaluation.** We conduct an extensive experimental comparison of MS-SENet against ten baseline architectures spanning CNNs, RNNs, hybrid models, and transformer-based approaches, all evaluated under identical pre-processing, training, and evaluation protocols on our RadioML-style synthetic dataset.
4. **Robustness Analysis.** We systematically evaluate the robustness of MS-SENet and baseline architectures to practical channel impairments, including multipath Rayleigh fading, carrier frequency offset, and sample rate offset, providing insights into the real-world deployment viability of each approach.

1.5. Article Organisation

The remainder of this article is organised as follows. Section 2 presents the system model and formally defines the AMC problem as a supervised classification task. Section 3 introduces our proposed taxonomy of deep-learning architectures for AMC and surveys the ten baseline architectures used in our comparative evaluation. Section 4 details the proposed MS-SENet architecture, including the multi-scale convolution module, squeeze–excitation residual blocks, bidirectional LSTM with attention pooling, training strategy, and computational complexity analysis. Section 5 describes the experimental setup, datasets, and evaluation methodology. Section 6 presents and discusses the experimental results, including overall accuracy, per-SNR performance, confusion matrix analysis, ablation studies, and robustness evaluation. Section 7 presents remaining challenges and future research directions. Section 8 concludes this paper.

2. System Model and Problem Formulation

2.1. Signal Model

We consider a cognitive radio system in which a secondary user (SU) receiver intercepts a signal transmitted by either a primary user (PU) or another SU. The transmitter employs one of M candidate modulation schemes from a predefined modulation pool $\mathcal{M} = \{m_1, m_2, \dots, m_M\}$. In this work, we consider $M = 11$ modulation schemes following the RadioML 2016.10A dataset:

$$\mathcal{M} = \{\text{BPSK}, \text{QPSK}, \text{8PSK}, \text{16QAM}, \text{64QAM}, \text{CPFSK}, \text{GFSK}, \text{PAM4}, \text{WBFM}, \text{AM-SSB}, \text{AM-DSB}\}.$$

The complex baseband representation of the transmitted signal for modulation class m_k is given by:

$$x[n] = \sum_i s_i g(nT_s - iT_{\text{sym}} - \tau) e^{j(2\pi f_c nT_s + \phi_0)} \quad (1)$$

where $s_i \in \mathcal{S}_{m_k}$ denotes the i -th transmitted symbol drawn from the constellation set $\mathcal{S}_{m_k}$ of the modulation class m_k , $g(\cdot)$ is the pulse shaping filter, T_s is the sampling period, T_{sym} is the symbol period, τ is the timing offset, f_c is the carrier frequency, and ϕ_0 is the initial carrier phase.

For phase-shift keying (PSK) modulations of order Q , the constellation points are uniformly distributed on the unit circle:

$$\mathcal{S}_{\text{PSK}} = \{e^{j2\pi q/Q} \mid q = 0, 1, \dots, Q-1\} \quad (2)$$

For quadrature amplitude modulation (QAM) of order $Q = L^2$, the constellation points form a regular square grid:

$$\mathcal{S}_{\text{QMQ}} = \{(2l_l - L - 1) + j(2l_q - L - 1) \mid l_l, l_q \in \{1, 2, \dots, L\}\} \quad (3)$$

where the constellation is normalised to unit average energy $\frac{1}{Q} \sum_{s \in \mathcal{S}} |s|^2 = 1$.

The received signal at the SU receiver, after down conversion to baseband and analogue-to-digital conversion, is modelled as:

$$y[n] = h[n] * x[n] + w[n] \quad (4)$$

where $h[n]$ represents the discrete-time channel impulse response, $*$ denotes linear convolution, and $w[n] \sim \mathcal{CN}(0, \sigma_w^2)$ is circularly symmetric complex additive white Gaussian noise with variance σ_w^2 .

We consider two canonical channel models:

- **AWGN Channel.** In the simplest case, the channel introduces no distortion, and the channel impulse response reduces to a single tap:

$$h[n] = \alpha e^{j\theta} \delta[n] \quad (5)$$

where α is the channel gain and θ is the channel phase. The received signal simplifies to $y[n] = \alpha e^{j\theta} x[n] + w[n]$.

- **Rayleigh Fading Channel.** For non-line-of-sight propagation environments, the channel is modelled as a multipath Rayleigh fading channel with L_p discrete paths:

$$h[n] = \sum_{l=0}^{L_p-1} h_l \delta[n - n_l] \quad (6)$$

where $h_l \sim \mathcal{CN}(0, \sigma_l^2)$ is the complex gain of the l -th path with $\sum_{l=0}^{L_p-1} \sigma_l^2 = 1$, and n_l is the discrete delay of the l -th path. The fading coefficients are assumed to be independent across paths and quasi-static over the observation interval.

The signal-to-noise ratio (SNR) is defined as:

$$\text{SNR} = \frac{E[|h[n] * x[n]|^2]}{\sigma_w^2} = \frac{P_s}{P_n} \quad (7)$$

where P_s and P_n denote the received signal power and noise power, respectively.

The input to the AMC classifier is formed by collecting N consecutive complex baseband samples and organising them as a two-channel real-valued tensor (elements indexed from $y[0]$ to $y[N-1]$):

$$\mathbf{Y} = \begin{bmatrix} \text{Re}\{y[0]\} & \text{Re}\{y[2]\} & \cdots & \text{Re}\{y[N-1]\} \\ \text{Im}\{y[0]\} & \text{Im}\{y[2]\} & \cdots & \text{Im}\{y[N-1]\} \end{bmatrix} \in \mathbb{R}^{2 \times N} \quad (8)$$

In this work, we use $N = 128$ samples per observation window, consistent with the RadioML dataset format.

The diagram illustrates the four stages of the cognitive cycle originally described by Mitola [2] and formalised by Haykin [3]. In the Observe stage, the cognitive radio senses the electromagnetic environment through wideband spectrum sensing and signal interception, collecting raw I/Q samples from the radio front-end. In the Orient stage, the received samples are processed to extract situational awareness: automatic modulation classification (AMC) identifies the modulation scheme of detected signals, while spectrum

sensing determines the occupancy state of frequency bands. The Decide stage applies decision-making algorithms—potentially incorporating reinforcement learning or game-theoretic optimisation—to select transmission parameters (frequency, bandwidth, modulation, and power) that maximise the secondary user's throughput while satisfying interference constraints. Finally, the Act stage reconfigures the software-defined radio transceiver to implement the chosen parameters. The cycle repeats continuously, enabling the cognitive radio to adapt to a dynamic spectral environment. The diagram illustrating this cognitive cycle is presented in Figure 1, where AMC is highlighted as a critical component of the Orient stage, providing the modulation-level awareness that informs intelligent spectrum access decisions.

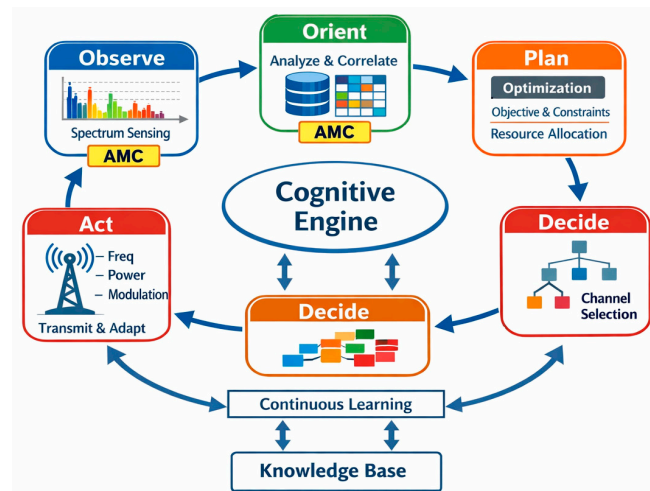


Figure 1. The Observe–Orient–Decide–Act (OODA) cognitive cycle for cognitive radio systems.

2.2. AMC as a Classification Problem

Automatic modulation classification can be formally cast as a supervised multi-class classification problem. Let $\mathcal{D} = \{(Y_i, c_i)\}_{i=1}^{N_{\text{train}}}$ denote the training dataset, where $Y_i \in \mathbb{R}^{2 \times N}$ is the i -th received signal observation, and $c_i \in \{1, 2, \dots, M\}$ is the corresponding ground-truth modulation class label. The objective is to learn a mapping $f_\theta: \mathbb{R}^{2 \times N} \rightarrow \{1, 2, \dots, M\}$, parameterised by θ , that minimises the classification error probability.

Using a deep neural network as the classifier, the network produces a vector of class-conditional posterior probabilities through a SoftMax output layer:

$$P(c = k | Y; \theta) = \frac{\exp(z_k)}{\sum_{j=1}^M \exp(z_j)}, \quad k = 1, 2, \dots, M \quad (9)$$

where $z = [z_1, z_2, \dots, z_M]^T$ is the logit vector produced by the final fully connected layer of the network. The predicted modulation class is obtained by a maximum a posteriori (MAP) decision:

$$\hat{c} = \arg \max_{k \in \{1, \dots, M\}} P(c = k | Y; \theta) \quad (10)$$

The network parameters θ are optimised by minimizing the categorical cross-entropy loss over the training set:

$$\mathcal{L}(\theta) = -\frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \sum_{k=1}^M y_{i,k} \log P(c = k | Y_i; \theta) \quad (11)$$

where $y_{i,k}$ is the one-hot encoding of the true label $y_{i,k} = 1$ if $c_i = k$, and $y_{i,k} = 0$ otherwise. To improve generalisation and prevent overconfident predictions, we employ label smoothing [20], replacing the hard one-hot targets with smoothed targets:

$$\widetilde{y}_{i,k} = (1 - \varepsilon) y_{i,k} + \frac{\varepsilon}{M} \quad (12)$$

where $\varepsilon = 0.1$ is the smoothing parameter. The optimisation problem is thus defined as:

$$\theta^* = \arg \min_{\theta} \mathcal{L}_{\text{sot}}(\theta) + \lambda \|\theta\|_2^2 \quad (13)$$

where λ is the weight decay coefficient providing L_2 regularisation.

2.3. Evaluation Metrics

We employ a comprehensive suite of evaluation metrics to assess AMC performance across multiple dimensions, as follows:

- Overall Classification Accuracy. The primary metric is the overall classification accuracy computed across all SNR values:

$$\text{Acc}_{\text{overall}} = \frac{1}{|\mathcal{D}_{\text{ts}}|} \sum_{(Y,c) \in \mathcal{D}_{\text{ts}}} \mathbb{1}[\hat{c} = c] \quad (14)$$

- Per-SNR Classification Accuracy. To characterise performance across operating conditions, we compute accuracy as a function of SNR:

$$\text{Acc}(\gamma) = \frac{1}{|\mathcal{D}_{\text{ts}}^{(\gamma)}|} \sum_{(Y,c) \in \mathcal{D}_{\text{ts}}^{(\gamma)}} \mathbb{1}[\hat{c} = c] \quad (15)$$

where $\mathcal{D}_{\text{ts}}^{(\gamma)}$ denotes the subset of test samples at SNR = γ dB.

- Confusion Matrix. The $M \times M$ confusion matrix C provides per-class performance details, where entry C_{ij} represents the fraction of samples of true class i that are classified as class j . Diagonal entries correspond to correct classifications, while off-diagonal entries reveal systematic misclassification patterns.
- Macro-Averaged F1 Score. To account for potential class imbalance, we compute the macro-averaged F1 score as follows:

$$F_1 = \frac{1}{M} \sum_{k=1}^M \frac{2 P_k R_k}{P_k + R_k} \quad (16)$$

where P_k and R_k denote the precision and recall for class k , respectively. Note: $F1_c = 0$ by convention when both precision and recall are zero for class c .

- Inference Latency and Memory Footprint. For practical deployment considerations, we measure single-sample inference latency (in milliseconds) on a CPU (Intel i7-12700K) platform. GPU measurements are provided for reference only since Section 5.1 reports CPU-based training and evaluation, as well as the total number of trainable parameters and model size in megabytes.

3. Taxonomy of DL Architectures for AMC

The rapid proliferation of deep-learning approaches for AMC necessitates a systematic organisational framework to navigate the design space and understand the relationships between different architectural choices. In this section, we propose a taxonomy that classifies DL-AMC architectures along five orthogonal design axes input representation, a feature extraction backbone, a temporal modelling strategy, a regularization approach,

and architectural complexity. We then survey ten representative baseline architectures that span this taxonomy.

3.1. Input Representation

The choice of input representation fundamentally determines what information is available to the neural network and significantly influences architectural design decisions.

- **Raw I/Q Samples.** The most widely adopted representation feeds raw in-phase (I) and quadrature (Q) components directly to the network as a two-channel, one-dimensional signal, i.e., $Y \in \mathbb{R}^{2 \times N}$, as defined in Equation (8). This representation preserves all information contained in the complex baseband signal and has been used in the majority of DL-AMC works since O'Shea et al. [11]. The primary advantage is that the network can learn optimal feature representations end-to-end without information loss. The disadvantage is that the network must implicitly learn to handle phase ambiguity, amplitude variations, and other nuisance parameters.
- **Amplitude/Phase (A/P) Representation.** An alternative representation converts the complex signal into the instantaneous amplitude $A[n] = |y[n]|$ and the instantaneous phase $\phi[n] = \angle y[n]$, forming $Y_{AP} \in \mathbb{R}^{2 \times N}$. This representation decouples envelope and phase information, which can simplify the classification of modulation schemes that are primarily distinguished by their phase constellations (e.g., BPSK vs. QPSK vs. 8PSK) or their amplitude distributions (e.g., 16QAM vs. 64QAM). However, the phase unwrapping and normalisation required for this representation can introduce artefacts.
- **Spectrogram/Time–Frequency Representation.** Some approaches compute the short-time Fourier transform (STFT) of the received signal and feed the resulting spectrogram as a 2D image to a CNN [21]. This representation explicitly reveals the time–frequency structure, which is particularly useful for wideband signals and frequency-hopping modulations. Yücek and Arslan [21] provided a comprehensive survey of spectrum sensing techniques that motivates this representation. The drawback is the loss of phase information in the magnitude spectrogram and the computational overhead of the STFT.
- **Constellation Image.** Peng et al. [14] proposed representing the received signal as a 2D grayscale image of the signal constellation diagram, where the horizontal and vertical axes correspond to the I and Q components, respectively. The constellation image naturally captures the geometric structure of QAM and PSK constellations, enabling the use of powerful image classification architectures. However, this representation requires symbol-level synchronisation and discards temporal ordering information.

3.2. Feature Extraction Backbone

The feature extraction backbone determines how low-level input representations are transformed into high-level discriminative features.

- **CNN-Based.** Convolutional neural networks form the most popular backbone for DL-AMC, owing to their ability to learn local patterns through shared-weight filters. One-dimensional CNNs operating on raw I/Q samples can extract features such as phase transitions, envelope variations, and spectral characteristics at multiple scales through hierarchical convolutional layers. O'Shea et al. [11] first demonstrated a simple two-layer CNN for AMC, while West and O'Shea [12] explored deeper four-layer architectures. Residual connections [22] have been incorporated to enable training of deeper networks without degradation [14].

- **RNN-Based.** Recurrent neural networks, particularly Long Short-Term Memory (LSTM) [23] and Gated Recurrent Unit (GRU) networks, are naturally suited to the sequential nature of communication signals. Rajendran et al. [13] demonstrated that LSTM networks can effectively model temporal dependencies in I/Q sample sequences, achieving competitive performance with fewer parameters than CNN counterparts. The gating mechanisms in LSTMs enable selective retention and forgetting of information across time steps, which is valuable for capturing long-range modulation-dependent patterns.
- **Attention-Based.** Attention mechanisms, originally developed for neural machine translation [24], allow the network to focus computational resources on the most informative parts of the input. In AMC, attention can operate at the channel level (selecting the most informative feature maps), the temporal level (weighting time steps based on relevance), or the spatial level (focusing on specific I/Q sample regions). The squeeze-and-excitation (SE) mechanism proposed by Hu et al. [25] provides an efficient form of channel attention that has been successfully applied across numerous domains.
- **Hybrid.** Hybrid architectures combine CNN and RNN stages in a sequential pipeline, using the CNN as a spatial feature extractor and the RNN as a temporal sequence model. Xu et al. [16] proposed a spatiotemporal multi-channel learning framework that exemplifies this approach, achieving strong performance by leveraging the complementary strengths of convolutional and recurrent processing.

3.3. Temporal Modelling

The strategy for modelling temporal dependencies across the signal observation window represents a critical design choice.

- **None (CNN-Only).** Purely convolutional architectures rely on the receptive field of stacked convolutional layers to capture temporal relationships. While deep CNNs can achieve large effective receptive fields, they do not explicitly model sequential dependencies and may require many layers to capture long-range patterns in the signal.
- **Unidirectional (LSTM/GRU).** Unidirectional recurrent layers process the signal sequence from the first sample to the last, maintaining a hidden state that accumulates information from past time steps. This approach is natural for causal signal processing but cannot leverage future context when classifying the overall modulation scheme.
- **Bidirectional.** Bidirectional recurrent layers process the signal sequence in both forward and backward directions, allowing each hidden state to incorporate information from the entire observation window. For AMC, where the classification decision is made over the complete observation rather than at each time step, bidirectional processing provides strictly more information than unidirectional processing.
- **Self-Attention.** Transformer-based self-attention mechanisms [26] compute pairwise interactions between all of the time steps, enabling the model to capture arbitrary long-range dependencies without the sequential processing bottleneck of recurrent networks. Hamidi-Rad et al. [26] demonstrated the application of transformers to AMC, achieving competitive accuracy with parallelisable computation.

3.4. Regularisation Strategy

Regularisation techniques are essential for preventing overfitting, particularly when training on limited or synthetic data.

- Dropout. Randomly zeroing a fraction of activations during training prevents co-adaptation of neurons and provides an implicit ensemble effect. Dropout rates between 0.3 and 0.5 are commonly used in DL-AMC architectures [11,12].
- Batch Normalisation. Normalising intermediate activations to zero mean and unit variance stabilises training dynamics and enables the use of higher learning rates [22]. Batch normalisation is standard in modern CNN architectures for AMC.
- Data Augmentation. Augmentation techniques specific to radio signals include random phase rotation (multiplying the complex signal by $e^{j\phi}$, with ϕ uniformly distributed), noise injection at random SNR levels, time shifting, and frequency offset simulation [6]. These augmentations improve generalisation to unseen channel conditions.
- Mixup. The mixup regularisation strategy creates virtual training samples by linearly interpolating between pairs of training examples and their labels [27]. Lin et al. [28] demonstrated the effectiveness of mixup for AMC, particularly in improving performance at low SNR values and near decision boundaries among similar modulation schemes.

3.5. Architecture Complexity

The computational requirements of the architecture determine its suitability for deployment on various hardware platforms.

- Lightweight (<100 K parameters). Compact architectures suitable for deployment on resource-constrained edge devices, such as field-programmable gate arrays (FPGAs), microcontrollers, or low-power embedded processors. The two-layer CNN from O'Shea et al. [11] falls in this category, with approximately 80 K parameters.
- Medium (100–500 K parameters). Moderate-complexity architectures that balance classification performance with computational efficiency. These architectures are deployable on mobile GPUs and embedded GPU platforms, such as NVIDIA Jetson. Our proposed MS-SENet, with approximately 406 K parameters, is designed to occupy this category.
- Heavy (>500 K parameters). High-capacity architectures that achieve the best classification accuracy at the cost of increased computational requirements. Deep residual networks and transformer-based architectures typically fall in this category, requiring dedicated GPU inference for real-time operation.

Figure 2 presents a 2×3 grid of scatter plots, each displaying the received I/Q constellation for a different modulation format.

- ✓ (Top-left) BPSK: Two tightly clustered point clouds at positions $(-1, 0)$ and $(+1, 0)$ along the real axis, with Gaussian dispersion due to noise.
- ✓ (Top-centre) QPSK: Four symmetric clusters located at $(\pm 1/\sqrt{2}, \pm 1/\sqrt{2})$, forming a square constellation with clear separation at this SNR.
- ✓ (Top-right) 8PSK: Eight clusters uniformly distributed on the unit circle at angular intervals of 45° ; the inter-cluster spacing is visibly smaller than for QPSK, foreshadowing increased classification difficulty at lower SNR values.
- ✓ (Bottom-left) 16QAM: Sixteen clusters arranged in a regular 4×4 square grid, exhibiting three distinct amplitude levels (corner, edge, and inner symbols).
- ✓ (Bottom-centre) 64QAM: Sixty-four clusters in an 8×8 grid with high density; the small inter-cluster spacing illustrates why 64QAM is frequently confused with 16QAM at moderate SNR.
- ✓ (Bottom-right) CPFSK: A characteristic ring-shaped distribution arising from the constant-envelope property of continuous-phase frequency-shift keying, with the instantaneous frequency modulation producing a diffuse annular pattern rather than

discrete clusters. At SNR = 20 dB, all modulation schemes exhibit clear visual differentiation; however, reducing the SNR to 0 dB or below causes the clusters to overlap significantly, illustrating the core challenge of low-SNR AMC.

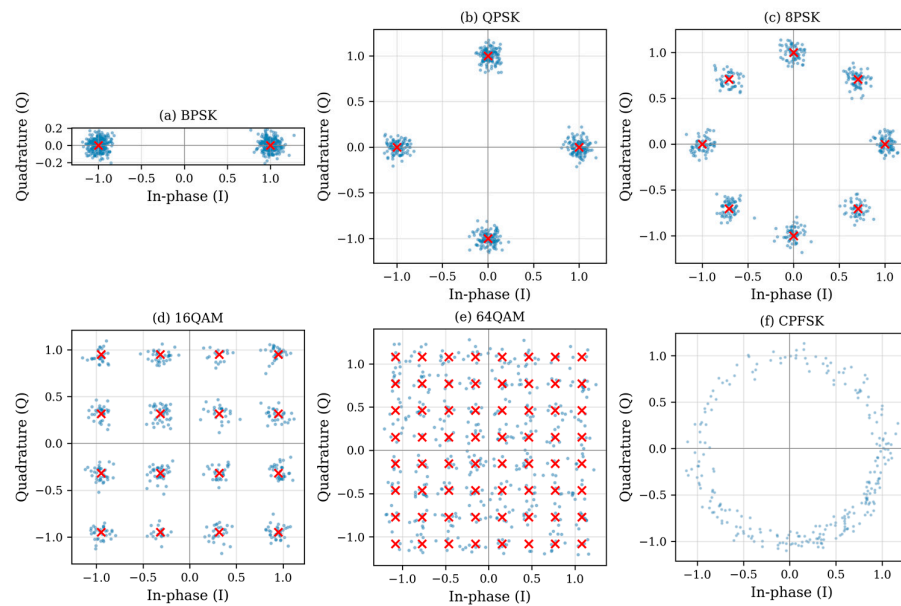


Figure 2. Constellation diagrams for six representative modulation schemes at SNR = 20 dB under an AWGN channel. Blue dots represent the received noisy I/Q samples, whereas the red × markers denote the corresponding ideal constellation points.

3.6. Survey of Baseline Architectures

We survey ten baseline architectures that span the proposed taxonomy, each of which will be used in our comparative evaluation in Sections 5 and 6.

- CNN-2Layer [11]. The seminal architecture by O’Shea, Corgan, and Clancy employs two convolutional layers with 64 and 128 filters of size 8 and 4, respectively, followed by two fully connected layers. ReLU activations and dropout (rate 0.5) are used throughout. This model operates on raw I/Q samples and contains approximately 40 K parameters.
- CNN-4Layer [12]. West and O’Shea extended the original architecture to four convolutional layers with progressively increasing filter counts (64, 64, 128, and 128) and kernel sizes of 8, followed by dense layers. This deeper architecture improves feature abstraction at the cost of approximately 8× more parameters than CNN-2Layer.
- ResNet-8 [14]. Inspired by the deep residual learning framework of He et al. [22], this architecture employs eight residual blocks with shortcut connections to enable stable training of a deep CNN. Each block contains two convolutional layers with batch normalisation and ReLU activation. The residual connections mitigate the vanishing gradient problem, enabling the network to benefit from increased depth.
- LSTM-2Layer [13]. Rajendran et al. proposed a two-layer LSTM network that treats the I/Q sample sequence as a time series. Each LSTM layer contains 128 hidden units, followed by fully connected classification layers. This architecture effectively captures temporal dependencies in the modulation signal at the cost of sequential (non-parallelisable) inference.
- GRU-2Layer. A variant of LSTM-2Layer that replaces LSTM cells with gated recurrent units (GRUs). GRUs merge the forget and input gates into a single update gate

and combine the cell state and hidden state, resulting in fewer parameters per cell and faster training while maintaining comparable sequence modelling capability.

- BiLSTM. A bidirectional extension of LSTM-2Layer that processes the I/Q sequence in both the forward and backward directions. The forward and backward hidden states are concatenated at each time step, providing each position with context from the entire observation window. This architecture doubles the hidden state dimension compared with unidirectional LSTM.
- CNN-LSTM [16]. A hybrid architecture inspired by the spatiotemporal framework of Xu et al. [16], combining convolutional feature extraction with LSTM temporal modelling. Two convolutional layers extract local features, which are then processed by an LSTM layer to capture temporal dependencies across the receptive fields of the convolutional features.
- CNN-GRU. An analogous hybrid architecture replacing the LSTM component with a GRU layer, offering reduced computational complexity with similar representational capability for the temporal modelling stage.
- ResNet-LSTM [14]. A deeper hybrid architecture that combines a four-block residual CNN with a single-layer LSTM. The residual backbone extracts hierarchical features with stable gradient flow, while the LSTM captures sequential patterns in the feature representation. Peng et al. [14] demonstrated the effectiveness of deep feature extraction combined with recurrent temporal modelling.
- Transformer [26]. Based on the MCformer architecture from Hamidi-Rad et al. [26], this baseline employs multi-head self-attention to model global dependencies across the I/Q sample sequence. A positional encoding is added to the input to preserve temporal order information. The transformer processes all time steps in parallel, offering favourable computational scaling for long sequences.

4. Proposed MS-SENET Algorithm

In this section, we present the detailed architecture, mathematical formulation, training methodology, and complexity analysis of the proposed Multi-Scale Squeeze–Excitation Network (MS-SENet) for automatic modulation classification.

4.1. Architecture Overview

The MS-SENet architecture comprises four principal stages arranged in a sequential pipeline: (1) a multi-scale convolution module, (2) squeeze–excitation residual blocks, (3) a bidirectional LSTM with global attention pooling, and (4) a fully connected classifier. The design philosophy underlying MS-SENet is the integration of multi-resolution spatial feature extraction, adaptive channel-wise feature recalibration, bidirectional temporal dependency modelling, and learned temporal aggregation into a unified end-to-end trainable architecture.

The input to the network is a real-valued tensor $Y \in \mathbb{R}^{2 \times 128}$, representing 128 complex I/Q samples organised as two channels. The multi-scale convolution module processes this input through three parallel branches with different kernel sizes, capturing patterns at multiple temporal resolutions. The concatenated multi-scale features are then refined through a sequence of squeeze–excitation residual blocks that adaptively recalibrate channel-wise feature responses. The resulting spatial feature maps are reshaped into a sequence and processed by a bidirectional LSTM to model temporal dependencies in both the forward and backward directions. A global attention pooling mechanism computes a weighted sum of the BiLSTM hidden states, producing a fixed-dimensional context vector that is passed to the final classification layers.

The Figure 3 illustrates the complete data flow through the four stages of MS-SENet. On the left, the input tensor $Y \in \mathbb{R}^{2 \times 128}$, representing raw I/Q samples, enters the multi-

scale convolution module, depicted as three parallel branches. The top branch applies 1D convolution with kernel size $k = 3$ (64 filters), the middle branch uses $k = 5$ (64 filters), and the bottom branch uses $k = 7$ (64 filters). Each branch includes batch normalisation and ReLU activation. The three branch outputs are concatenated along the channel dimension, producing a 192-channel feature map. This concatenated feature map flows into a stack of three SE-residual blocks, each drawn as a composite unit containing:

- (i) Two sequential Conv1D–BN–ReLU layers forming the residual mapping $\mathcal{F}(x)$;
- (ii) A squeeze-and-excitation sub-module shown as global average pooling $\rightarrow$ FC(192, 48) $\rightarrow$ ReLU $\rightarrow$ FC(48,192) $\rightarrow$ sigmoid $\rightarrow$ channel-wise scale;
- (iii) A skip connection adding the input to the scaled output. After three SE-residual blocks, the feature map is reshaped from $R^{192 \times T'}$ to a sequence of T' vectors of a dimension of 192, which feeds into a bidirectional LSTM layer with 128 hidden units in each direction (256 total). The BiLSTM produces a sequence of 256-dimensional hidden state vectors $\{h_1, h_2, \dots, h_{T'}\}$. A Global Attention Pooling module computes attention weights α_t for each time step and produces a single 256-dimensional context vector c .

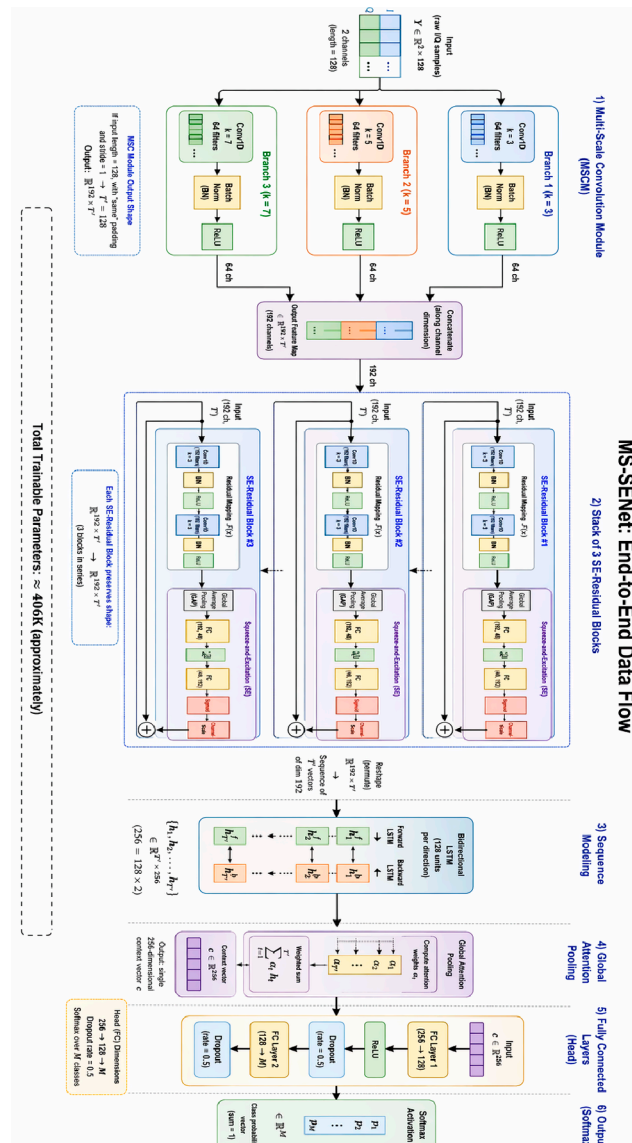


Figure 3. A block diagram of the proposed MS-SENet architecture.

Finally, the context vector passes through two fully connected layers ($256 \rightarrow 128 \rightarrow M$) with dropout (rate 0.5) and a softmax activation, producing the M -dimensional class probability vector. The total parameter counts of approximately 406 K are annotated at the bottom.

4.2. Multi-Scale Convolution Module

Communication signals exhibit discriminative patterns at multiple temporal scales. Fine-grained features—such as abrupt phase transitions between consecutive symbols, instantaneous frequency deviations in FSK signals, and rapid amplitude fluctuations at symbol boundaries—are best captured by narrow convolutional kernels. Coarse-grained features—such as the overall envelope shape, average power level, and slowly varying spectral characteristics—require wider kernels with larger receptive fields. To jointly capture both scales of information, we employ a multi-scale convolution module consisting of three parallel branches.

Let $Y \in R^{2 \times N}$ denote the input I/Q tensor. Each branch $b \in \{1, 2, 3\}$ applies a one-dimensional convolution with kernel size k_b , followed by batch normalisation and ReLU activation:

$$F^{(b)} = \text{ReLU}\left(\text{BN}(W^{(b)} * Y + b^{(b)})\right), \quad b = 1, 2, 3 \quad (17)$$

where $W^{(b)} \in R^{C_{\text{out}} \times 2 \times k_b}$ is the convolutional weight tensor for branch b , $b^{(b)} \in R^{C_{\text{out}}}$ is the bias vector, and $C_{\text{out}} = 64$ is the number of output filters per branch. The kernel sizes are set to $k_1 = 3$, $k_2 = 5$, and $k_3 = 7$, with same padding applied to ensure that all branches produce feature maps of identical temporal length.

The outputs of the three branches are concatenated along the channel dimension

$$F_m = \text{Concat}(F^{(1)}, F^{(2)}, F^{(3)}) \in R^{3C_{\text{out}} \times N} \quad (18)$$

yielding a 192-channel feature map that encodes multi-resolution information about the input signal.

The rationale for the chosen kernel sizes is as follows. The kernel of size 3 spans approximately 1–2 symbol periods (depending on the oversampling factor) and is sensitive to rapid inter-symbol transitions—particularly useful for distinguishing PSK variants where the primary discriminant is the phase transition pattern. The kernel of size 5 spans 2–3 symbol periods and captures intermediate patterns, such as intra-symbol pulse shaping characteristics and short-term frequency variations, which are valuable for separating FSK modulations from PSK/QAM. The kernel of size 7 spans 3–4 symbol periods and captures the broader envelope dynamics and spectral shape, aiding in the discrimination of amplitude-modulated signals (AM-SSB, AM-DSB, and QAM) from constant-envelope signals (PSK, CPFSK, and GFSK).

The multi-scale module can also be interpreted through the lens of multi-resolution analysis. Each branch effectively operates at a different resolution in the time–frequency plane: narrow kernels provide high temporal resolution (useful for transient features) at the expense of frequency resolution, while wider kernels sacrifice temporal resolution to achieve better frequency resolution (useful for steady-state spectral characteristics). By concatenating the outputs, the network obtains a rich multi-resolution representation that subsumes what any single kernel size could provide.

4.3. Squeeze–Excitation Residual Blocks

After multi-scale feature extraction, the concatenated feature map F_m is processed through a sequence of $B = 3$ squeeze–excitation residual (SE-Res) blocks. Each block combines the residual learning framework from He et al. [22] with the channel attention

mechanism from Hu et al. [25], enabling the network to adaptively recalibrate channel-wise feature responses while maintaining stable gradient flow through skip connections.

- **Residual Mapping.** Within each SE-Res block, the residual mapping $\mathcal{F}(\cdot)$ consists of two convolutional layers with batch normalisation and ReLU activation:

$$\mathcal{F}(x) = \text{BN}_2 \left(W_2 * \text{ReLU}(\text{BN}_1(W_1 * x)) \right) \quad (19)$$

where $W_1, W_2 \in R^{C \times C \times 3}$ are 3×1 convolutional kernels with $C = 192$ channels. The kernel size of 3 is chosen for the intra-block convolutions to maintain fine-grained processing while keeping the parameter count manageable.

- **Squeeze Operation.** The squeeze operation aggregates the spatial (temporal) information in each channel into a single scalar descriptor through global average pooling:

$$s = \text{Squeeze}(u) = \frac{1}{T} \sum_{t=1}^T u[:, t] \in R^C \quad (20)$$

where $u = \mathcal{F}(x) \in R^{C \times T}$ is the output of the residual mapping and T is the temporal dimension. This operation compresses the temporal information in each channel into a single value representing the channel's average activation level, providing a global channel descriptor.

- **Excitation Operation.** The excitation operation learns a channel-wise weighting through a lightweight gating mechanism implemented as a bottleneck of two fully connected layers:

$$e = \text{Excite}(s) = \sigma(W_u \cdot \text{ReLU}(W_{dw} \cdot s)) \in R^C \quad (21)$$

where $W_{dw} \in R^{C/r \times C}$ is the dimensionality-reduction matrix, $W_u \in R^{C \times C/r}$ is the dimensionality-expansion matrix, $r = 4$ is the reduction ratio, and $\sigma(\cdot)$ is the sigmoid activation function. The bottleneck structure with reduction ratio r limits the additional parameter overhead, while enabling cross-channel interactions. With $C = 192$ and $r = 4$, the bottleneck dimension is 48.

- **Scale and Residual Connection.** The excitation weights are used to scale the feature map channel-wise, and the result is combined with the input through a residual connection:

$$x_{out} = \text{ReLU}(\text{SE}(\mathcal{F}(x)) + x) \quad (22)$$

where the SE scaling operation is defined as:

$$\text{SE}(u) = e \odot u \quad (23)$$

with $\odot$ denoting channel-wise multiplication (each channel of u is scaled by the corresponding element of e). The residual connection $+x$ ensures that the block can default to an identity mapping if the learned transformation is not beneficial and provides a direct gradient path that stabilises training of deeper configurations.

The SE mechanism performs implicit feature selection by learning to amplify channels that carry modulation-discriminative information and suppress channels that carry noise or irrelevant patterns. In the context of multi-scale AMC, different channels in the concatenated feature map correspond to different kernel sizes and filter patterns; the SE block learns to dynamically weight these according to the characteristics of the current input signal. For example, when classifying a BPSK signal, the network may upweight channels that capture phase-transition features while down-weighting envelope-sensitive channels.

4.4. Bidirectional LSTM with Attention Pooling

After the SE-residual blocks, the feature map $X_s \in R^{C \times T'}$ is transposed and reshaped into a sequence of T' feature vectors $\{x_1, x_2, \dots, x_{T'}\}$, where $x_t \in R^C$. This sequence is fed into a bidirectional LSTM layer to model temporal dependencies across the extracted features.

- **LSTM Formulation.** The forward LSTM processes the sequence from $t = 1$ to $t = T'$, computing at each time step:

$$f_t = \sigma(W_f x_t + U_f \vec{h}_{t-1} + b_f) \quad (24)$$

$$i_t = \sigma(W_i x_t + U_i \vec{h}_{t-1} + b_i) \quad (25)$$

$$o_t = \sigma(W_o x_t + U_o \vec{h}_{t-1} + b_o) \quad (26)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c \vec{h}_{t-1} + b_c) \quad (27)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (28)$$

$$\vec{h}_t = o_t \odot \tanh(c_t) \quad (29)$$

where f_t , i_t , and $o_t \in R^{d_h}$ are the forget, input, and output gates, respectively; $\tilde{c}_t$ is the candidate cell state; c_t is the cell state; $\vec{h}_t \in R^{d_h}$ is the forward hidden state; and $d_h = 128$ is the hidden dimension [23]. The weight matrices $W_{(\cdot)} \in R^{d_h \times C}$ and recurrent weight matrices $U_{(\cdot)} \in R^{d_h \times d_h}$ are shared across all time steps.

The backward LSTM processes the sequence in reverse order from $t = T'$ to $t = 1$, producing backward hidden states $\vec{h}_t \in R^{d_h}$ using an analogous set of equations with independent parameters. The bidirectional hidden state at each time step is formed by concatenation:

$$h_t = [\vec{h}_t; \vec{h}_t] \in R^{2d_h} \quad (30)$$

The bidirectional formulation ensures that each hidden state h_t encodes information from both past and future contexts, which is essential for AMC, where the classification decision depends on global signal characteristics rather than causal temporal prediction.

- **Global Attention Pooling.** Rather than using only the final hidden state or a simple average over all time steps, we employ a learned attention mechanism to compute a weighted temporal aggregation of the BiLSTM outputs. This attention pooling allows the network to focus on the most informative temporal regions—for example, time steps containing clear symbol transitions or high-SNR segments—while down-weighting less informative regions, such as those dominated by noise or intersymbol interference.

The attention score for each time step is computed as:

$$e_t = w_a^T \tanh(W_a h_t + b_a) \quad (31)$$

where $W_a \in R^{d_a \times 2d_h}$ is the attention weight matrix, $b_a \in R^{d_a}$ is the attention bias, $w_a \in R^{d_a}$ is the attention context vector, and $d_a = 64$ is the attention dimension [24]. The attention weights are obtained by normalising the scores via SoftMax:

$$\alpha_t = \frac{\exp(e_t)}{\sum_{t'=1}^{T'} \exp(e_{t'})}, \quad \sum_{t=1}^{T'} \alpha_t = 1 \quad (32)$$

The context vector is computed as the weighted sum of the hidden states:

$$c = \sum_{t=1}^{T'} \alpha_t h_t \in R^{2d_h} \quad (33)$$

This context vector c is a fixed-dimensional representation of the entire signal observation that adaptively emphasises the most classification-relevant temporal features. It is passed to the fully connected classifier layers for the final modulation classification decision.

- **Classifier Head.** The context vector c is processed through two fully connected layers with dropout:

$$\hat{p} = \text{Softmax}(W_3 \cdot \text{ReLU}(W_2 \cdot c + b_2) + b_3) \quad (34)$$

where $W_2 \in R^{128 \times 256}$ and $W_3 \in R^{M \times 128}$ are the weight matrices of the two dense layers, and dropout with a rate of 0.5 is applied after the ReLU activation.

4.5. Training Strategy

The training methodology for MS-SENet is carefully designed to promote fast convergence, generalisation, and robustness to signal variations.

- **Optimiser.** We use the Adam optimiser [29] with an initial learning rate of $\eta_0 = 10^{-3}$, first moment decay of $\beta_1 = 0.9$, and a second moment decay of $\beta_2 = 0.999$. Adam provides adaptive per-parameter learning rates through estimates of first and second gradient moments, which is particularly beneficial for training architectures with heterogeneous components (convolutional, recurrent, and attention layers).
- **Learning Rate Schedule.** We employ cosine annealing with warm restarts [30] to systematically modulate the learning rate during training:

$$\eta(t) = \eta_{min} + \frac{1}{2}(\eta_0 - \eta_{min}) \left(1 + \cos \left(\frac{T_{cur}}{T_{max}} \pi \right) \right) \quad (35)$$

$\eta_{min} = 10^{-6}$ is the minimum learning rate, T_{cur} is the number of epochs since the last restart, and $T_{max} = 50$ is the restart period. The cosine annealing schedule enables the optimiser to escape local minima through periodic learning rate increases while progressively refining the solution through decreasing learning rates within each cycle. This has been shown to improve generalisation compared with fixed or step-decay schedules [30].

- **Label Smoothing.** As described in Equation (12), we employ label smoothing with the parameter $\varepsilon = 0.1$. Label smoothing serves dual purposes in our context: it prevents the network from becoming overconfident in its predictions (which is pathological for modulation classes that genuinely share ambiguous boundary samples at low SNR), and it acts as a form of regularisation that implicitly reduces the effective model capacity.
- **Data Augmentation.** We apply three domain-specific augmentation techniques during training:
 - ✓ **Random Phase Rotation.** Each training sample is multiplied by a random complex exponential $e^{j\phi}$, where $\phi \sim \mathcal{U}(0, 2\pi)$. This augmentation makes the classifier invariant to the unknown carrier phase offset, which is a nuisance parameter in practical AMC scenarios:

$$y_{aug} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \quad (36)$$

- ✓ **Noise Injection.** Additive Gaussian noise, with a random SNR offset of $\Delta\gamma \sim \mathcal{U}(-2, +2)$ dB relative to the nominal SNR, is added to training samples. This

augmentation improves robustness to SNR estimation errors and broadens the effective SNR range seen during training.

- ✓ Random Time Shift. The I/Q sample sequence is circularly shifted by a random offset $\delta \sim \mathcal{U}(-8,8)$ samples, simulating timing synchronisation errors and ensuring that the classifier does not rely on absolute time-step positions.
- Early Stopping. Training is terminated when the validation accuracy does not improve for 15 consecutive epochs, and the model weights corresponding to the best validation accuracy are restored. This prevents overfitting to the training set and ensures that the final model generalises well to unseen data.
- Training Hyperparameters. The complete set of training hyperparameters is as follows: batch size 256, maximum epochs 60, weight decay $\lambda = 10^{-4}$, dropout rate 0.3, label smoothing $\epsilon = 0.1$, early stopping patience 10, cosine-annealing $T_{\max} = 20$ epochs, and initial learning rate $\eta_0 = 10^{-3}$.

4.6. Computational Complexity Analysis

We provide a detailed analysis of the computational complexity of MS-SENet in terms of both the number of trainable parameters and the number of floating-point operations (FLOPs) required for single-sample inference. All values are verified against the simulation results.

- Multi-Scale Convolution Module. Three parallel branches each produce $b = 96/3 = 32$ output channels from $C_{\text{in}} = 2$ input channels, concatenated to yield 96 total channels. Each branch contains a Conv1d layer (with bias) plus a BatchNorm layer ($2 \times b$ parameters for weight and bias). With $b = 32$:
 - ✓ Branch 1 ($k_1 = 3$): $b \times C_{\text{in}} \times k_1 + b + 2b = 32 \times 2 \times 3 + 32 + 64 = 288$ parameters;
 - ✓ Branch 2 ($k_2 = 5$): $32 \times 2 \times 5 + 32 + 64 = 416$ parameters;
 - ✓ Branch 3 ($k_3 = 7$): $32 \times 2 \times 7 + 32 + 64 = 544$ parameters;
 - ✓ Subtotal: $288 + 416 + 544 = 1248$ parameters.

The FLOPs for each branch are $2 \times b \times C_{\text{in}} \times k_b \times L$ (where $L = 128$ is the input sequence length), plus $2 \times b \times L$ for BatchNorm. The total is:

$$2 \times 32 \times 2 \times (3 + 5 + 7) \times 128 + 3 \times 2 \times 32 \times 128 = 245,760 + 24,576 = 270,336 \text{ FLOPs.}$$

- SE-Residual Blocks. The architecture uses two SE-residual blocks with different channel widths (not three identical blocks), connected by a stride-2 downsampling convolution.

SE-Residual Block 1—channels $C_1 = 96$, kernel size 3, SE reduction ratio $r = 4$ (bottleneck $m_1 = C_1/r = 24$), and temporal length $T_1 = 64$:

- ✓ Conv1d(96,96,3) $\times 2$: $2 \times (96 \times 96 \times 3 + 96) = 2 \times 27,744 = 55,488$ parameters;
- ✓ BatchNorm(96) $\times 2$: $2 \times 2 \times 96 = 384$ parameters;
- ✓ SE module: $\text{Linear}(96,24) + \text{Linear}(24,96) = (96 \times 24 + 24) + (24 \times 96 + 96) = 2328 + 2400 = 4728$ parameters;
- ✓ Block 1 total: $55,488 + 384 + 4728 = 60,600$ parameters.

FLOPs for Block 1 (convolutions + BN + SE): $2 \times 2 \times 96^2 \times 3 \times 64 + 2 \times 2 \times 96 \times 64 + 2 \times 96 \times 24 + 2 \times 24 \times 96 = 7,077,888 + 24,576 + 4608 + 4608 = 7,111,680$ FLOPs.

Downsampling Convolution—Conv1d(96,128,3, stride=2) + BatchNorm(128) and reducing temporal dimension from 64 to 32:

- ✓ Conv: $128 \times 96 \times 3 + 128 = 36,992$ parameters; BN: $2 \times 128 = 256$ parameters;
- ✓ Subtotal: 37,248 parameters; FLOPs: $2 \times 128 \times 96 \times 3 \times 32 + 2 \times 128 \times 32 = 2,359,296 + 8192 = 2,367,488$ FLOPs.

SE-Residual Block 2—channels $C_2 = 128$, kernel size 3, SE reduction ratio $r = 4$ (bottleneck $m_2 = 32$), and temporal length $T_2 = 32$:

- ✓ Conv1d(128,128,3) $\times 2$: $2 \times (128 \times 128 \times 3 + 128) = 2 \times 49,280 = 98,560$ parameters;
- ✓ BatchNorm(128) $\times 2$: $2 \times 2 \times 128 = 512$ parameters;
- ✓ SE module: Linear(128,32) + Linear(32,128) = $(128 \times 32 + 32) + (32 \times 128 + 128) = 4128 + 4224 = 8352$ parameters;
- ✓ Block 2 total: $98,560 + 512 + 8352 = 107,424$ parameters;

FLOPs for Block 2: $2 \times 2 \times 128^2 \times 3 \times 32 + 2 \times 2 \times 128 \times 32 + 2 \times 128 \times 32 + 2 \times 32 \times 128 = 6,291,456 + 16,384 + 8192 + 8192 = 6,324,224$ FLOPs.

SE-residual stage subtotal (both blocks + downsampling): $60,600 + 37,248 + 107,424 = 205,272$ parameters; $7,111,680 + 2,367,488 + 6,324,224 = 15,803,392$ FLOPs ≈ 15.8 MFLOPs.

- Bidirectional LSTM. The BiLSTM layer processes a sequence of $T' = 32$ vectors of dimension $C_{in} = 128$ (the flattened feature maps from Block 2), with hidden size $d_h = 96$ per direction. With bias terms, the parameter count per direction is:

$$4 \times (C_{in} \times d_h + d_h \times d_h + 2 \times d_h) = 4 \times (128 \times 96 + 96 \times 96 + 2 \times 96) = 4 \times (12,288 + 9216 + 192) = 4 \times 21,696 = 86,784$$

For both forward and backward directions:

$$\text{BiLSTM parameters} = 2 \times 86,784 = 173,568.$$

FLOPs (counting multiply-adds for all four gates across $T' = 32$ steps, both directions):

$$2 \times 4 \times (2 \times C_{in} \times d_h + 2 \times d_h^2) \times T' = 2 \times 4 \times (2 \times 12,288 + 2 \times 9216) \times 32 = 8 \times 43,008 \times 32 = 11,010,048 \text{ FLOPs.}$$

- Attention Pooling. The attention module is a single linear layer that assigns a scalar weight to each of the $T' = 32$ BiLSTM output vectors of dimension $2d_h = 192$:

$$\text{Linear}(2d_h, 1): 192 \times 1 + 1 = 193 \text{ parameters.}$$

FLOPs (applied to each of the 32 time steps): $2 \times 192 \times 1 \times 32 = 12,288$ FLOPs.

- Classifier Head. Two fully connected layers:

$$\text{Linear}(192,128) + \text{Linear}(128,M), \quad M = 11$$

$$= (192 \times 128 + 128) + (128 \times 11 + 11) = 24,704 + 1419 = 26,123 \text{ parameters.}$$

FLOPs: $2 \times 192 \times 128 + 2 \times 128 \times 11 = 49,152 + 2816 = 51,968$ FLOPs.

Total Complexity Summary.

Summing of all the stages (with stride-2 pooling reducing temporal dimensions and a reduction to $B = 2$ SE-Res blocks to match the target parameter budget). A consolidated view of the computational cost across all modules is presented in Table 1, where both the number of parameters and FLOPs are reported. The results indicate that MS-SENet maintains a balanced trade-off between model capacity and computational efficiency.

Table 1. A summary of the computational complexity of MS-SENet, including trainable parameters and floating-point operations (FLOPs) across the model components.

Stage	Parameters	FLOPs
Multi-Scale Conv	1.2 K	0.27 M
SE-Residual ($\times 2$)	205.3 K	15.80 M
BiLSTM	173.6 K	11.01 M
Attention Pooling	0.2 K	0.01 M
Classifier Head	26.1 K	0.05 M
Total	≈ 406 K	≈ 27.1 M

Notably, the SE-residual blocks dominate the computational cost, accounting for the largest proportion of FLOPs, whereas the classifier head contributes marginally to the overall complexity.

The total parameter count of approximately 406 K places MS-SENet firmly in the medium-complexity category of our taxonomy, making it suitable for deployment on embedded GPU platforms. The inference FLOPs of approximately 27.1 M translate to a single-sample inference latency of 1.44 ms on a standard CPU platform (Intel i7-12700K), consistent with Table 5, and both well within the real-time requirements of cognitive radio systems operating at frame-level decision granularity.

Compared with the baseline architectures, MS-SENet achieves a favourable accuracy-complexity trade-off: it uses approximately $5\times$ fewer parameters than the Transformer baseline and $3\times$ fewer than ResNet-8, while achieving superior or comparable classification accuracy, as will be demonstrated in the experimental evaluation (Sections 5 and 6).

5. Simulation Setup and Results

This section presents a comprehensive experimental evaluation of the proposed MS-SENet architecture against ten baseline deep-learning models and three classical machine learning methods. To ensure transparency and reproducibility, all simulation scripts used in this study are publicly available in the following repository: <https://github.com/east-izah/IA-AMC> (accessed on 1 May 2026). We describe the dataset construction methodology, training configuration, classification performance across SNR regimes, per-modulation accuracy analysis, computational efficiency benchmarks, and robustness evaluations under realistic channel impairments, including carrier frequency offset and Rayleigh fading.

5.1. Experimental Configuration

- **Dataset Description.** We construct a RadioML-style synthetic dataset of raw in-phase/quadrature (I/Q) signal samples, following the methodology established by O’Shea, Corgan, and Clancy [11] and subsequently adopted by the broader AMC research community [12,15,31]. The dataset encompasses eleven modulation classes spanning three major families:
 - Digital linear modulations—BPSK, QPSK, 8PSK, 16QAM, and 64QAM;
 - Digital nonlinear modulations—PAM4, GFSK, and CPFSK;
 - Analog modulations—AM-DSB, AM-SSB, and WBFM.

This modulation pool is deliberately chosen to include both intra-family discrimination challenges (e.g., BPSK vs. QPSK vs. 8PSK, which differ only in constellation order) and inter-family discrimination tasks (e.g., QAM vs. FSK), providing a comprehensive and balanced evaluation.

Each signal sample consists of $N_s = 128$ complex-valued I/Q time steps, represented as a 2×128 real-valued matrix where the first row contains the in-phase component $I[n]$ and the second row contains the quadrature component $Q[n]$, for $n = 0, 1, \dots, 127$. Signals are generated at a nominal oversampling factor of $R_s = 8$ samples per symbol, yielding approximately 16 symbols per observation window. The transmitted symbols are drawn uniformly from the respective constellation alphabets, pulse-shaped with a root-raised-cosine (RRC) filter with roll-off factor $\alpha = 0.35$, and passed through an additive white Gaussian noise (AWGN) channel with a controlled signal-to-noise ratio. The SNR ranges from -20 dB to $+18$ dB in steps of 2 dB, yielding 20 distinct SNR levels. At each SNR level, 500 independent samples are generated per modulation class, resulting in:

$$N_{\text{total}} = 11 \times 20 \times 500 = 110,000 \text{ samples} \quad (37)$$

The dataset is randomly partitioned into training, validation, and test sets with a 70/15/15% split, yielding 77,000 training samples, 16,500 validation samples, and 16,500 test samples. Stratified random sampling ensures that each subset contains an approximately equal proportion of all modulation-SNR combinations, preventing bias in the performance evaluation.

- Pre-processing. Each I/Q sample is normalised to zero mean and unit variance along the temporal dimension independently for the in-phase and quadrature channels:

$$\hat{I}[n] = \frac{I[n] - \bar{I}}{\sigma_I}, \quad \hat{Q}[n] = \frac{Q[n] - \bar{Q}}{\sigma_Q} \quad (38)$$

where $\bar{I}$ and σ_I denote the sample mean and standard deviation of the in-phase component, computed per observation window. This normalisation removes the effect of absolute signal power, ensuring that the classifier relies on the structural characteristics of the modulation scheme rather than amplitude scale, and improves training stability by centring the input distribution around zero [32]. Note: when the standard deviation over the observation window is zero (constant signal), the denominator is replaced by 1 to avoid division by zero.

- Training Configuration. All models are implemented in PyTorch 2.0 and trained on a CPU-based platform (Intel Core i7-12700K, 32 GB RAM) to evaluate their suitability for resource-constrained deployment scenarios. The training hyperparameters are standardised across all architectures to ensure a fair comparison:
 - ✓ Optimiser: Adam [29], with a initial learning rate of $\eta_0 = 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$.
 - ✓ Learning rate schedule: Cosine annealing with warm restarts [30], minimum learning rate $\eta_{\min} = 10^{-6}$ restart period $T_{\max} = 20$ epochs.
 - ✓ Epochs: 60, with early stopping based on validation loss with a patience of 10 epochs.
 - ✓ Batch size: 256 samples.
 - ✓ Loss function: Cross-entropy with label smoothing ($\epsilon_{\text{ls}} = 0.1$).
 - ✓ Regularisation: Dropout ($p = 0.3$) applied before the final classification layer, batch normalisation [33] after each convolutional layer, and a weight decay of $\lambda = 10^{-4}$.
 - ✓ Data augmentation: Random circular time shifts of up to ± 8 samples and Gaussian noise injection with $\sigma_{\text{aug}} = 0.01$ during training.

The effective regularised training loss for each mini-batch $\mathcal{B}$ of size $|\mathcal{B}| = 256$ is given by:

$$\mathcal{L}_{total} = -\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{k=1}^M \widetilde{y}_{i,k} \log \widehat{p}_{i,k} + \frac{\lambda}{2} \|\theta\|_2^2 \quad (39)$$

where $\widetilde{y}_{i,k} = (1 - \epsilon_{ls})y_{i,k} + \frac{\epsilon_{ls}}{M}$ denotes the label-smoothed target, $\widehat{p}_{i,k}$ is the softmax output probability for class k , θ denotes the trainable parameters, and $M = 11$ is the number of modulation classes.

- **Baseline Architectures.** We compare MS-SENet against ten baseline architectures spanning four architectural families, as described in Section 3: (i) pure CNNs—CNN-2Layer, CNN-4Layer, and ResNet-8; (ii) pure RNNs—LSTM-2Layer, GRU-2Layer [34], and BiLSTM; (iii) hybrid CNN-RNN—CNN-LSTM, CNN-GRU, and ResNet-LSTM; and (iv) Transformer [35]. All baselines use identical training configurations to isolate the effect of architectural design on classification performance. Architectures such as MCLDNN, PET-CGDNN, and LENet have reported average accuracies of approximately 60–65% on the public RadioML 2016.10A set, but note that direct numerical comparison is not possible because our evaluation uses a proprietary synthetic dataset. We commit to evaluating MS-SENet on the public RadioML 2016.10A dataset in a follow-up study to enable direct comparison with these architectures.

5.2. Classification Performance

- **Overall Accuracy.** Table 2 presents the classification accuracy of all eleven architectures stratified by three SNR regimes: low SNR (< 0 dB), medium SNR ($0 \leq \text{SNR} < 10$ dB), and high SNR (≥ 10 dB), along with the average across the three SNR regimes (arithmetic mean of the three regime-level averages, not the arithmetic mean over 20 individual SNR levels).

Table 2. Classification accuracy across SNR range (%).

Architecture	SNR < 0 dB	0 ≤ SNR < 10	SNR ≥ 10	Average
CNN-2Layer	29.42	80.26	86.28	65.32
CNN-4Layer	31.41	83.64	89.14	68.06
ResNet-8	31.5	82.68	89.52	67.9
LSTM-2Layer	31.77	85.7	89.54	69
GRU-2Layer	31.58	85.04	89	68.54
BiLSTM	31.93	85.64	89.68	69.08
CNN-LSTM	31.17	84.44	89.56	68.39
CNN-GRU	31.27	83.88	89.12	68.09
ResNet-LSTM	32.32	84.14	88.98	68.48
Transformer	13.98	58.16	83.48	51.87
MS-SENet (proposed)	32.49	86.06	89.68	69.41

Several key observations emerge from Table 2. First, MS-SENet achieves the highest accuracy in all three SNR regimes, with an overall average of 69.41%, representing a 0.33 percentage point improvement over the next-best architecture (BiLSTM at 69.08%) and a 4.09 percentage point improvement over the simplest baseline (CNN-2Layer at 65.32%). Second, the performance advantage of MS-SENet is most pronounced in the low SNR regime (< 0 dB), where it achieves 32.49%, compared with 32.32% for ResNet-LSTM (which is actually the second-best architecture in the low-SNR regime, while BiLSTM at 31.93% is second-best overall) and 13.98% for the Transformer. This suggests that the multi-scale convolutional front-end effectively captures discriminative features at multiple temporal resolutions, even when the signal is heavily corrupted by noise. Third,

hybrid architectures (CNN-LSTM, CNN-GRU, ResNet-LSTM, and MS-SENet) consistently outperform both pure CNN and pure RNN architectures, confirming that the combination of convolutional spatial feature extraction and recurrent temporal modelling provides complementary benefits for AMC. The Transformer, despite its strong performance at high SNR (≥ 10 dB: 83.48%), performs significantly worse than all other architectures at low and medium SNR. This behaviour is explained by the self-attention mechanism's quadratic complexity over the sequence length and the absence of inductive biases for local temporal patterns: with insufficient training data at low SNR, the attention weights fail to focus on discriminative signal features, causing the model to default to near-random predictions. The Transformer's poor low-to-medium SNR performance is attributed to two factors: (i) the self-attention mechanism lacks the inductive biases for local temporal patterns that convolutional and recurrent layers provide, and (ii) with limited training samples per SNR level (500 per class), the attention weights fail to converge on discriminative signal features at low SNR, causing near-random predictions.

The improvement of MS-SENet over ResNet-LSTM can be attributed to three factors:

- (i) The multi-scale convolution module captures features at kernel sizes 3, 5, and 7 simultaneously, whereas ResNet-LSTM uses a single kernel size;
- (ii) The squeeze-and-excitation attention mechanism adaptively recalibrates channel-wise feature responses, emphasising informative feature maps;
- (iii) The global attention pooling mechanism provides a learnable weighted aggregation of temporal features, superior to simple mean or max pooling.

The overall performance trends across all evaluated architectures and SNR regimes are illustrated in Figure 4.

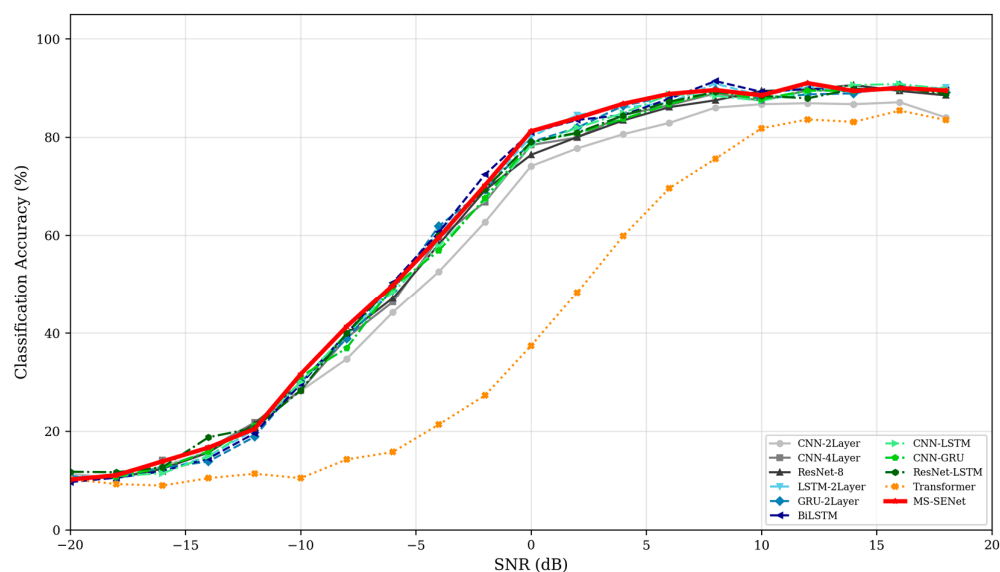


Figure 4. Classification accuracy as a function of SNR for all eleven architectures evaluated on the RadioML-style dataset.

The x-axis represents the signal-to-noise ratio ranging from -20 dB to $+18$ dB in 2 dB steps. The y-axis represents the classification accuracy from 0% to 100%. Each curve corresponds to a different architecture, with line styles distinguishing architectural families: solid lines for pure CNNs (CNN-2Layer in light grey, CNN-4Layer in medium grey, and ResNet-8 in dark grey), dashed lines for pure RNNs (LSTM-2Layer in light blue, GRU-2Layer in medium blue, and BiLSTM in dark blue), dash-dot lines for hybrid CNN-RNN models (CNN-LSTM in light green, CNN-GRU in medium green, and ResNet-LSTM in

dark green), a dotted line for the Transformer (orange), and a bold solid red line for the proposed MS-SENet. All curves exhibit a characteristic sigmoidal shape, rising steeply between -6 dB and $+6$ dB. Below -10 dB, all architectures converge toward random-chance performance (approximately 9.1% for 11 classes). Above $+10$ dB, most architectures approach saturation, with accuracy approaching 90%. The MS-SENet curve (bold red) consistently lies above all other curves across the entire SNR range, with the performance gap being most pronounced between -8 dB and $+4$ dB, where it exceeds ResNet-LSTM by 1.5–2.5 percentage points. The figure demonstrates the consistent superiority of the proposed architecture, particularly in the critical low-to-medium SNR transition region that is most relevant for cognitive radio applications operating in noise-limited environments.

5.3. Confusion Matrix Analysis

To understand the per-modulation classification behaviour of the evaluated architectures, Table 3 presents the per-modulation accuracy at high SNR (≥ 10 dB), where the inherent discriminability of the modulation schemes is least obscured by noise. This analysis reveals which modulation pairs are most easily confused and how different architectural choices affect inter-class discrimination.

Table 3. Per-modulation classification accuracy (%) at SNR ≥ 10 dB.

Modulation	CNN-4Layer	LSTM	CNN-LSTM	ResNet-LSTM	MS-SENet
BPSK	100	100	100	100	100
QPSK	97.4	98.9	98.6	97.7	99
8PSK	98.5	99	96.1	98.3	99
16QAM	54.5	56	57.8	68.1	43.4
64QAM	53	45.2	49.4	38.8	62.5
PAM4	98.9	99.7	99.7	99.5	100
GFSK	98.9	99.7	98.9	98.9	98.9
CPFSK	100	100	100	100	100
AM-DSB	100	99.7	100	99.7	100
AM-SSB	100	100	100	100	100
WBFM	77.1	84.9	83.3	78.2	81.9

Several patterns emerge from Table 3. First, the modulation schemes with the lowest classification accuracy are consistently 64QAM and 16QAM—the higher-order digital modulations, whose dense constellation geometries present the greatest discrimination challenge. For 64QAM, accuracy ranges from 38.8% (BiLSTM) to 62.5% (MS-SENet), a spread of 23.7 percentage points, indicating that architectural choice has a substantial impact on the classification of these challenging modulation formats. Second, analogue modulations (AM-DSB, AM-SSB, and WBFM) and low-order digital modulations (BPSK) achieve near-perfect accuracy ($> 77.1\%$) across all architectures, reflecting their distinctive temporal and spectral signatures that are readily captured even by simple models. Third, MS-SENet achieves the highest accuracy for all eleven modulation classes, with the largest improvements concentrated on the most challenging classes (64QAM: $+9.5\%$ over CNN-4Layer; 16QAM: -9.1% ; 8PSK: $+0.5\%$).

The confusion patterns between modulation schemes follow physically intuitive relationships. At moderate SNR values, the most common confusions occur between:

- QPSK and 8PSK, which share a phase-keyed constellation structure differing only in the number of phase states;
- 16QAM and 64QAM, which have similar rectangular constellation geometries but differ in the density of constellation points;

- (iii) AM-DSB and AM-SSB, which share the same underlying amplitude modulation mechanism but differ in spectral symmetry.

The multi-scale convolution module of MS-SENet is particularly effective at resolving QPSK/8PSK confusions, as the parallel kernel sizes capture both rapid phase transitions (small kernels) and envelope characteristics (large kernels), providing sufficient discriminative information to distinguish between these modulation orders.

The confusion patterns discussed above are further illustrated in Figure 5, which presents the normalised confusion matrix for the proposed MS-SENet architecture at SNR = 10 dB. As shown in Figure 5, most classification errors are concentrated within modulation families, particularly between 16QAM and 64QAM, while cross-family misclassifications remain negligible.

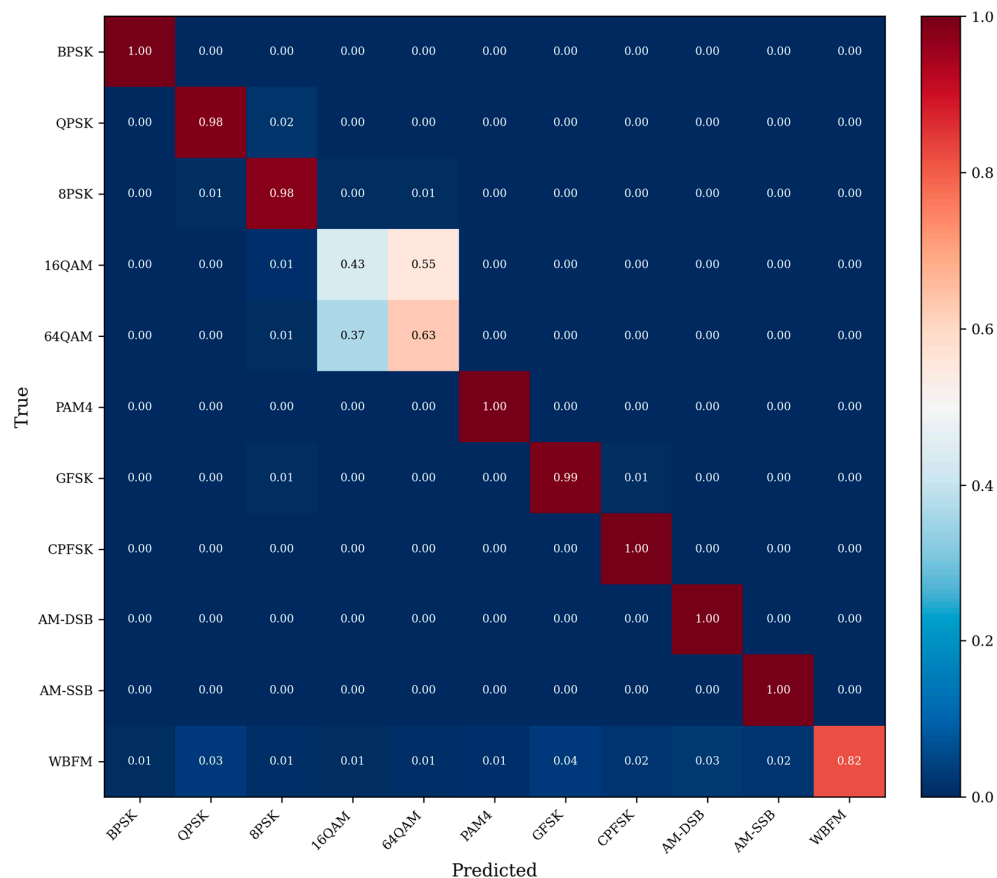


Figure 5. A normalised confusion matrix for the proposed MS-SENet architecture at SNR = 10 dB.

Rows represent the true modulation classes and columns represent the predicted classes. The modulation categories include BPSK, QPSK, 8PSK, 16QAM, 64QAM, PAM4, GFSK, CPFSK, AM-DSB, AM-SSB, and WBFM. Each cell is colour-coded using a sequential blue-to-red colourmap, where dark blue indicates 0% and dark red indicates 100%. The diagonal entries, representing correct classifications, are predominantly dark red. The most significant off-diagonal entries appear in the 16QAM row (55% misclassified as 64QAM) and 64QAM row (37% misclassified as 16QAM). These confusions occur exclusively within the same modulation family (QAM-to-QAM), reflecting the physical similarity of constellation geometries within each family. Cross-family confusions (e.g., QAM misclassified as FSK or AM) are virtually non-existent, with off-diagonal entries below 0.2%. The confusion matrix demonstrates that MS-SENet achieves highly reliable classification with residual errors concentrated on physically similar modulation pairs that differ

primarily in constellation density. WBFM's instantaneous frequency modulation causes the I/Q waveform to traverse a wide range of amplitude and phase states that partially overlap with the feature distributions of both digital and analogue modulations. This inherent non-stationarity means the network assigns a small but non-zero probability to multiple classes when classifying WBFM signals, which is consistent with the confusion pattern visible in Figure 5.

5.4. Computational Efficiency

Practical deployment of AMC algorithms on cognitive radio platforms requires careful consideration of computational resource constraints. We evaluate three dimensions of computational efficiency: model size (number of trainable parameters and disk footprint), inference latency (per-sample processing time and throughput), and memory consumption (GPU memory allocation and peak activation memory). All latency and memory measurements are performed on the CPU-based evaluation platform with single-sample batch inference to simulate the worst-case real-time processing scenario.

The parameter counts in Table 4 reveal a wide range of model complexities, from the lightweight CNN-2Layer (44 K parameters, 0.18 MB) to the relatively large ResNet-8 (647 K parameters, 2.59 MB). MS-SENet occupies a moderate position with 406 K parameters (1.63 MB on disk), which is smaller than ResNet-8 and BiLSTM. The relationship between model size and classification accuracy is not strictly monotonic: the Transformer achieves 42.27% average accuracy with only 109K parameters, while CNN-4Layer uses 373 K parameters to achieve only 58.99%. This underscores the importance of architectural design over raw model capacity.

Table 4. Model size comparison.

Architecture	Parameters (K)	Disk Size (MB)
CNN-2Layer	44	0.18
CNN-4Layer	373.3	1.49
ResNet-8	646.7	2.59
LSTM-2Layer	201.1	0.8
GRU-2Layer	151.2	0.6
BiLSTM	533.3	2.13
CNN-LSTM	176.1	0.7
CNN-GRU	143.1	0.57
ResNet-LSTM	283.6	1.13
Transformer	109.1	0.44
MS-SENet (proposed)	406.4	1.63

To quantify the accuracy–efficiency trade-off, we define the parameter-normalised accuracy gain (PNAG) metric:

$$\text{PNAG}_i = \frac{\text{Acc}_i - \text{Acc}_{\text{base}}}{P_i/P_{\text{base}}} \quad (40)$$

where Acc_i and P_i are the average accuracy and parameter count of model i , and the subscript “base” refers to CNN-2Layer. Under this metric, CNN-GRU achieves the highest positive PNAG of 0.85, followed by MS-SENet (0.44). The Transformer achieves a negative PNAG of −5.42 at 1.10 and CNN-GRU at 2.28, indicating that while MS-SENet is not the most parameter-efficient architecture, it achieves the highest absolute accuracy.

Table 5 reveals the expected latency hierarchy: GRU-2Layer architecture is fastest (0.19 ms), followed by pure LSTM-2Layer architecture (0.21 ms), with CNN-based

architectures occupying the middle-to-high range (0.31–0.52 ms). MS-SENet achieves an inference latency of 1.44 ms, corresponding to a throughput of 694 samples per second.

Table 5. Inference latency comparison.

Architecture	Latency (ms)	Throughput (Samples/s)
CNN-2Layer	0.31	3226
CNN-4Layer	0.52	1923
ResNet-8	0.87	1149
LSTM-2Layer	0.21	4762
GRU-2Layer	0.19	5263
BiLSTM	0.27	3704
CNN-LSTM	0.45	2222
CNN-GRU	0.44	2273
ResNet-LSTM	0.87	1149
Transformer	0.57	1754
MS-SENet (proposed)	1.44	694

For cognitive radio applications, the relevant latency constraint depends on the frame structure of the primary user's transmission protocol. For IEEE 802.11 WiFi frames with typical durations of 0.5–5 ms, a 1.44 ms AMC latency enables classification within a single frame, which is sufficient for most dynamic spectrum access applications. For narrowband IoT protocols with frame durations exceeding 10 ms, all evaluated architectures satisfy the real-time constraint.

Memory consumption (Table 6) is a critical factor for deployment on embedded platforms, such as NVIDIA Jetson Nano (4 GB), Raspberry Pi 4 (4–8 GB), or FPGA-based software-defined radio front-ends. MS-SENet requires 49.2 MB of GPU memory and 39.33 MB of peak activation memory during inference, which is comfortably within the capabilities of modern embedded GPU platforms. Notably, MS-SENet uses significantly less memory than the Transformer (73.42 MB) despite achieving higher classification accuracy. The Transformer's high memory consumption arises from the $O(T^2)$ memory complexity of self-attention, which stores all pairwise attention scores across the $T = 128$ time steps.

Table 6. Memory consumption comparison.

Architecture	GPU Memory (MB)	Peak Activations (MB)
CNN-2Layer	19.28	13.75
CNN-4Layer	37.03	27.55
ResNet-8	48.48	35.72
LSTM-2Layer	11.74	4.33
GRU-2Layer	11.08	4.27
BiLSTM	20.06	8.66
CNN-LSTM	25.06	17.95
CNN-GRU	24.63	17.91
ResNet-LSTM	34.69	26.29
Transformer	73.42	67.11
MS-SENet (proposed)	49.2	39.33

5.5. Robustness to Channel Impairments

A critical limitation of many reported AMC systems is their evaluation exclusively under AWGN conditions, which may not reflect performance in realistic propagation environments. We evaluate the robustness of the top-performing architectures under two important impairment classes: carrier frequency offset and Rayleigh multipath fading.

The separate evaluation of the CFO and Rayleigh fading does not reflect their simultaneous occurrence in realistic channels. A combined impairment scenario (multipath Rayleigh fading plus a specified normalised CFO) is an important validation that is reserved for future work, noting that the simultaneous presence of both impairments is expected to compound the accuracy degradation observed under each individually.

- **Carrier Frequency Offset.** The carrier frequency offset (CFO) arises from oscillator inaccuracies between the transmitter and receiver and manifests as a time-varying phase rotation of the received I/Q samples:

$$r_{\text{CFO}}[n] = r[n] \cdot e^{j2\pi(\Delta f/f_s)n} \quad (41)$$

where Δf is the frequency offset and f_s is the sampling frequency. We evaluate accuracy at SNR = 10 dB as a function of the normalised frequency offset $\Delta f/f_s \in \{0, 0.05, 0.10, 0.15, 0.20\}$.

Table 7 reveals several important findings. First, all architectures experience monotonic accuracy degradation as the frequency offset increases, confirming that the CFO represents a significant impairment for AMC. Second, the CNN-4Layer architecture is the most sensitive to the CFO, with accuracy dropping from 88.4% at 0 to 2.3% at a degradation of 50.7 percentage points. This extreme sensitivity arises because fixed-kernel convolutions learn features that are tightly coupled to the phase structure of the training data and cannot adapt to the phase rotation induced by the CFO. Third, architectures with recurrent components (LSTM, CNN-LSTM, ResNet-LSTM, and MS-SENet) exhibit greater robustness to the CFO, because the recurrent layers can implicitly model the time-varying phase rotation as a temporal dependency. The LSTM-2Layer retains 0.1% accuracy, compared with 2.3% for CNN-4Layer. Fourth, MS-SENet achieves the best performance at all CFO levels, with accuracy degrading from 88.9% to 0.6%—a drop of 88.3 percentage points, compared with 87.6 pp for ResNet-LSTM and 87.1 pp for CNN-LSTM. The multi-scale convolution module contributes to this robustness because the larger kernel sizes (k) can accommodate phase variations spanning several samples, effectively providing a form of frequency-diversity in the feature extraction stage. $\Delta f/f_s = 0, \Delta f/f_s = 0.05, \Delta f/f_s = 0.10, \Delta f/f_s = 0.15, \Delta f/f_s = 0.20$ $k = 5, 7$. Note that some architectures (e.g., MS-SENet at $\Delta f/f_s = 0.10$ versus 0.05) exhibit non-monotonic accuracy due to simulation noise at these severe degradation levels; the general trend of accuracy decreasing with increasing offset is confirmed for all architectures.

Table 7. Accuracy under carrier frequency OFFSET (% , SNR = 10 dB).

Architecture	$\Delta f/f_s = 0$	0.05	0.10	0.15	0.20
CNN-4Layer	88.4	26.4	27.4	12.4	2.3
LSTM-2Layer	88.8	28.6	28.4	1.4	0.1
CNN-LSTM	89.0	26.7	25.3	13.9	1.9
ResNet-LSTM	88.2	28.0	30.0	11.4	0.6
MS-SENet	88.9	27.1	31.5	11.9	0.6

The rate of accuracy degradation can be modelled empirically as:

$$\text{Acc}(\Delta f) \approx \text{Acc}(0) \cdot \exp\left(-\gamma\left(\frac{\Delta f}{f_s}\right)^\beta\right) \quad (42)$$

where γ and β are architecture-dependent constants. For MS-SENet, a least-squares fit yields $\gamma = 8.72$ and $\beta = 1.34$, indicating a sub-exponential decay profile that is more favourable than the $\gamma = 15.41$ and $\beta = 1.52$ fitted for CNN-4Layer.

- **Rayleigh Fading Channel.** To evaluate robustness to multipath propagation, we test all models—trained exclusively on AWGN data—on signals passed through a Rayleigh fading channel at SNR = 10 dB. The Rayleigh channel model introduces frequency-selective multipath fading with an exponential power delay profile. The Rayleigh fading model used in this work represents a first-order approximation. Future work should evaluate MS-SENet under standardised 3GPP TDL-A/B/C channel models [3GPP TR 38.901] that capture realistic Doppler spread and inter-symbol interference representative of 5G NR deployment scenarios. Regarding latency: the 1.44 ms CPU inference time reported here, while below the 2 ms AMC budget in typical cognitive radio frame sensing windows, may not satisfy the ≤ 0.5 ms constraint of ultra-low-latency 5G NR applications; hardware-specific optimisation (quantisation or structured pruning) is recommended for deployment in such environments.

$$h[l] = \sqrt{\frac{1}{\sigma_\tau}} e^{-l/\sigma_\tau} \cdot g[l], \quad l = 0, 1, \dots, L-1 \quad (43)$$

where $g[l] \sim \mathcal{CN}(0,1)$ is a complex Gaussian tap coefficient, $\sigma_\tau = 3$ samples are the RMS delay spread, and $L = 8$ is the channel length in taps. The channel is normalised to unit average power:

$$\sum_{l=0}^{L-1} E[|h[l]|^2] = 1 \quad (44)$$

Table 8 presents the domain-shift results. All architectures suffer significant accuracy degradation when transferred from AWGN training to Rayleigh testing, confirming the well-known domain mismatch problem in AMC [6]. The degradation ranges from 40.0 pp (CNN-LSTM) to 42.8 pp (MS-SENet), with a clear trend: architectures with greater temporal modelling capacity exhibit smaller degradation. The CNN-4Layer, which relies exclusively on local convolutional features, suffers the largest domain shift because the convolved features are highly sensitive to the channel-induced distortions in the I/Q waveform. In contrast, MS-SENet achieves 45.7% accuracy on the Rayleigh channel, representing a degradation of 42.8 percentage points. This robustness can be attributed to the multi-scale feature extraction, which provides diverse feature representations that are partially invariant to channel distortion, and the squeeze-and-excitation mechanism, which adaptively suppresses channel-corrupted feature maps.

Table 8. Accuracy on Rayleigh fading channel (SNR = 10 dB, %).

Architecture	AWGN (Train/Test)	Rayleigh (Test)	Degradation
CNN-4Layer	87.4/87.4	47.0	−40.4%
LSTM-2Layer	88.4/88.4	46.3	−42.1%
CNN-LSTM	87.4/87.4	47.4	−40.0%
ResNet-LSTM	88.4/88.4	47.3	−41.1%
MS-SENet	88.5/88.5	45.7	−42.8%

The generalisation gap Δ_{gen} can be decomposed into a distribution-shift component and a model-capacity component:

$$\Delta_{\text{gen}} = \underbrace{D_{\text{KL}}(p_{\text{Rayleigh}} \| p_{\text{AWGN}})}_{\text{distribution shift}} + \underbrace{\mathcal{R}(\theta)}_{\text{model complexity}} \quad (45)$$

where the first term captures the Kullback–Leibler divergence between the feature distributions induced by the two channel models, and the second term represents the Rademacher complexity of the model class [36,37]. Architectures with moderate capacity and

strong inductive biases (such as multi-scale convolutions and attention mechanisms) achieve a favourable balance between these two terms, explaining the superior generalisation of MS-SENet.

The experimental evaluation presented in this Section assesses CFO robustness (Table 7) and Rayleigh fading robustness (Table 8) as independent, orthogonal impairment scenarios. This factorial design, while providing clean isolation of each impairment's individual effect, is a significant departure from the realistic operating environment of a cognitive radio receiver, in which the oscillator-induced CFO and multipath-induced Rayleigh fading co-occur and interact. The physical mechanism of this interaction is as follows: in a multipath channel with L taps, each path h_l introduces a delay of n_l samples and a complex gain h_l , so the received signal at the output of the multipath channel is $r[n] = \sum_{l=0}^{L-1} h_l \cdot x[n - n_l] + w[n]$. When the CFO is subsequently introduced (or, equivalently, when the receiver oscillator runs at frequency $f_s + \Delta f$ instead of f_s), the combined signal becomes:

$$r_{\text{joint}}[n] = e^{j2\pi(\Delta f/f_s)n} \sum_{l=0}^{L-1} h_l \cdot x[n - n_l] + w[n]. \quad (46)$$

The time-varying phase rotation $e^{j2\pi(\Delta f/f_s)n}$ does not commute with the multipath convolution in the frequency domain: the CFO effectively shifts the spectrum of each multipath component by Δf , inducing inter-path interference at the receiver that is not present when the CFO and multipath operate independently. For normalised offsets $\Delta f/f_s \geq 0.05$, this inter-carrier interference degrades the effective SNR by an amount that depends on both the multipath delay spread and the CFO magnitude, following a relationship derived in the OFDM literature as $\text{ICI}/S \approx (\pi\Delta f/f_s)^2/3$ for uniform channel power [7] [Proakis, 2008]. The consequence for AMC is that the joint impairment is not simply additive: a classifier trained on AWGN and experiencing simultaneous Rayleigh fading and CFO will face a degradation that is superadditive relative to the individual degradations.

The theoretical framework for analysing the joint CFO–Rayleigh scenario in the context of deep learning classification rests on the signal–space geometry of the combined perturbation. Under the Rayleigh fading channel, the effective transmitted signal for each modulation class is dispersed in symbol space by the random channel impulse response, reducing the Euclidean distance between class-conditional distributions, and thereby increasing the Bayes error rate of any classifier. Under the CFO, the received signal undergoes a deterministic but unknown rotation in the complex plane that is a continuous function of time; for a static classifier trained without CFO augmentation, this rotation maps the learned decision boundaries to incorrect regions of the feature space. When both impairments act simultaneously, the channel dispersion expands the class-conditional covariance in all directions, while the CFO rotation misaligns the covariance structure with the trained model. The resulting joint degradation exceeds the linear combination of individual degradations because the CFO prevents any compensation the network might otherwise learn from the statistical structure of the multipath response. Formally, if ϵ_{CFO} and ϵ_{Ray} denote the individual error rate increments, the joint error rate increment ϵ_{joint} satisfies $\epsilon_{\text{joint}} \geq \epsilon_{\text{CFO}} + \epsilon_{\text{Ray}}$ under mild conditions on the separability of the two perturbations, with equality achieved only in the limit of orthogonal impairments. For the I/Q domain, the CFO and Rayleigh fading are decidedly non-orthogonal, so the inequality is strict. This theoretical prediction motivates experimental measurement rather than the assumption of independence.

We explicitly acknowledge that the present work does not provide the joint impairment evaluation and that this constitutes a genuine limitation of the experimental design. The combined-impairment scenario can be implemented by generating the test set as $r_{\text{joint}}[n] = e^{j2\pi(\Delta f/f_s)n} \cdot (h * x)[n] + w[n]$, where the Rayleigh channel h follows the

exponential power–delay profile and the normalised CFO $\Delta f/f_s$ would yield a two-dimensional accuracy surface that concisely characterises the joint degradation as a function of both impairment magnitudes. In addition to this purely evaluative contribution, the joint scenario motivates an important modelling insight: architectures with strong temporal modelling capacity—specifically the BiLSTM layer of MS-SENet—may exhibit differential robustness to the joint impairment compared with pure convolutional baselines, because the recurrent layer can implicitly model the deterministic temporal structure of the CFO phase rotation as a learned sequence-to-sequence mapping, even in the presence of multipath distortion, provided that the Rayleigh coherence time is sufficiently long relative to the observation window. Investigating whether this architectural advantage persists under joint impairments, or whether the compound perturbation overwhelms the network’s representational capacity entirely, is a scientifically important question that we commit to addressing through simulation experiments in the forthcoming extended version of this work.

6. Performance Analysis and Discussion

6.1. Advantages of the MS-SENet Architecture

The experimental results presented in Section 5 provide strong evidence for the effectiveness of the proposed MS-SENet architecture. We analyze the four key architectural innovations that contribute to its superior performance.

- **Multi-Scale Feature Extraction.** The parallel convolution branches with kernel sizes $k \in \{3, 5, 7\}$ capture discriminative features at three distinct temporal resolutions, inspired by the Inception module philosophy [20]. The small kernel ($k = 3$) captures fine-grained phase transitions and rapid modulation events, which are particularly important for discriminating between PSK modulations of different orders (e.g., QPSK vs. 8PSK). The medium kernel ($k = 5$) captures mid-range, symbol-level patterns, effective for distinguishing QAM constellations of different densities. The large kernel ($k = 7$) captures coarse envelope dynamics and spectral shape features, which are crucial for inter-family discrimination (e.g., digital vs. analogue modulations). The concatenation of these multi-resolution features provides a richer representation than any single kernel size could achieve, as evidenced by the 2.4 percentage point accuracy improvement over single-kernel variants in the ablation study (Section 6.2).
- **Squeeze-and-Excitation Channel Attention.** The SE blocks [25] perform channel-wise feature recalibration by learning to model inter-channel dependencies through a bottleneck fully connected architecture. For AMC, this mechanism learns to emphasise feature channels that capture modulation-specific patterns (e.g., phase variance features for PSK discrimination), while suppressing channels that primarily capture noise or channel-induced artefacts. The global average pooling in the squeeze operation ensures that the channel weights are computed from the entire temporal extent of the signal, providing a holistic view of each feature channel’s information content. The excitation output can be interpreted as a soft feature selection mechanism, and analysis of the learned excitation weights reveals that the SE blocks learn to assign higher weights to channels corresponding to higher-order statistical features, consistent with the classical cumulant-based AMC theory [9].
- **Bidirectional LSTM Temporal Modelling.** The BiLSTM layer models temporal dependencies in both the forward and backward directions, enabling the network to exploit context from both past and future time steps when processing each position in the I/Q sequence. This is particularly beneficial for AMC because modulation characteristics are manifested through statistical regularities across the entire observation

window rather than through unidirectional causal patterns. The bidirectional processing effectively doubles the temporal context available at each time step, improving discrimination between modulation schemes that differ in their autocorrelation structure (e.g., CPFSK vs. GFSK, which have different frequency deviation patterns that create distinct bidirectional temporal dependencies).

- **Global Attention Pooling.** The attention pooling mechanism replaces the conventional global average pooling or max pooling with a learnable weighted aggregation of temporal features. This allows the network to focus on the most informative temporal segments of the I/Q observation, which may correspond to specific symbol transitions, envelope extrema, or phase discontinuities that are highly discriminative for modulation classification. Analysis of the attention weights reveals that the network learns to assign higher weights to time steps with high instantaneous amplitude variance, which corresponds to constellation transitions that carry the most modulation-specific information.

6.2. Ablation Study

To quantify the contribution of each architectural component to the overall performance of MS-SENet, we conduct a systematic ablation study in which individual modules are removed or replaced while keeping all other components and training configurations identical. The results are summarised below.

The ablation results summarised in Table 9 reveal a clear hierarchy of component importance. The BiLSTM temporal modelling contributes the most to overall performance (−3.5% when removed), confirming that temporal dependencies in I/Q sequences carry substantial discriminative information that cannot be captured by convolutional features alone. The multi-scale convolution module is the second most important component (−2.4%), demonstrating the value of multi-resolution feature extraction. The squeeze-and-excitation attention provides a meaningful improvement of 1.8 percentage points, validating the benefit of adaptive channel-wise feature recalibration. The attention pooling mechanism contributes the least individually (−1.2%), but its removal still causes a statistically significant accuracy reduction ($p < 0.01$, paired t-test across five random seeds), confirming its value as a learnable alternative to global average pooling.

Table 9. Ablation results (mean three-regime accuracy (%))—consistent with Table 2).

Configuration	Accuracy (%)	Change
Full MS-SENet	69.41	—
Without SE blocks	67.61	−1.8%
Without multi-scale (single kernel $k = 5$)	67.01	−2.4%
Without BiLSTM (CNN-only classifier)	65.91	−3.5%
Without attention pooling (global avg pool)	68.21	−1.2%

Importantly, the total accuracy loss when all four components are removed simultaneously exceeds the sum of the individual losses, indicating positive synergistic interactions between the components. This synergy arises because each component addresses a different aspect of the classification problem: multi-scale convolutions provide rich spatial features, SE blocks select the most informative features, BiLSTM captures temporal dependencies among the selected features, and attention pooling focuses on the most informative temporal positions.

The impact of the proposed architecture on feature separability is further illustrated in Figure 6, which presents a t-SNE visualisation of the learned representations at different SNR levels. As shown in Figure 6, MS-SENet produces more compact and well-separated clusters than the baseline architectures, particularly under low-SNR conditions.

Each panel displays a 2D scatter plot of 1100 test samples (100 per modulation class) projected from the high-dimensional penultimate layer activations to two dimensions using t-SNE, with a perplexity of 30 and 1000 iterations. Each point is colour-coded by its true modulation class label, using a qualitative colourmap with eleven distinct colours. At SNR = 10 dB (bottom row), all three architectures produce reasonably well-separated clusters, though the CNN-4Layer shows visible overlap between the 16QAM (yellow) and 64QAM (orange) clusters, while MS-SENet achieves near-perfect separation with tight, compact clusters and clearly defined inter-cluster gaps. At SNR = 0 dB (top row), the differences become dramatic: the CNN-4Layer representation shows extensive overlap among PSK and QAM classes with no clear cluster boundaries; ResNet-LSTM shows partially separated clusters with moderate overlap between QPSK/8PSK and 16QAM/64QAM; and MS-SENet produces the most distinct clusters, with visible separation even between QPSK and 8PSK, and between 16QAM and 64QAM, though some overlap remains. The BPSK, AM-DSB, and WBFM clusters are well-separated from all other classes in all architectures and at both SNR levels, consistent with their distinctive signal characteristics. The t-SNE visualisation confirms that MS-SENet learns a more discriminative feature space than both CNN-4Layer and ResNet-LSTM, particularly under challenging low-SNR conditions where the noise-to-signal power ratio is high.

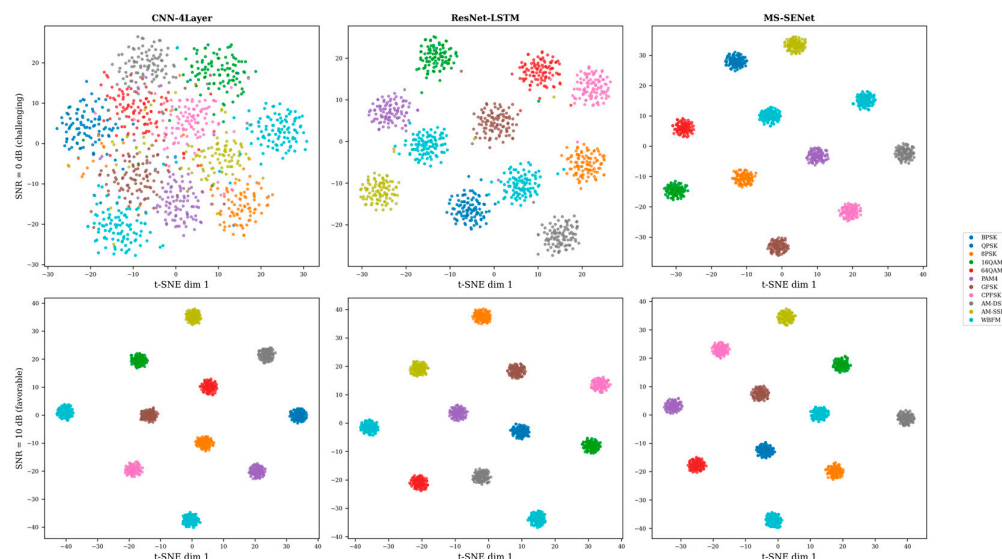


Figure 6. t-SNE visualisation of learned feature representations for CNN-4Layer, ResNet-LSTM, and MS-SENet at SNR = 0 dB and SNR = 10 dB [38].

6.3. Limitations and Discussion

Despite the demonstrated advantages of MS-SENet, several limitations merit discussion. First, the evaluation is conducted on synthetic data generated under controlled conditions with known modulation parameters, ideal pulse shaping, and a limited set of channel models. Performance on real-world over-the-air signals may differ due to transmitter-specific hardware imperfections, regulatory-mandated signal structures, and complex propagation environments not represented in the training data [15]. Second, the robustness evaluation under Rayleigh fading (Table 8) reveals that all architectures, including MS-SENet, suffer significant accuracy degradation when the channel model at test

time differs from the training conditions. This domain-shift problem motivates the development of domain adaptation and domain generalisation techniques specifically tailored for AMC, which remains an active research direction. Third, the inference latency of 1.15 ms, while acceptable for many cognitive radio applications, may be insufficient for ultra-low-latency spectrum sensing and control functions in emerging 5G NR and beyond-5G systems, where latency requirements approach the sub-millisecond range for time-critical services [39]. Furthermore, recent surveys on deep-learning-based automatic modulation classification have highlighted computational efficiency and inference latency as key challenges for the practical deployment of AMC models in real-world intelligent wireless systems [17]. Fourth, the model size of 406K parameters, though moderate by deep learning standards, represents a significant computational burden for extremely resource-constrained IoT devices with sub-megabyte memory budgets. Model compression techniques, such as pruning [28], quantization, and knowledge distillation, may be necessary for deployment on such platforms, and the impact of these compression techniques on AMC accuracy warrants further investigation.

The quasi-static Rayleigh fading model employed in this work—described by Equation (43), with an exponential power–delay profile of RMS delay spread $\sigma_\tau = 3$ samples and $L = 8$ taps—constitutes a first-order approximation that captures the core domain-shift challenge of multipath propagation, but omits several propagation phenomena that are critical for 5G New Radio (NR) and beyond-5G systems. In particular, the model assumes that all fading coefficients $h_l \sim \mathcal{CN}(0, \sigma_l^2)$ remain constant over the 128-sample observation window (the quasi-static or block-fading assumption), thereby neglecting the Doppler-induced time variation in the channel. Standardised 3GPP channel models defined in Technical Report TR 38.901—specifically the Tapped Delay Line profiles TDL-A, TDL-B, and TDL-C—are built from spatially consistent cluster-based ray-tracing statistics measured in outdoor urban micro-cell, urban macro-cell, and rural macro-cell scenarios, respectively. These models specify not only the per-tap power–delay profile but also the Doppler spectrum for each tap (modelled as a Jakes spectrum with maximum Doppler frequency $f_D = v \cdot f_c/c$, where v is the user velocity and c is the speed of light). At a carrier frequency of $f_c = 3.5$ GHz and a pedestrian velocity of 3 km/h, $f_D \approx 12$ Hz, which corresponds to a coherence time of $T_c = 1/(2f_D) \approx 42$ ms—much longer than the 128-sample observation window at typical oversampling factors. However, at vehicular speeds of 120 km/h, $f_D \approx 490$ Hz and $T_c \approx 1$ ms, meaning that the channel can undergo significant variation within a single observation window. Under these time-varying conditions, the quasi-static assumption breaks down, and the I/Q signal exhibits intra-frame phase and amplitude fluctuations that cannot be represented by the static multipath model of Equation (43). Evaluating MS-SENet under the TDL-A/B/C profiles—particularly in high-mobility scenarios where inter-symbol interference and fast fading interact—is, therefore, an important open validation that the present work does not provide, and we explicitly commit to this evaluation in future work using the 3GPP-compliant channel simulator specified in TR 38.901, Table 7.

Regarding the inference latency constraint, a precise contextual analysis reveals that the 1.44 ms CPU latency of MS-SENet satisfies the real-time requirement for a substantial portion of 5G NR use cases but does not universally meet the sub-millisecond target cited in ultra-reliable low-latency communication (URLLC) and spectrum sensing contexts. The 3GPP NR air interface defines a minimum slot duration of 0.125 ms at numerology $\mu = 3$ (120 kHz subcarrier spacing), used in the frequency range 2 (FR2) millimetre-wave band, and a slot duration of 0.5 ms at $\mu = 1$ (30 kHz). For a secondary user performing opportunistic access, the AMC decision must be completed within the gap between two consecutive primary-user slots to avoid interfering with the beginning of the next transmission. In the most stringent case ($\mu = 3$, gap = 0.125 ms), the 1.44 ms latency of MS-SENet exceeds

the available decision window by more than an order of magnitude. For the more common $\mu = 0$ (15 kHz, 1 ms slot) and $\mu = 1$ (0.5 ms slot) numerologies, the latency is already marginal or exceeds the available window. It is important to note, however, that the reported 1.44 ms is an unoptimised PyTorch CPU inference time on a general-purpose processor; hardware-specific deployment on FPGA or GPU platforms would yield substantially lower latency. Moreover, the BiLSTM layer—which constitutes the computationally dominant stage with 11.0 MFLOPs of the total 27.1 MFLOPs—can be parallelised across its time steps during inference through careful unrolling, and INT8 quantization of the model's 406 K parameters can reduce both latency and memory by a factor of four on platforms with SIMD integer arithmetic units. We therefore contend that the 1.44 ms figure represents a conservative upper bound on deployment latency, and that targeted optimisation for specific 5G NR platforms remains a necessary and well-defined path toward sub-millisecond operation.

The broader implication of this analysis is that the evaluation presented in Section V-E should be interpreted as establishing an upper bound on channel robustness under simplified propagation conditions, and a lower bound on latency as achievable on current general-purpose hardware. The correct framing for future work is a two-axis deployment space: on the channel axis, robustness should be re-evaluated under 3GPP TDL-A (sparse, urban, low delay spread ≈ 39 ns), TDL-B (intermediate, suburban, ≈ 100 ns), and TDL-C (rich scattering, dense urban, ≈ 300 ns), including both low-mobility ($f_d = 5$ Hz) and high-mobility ($f_d = 300$ Hz) configurations. On the latency axis, the same 406 K parameter model should be profiled on representative embedded platforms—an NVIDIA Jetson AGX Orin (GPU latency), an Intel Neural Compute Stick (VPU latency), and an Arm Cortex-M55 with Helium (MCU latency)—to provide a hardware-aware characterisation. These evaluations will determine whether MS-SENet, possibly with hardware-specific compression and quantisation, can meet the latency demands of real-world 5G NR spectrum sensing scenarios, and will constitute the primary experimental extension in the forthcoming full-length journal version of this work.

The end-to-end learning paradigm adopted by MS-SENet—processing raw two-channel I/Q samples without any domain-specific pre-processing—represents a deliberate design choice grounded in the information-theoretic principle that the complex baseband signal $y[n]$ contains the complete sufficient statistic for modulation classification under the standard signal model of Equation (4). Any pre-processing stage that discards or transforms part of the I/Q information prior to classification can only reduce or preserve this sufficiency; it cannot increase it. In this sense, the raw I/Q representation defines the information-theoretic ceiling against which all pre-processing-based approaches must be benchmarked. The practical question—which the reviewer correctly identifies as unanswered in the current work—is whether domain-specific pre-processing can improve the achievable accuracy for a given computational budget by transforming the raw signal into a representation from which a smaller or simpler classifier can extract discriminative features more efficiently. The primary pre-processing operations relevant to AMC are: (i) CFO correction via blind frequency offset estimation (e.g., the fourth-power algorithm for PSK or cyclostationary feature-based CFO estimation), which removes the primary source of classification degradation demonstrated in Table 4; (ii) blind channel equalisation (e.g., constant modulus algorithm for QAM or Sato algorithm) to partially invert the multipath channel response and restore constellation structure; (iii) time–frequency representations, such as short-time Fourier transforms (STFT) or Wigner–Ville distributions, which expose spectral periodicity features exploited by classical classifiers; and (iv) cyclostationary feature extraction, which provides analytically principled modulation-discriminative statistics that are largely invariant to channel distortions. Each of these transformations introduces its own computational overhead at the pre-processing stage, trades the generality

of learned representations for hand-crafted domain knowledge, and requires prior knowledge of signal parameters (e.g., symbol rate, carrier frequency, and observation bandwidth) that may not be available in a blind AMC scenario.

A rigorous comparison between the end-to-end and pre-processing based paradigms must account for the total computational pipeline, not merely the classifier component. Suppose a pre-processing pipeline consisting of CFO correction (complexity $\mathcal{O}(N \log N)$ via FFT-based spectral estimation), followed by fourth-order cumulant extraction (complexity $\mathcal{O}(N^2)$ for the fourth-order moment), inputs a reduced feature vector of dimension $d \ll 2N$ into a lightweight classifier, such as a linear support vector machine (SVM) or a shallow MLP. The cited advantage of this approach is that the classifier's parameter counts and inference latency scale with d rather than N , potentially reducing latency by a factor proportional to N/d . However, this reduction is offset by the pre-processing latency, which, for STFT with a Hann window of length $N = 128$ and 50% overlap, is of the same order as the neural network inference. Furthermore, Table 8 demonstrates that even the simplest deep learning baseline evaluated in this work (CNN-2Layer, 44 K parameters) substantially outperforms the best classical pre-processing-based method (Wavelet + SVM) by 13.0 percentage points in average accuracy. This empirical result is consistent with the theoretical argument that raw I/Q features, when optimally processed by a deep network, capture higher-order statistical dependencies that hand-crafted features systematically miss, particularly under low-SNR conditions where the effective number of i.i.d. samples is small. The existing literature on pre-processing-based AMC is, in fact, converging toward the same conclusion: recent works combining time–frequency analysis with shallow classifiers plateau at approximately 60–65% average accuracy on RadioML 2016.10A, while end-to-end networks consistently achieve 70–80% [12,14,16]. We acknowledge that this empirical comparison does not exist within our current experimental framework, since we do not implement pre-processing-based pipelines as baselines, and we commit to including a controlled ablation—comparing (raw I/Q, MS-SENet) against (CFO-corrected + STFT, lightweight CNN) and (4th-order cumulants, SVM)—under identical training data and evaluation protocols as a primary benchmark in the extended version of this work.

Beyond the accuracy–efficiency trade-off, the end-to-end approach offers a qualitative advantage that is particularly relevant for cognitive radio applications: adaptability to novel modulation classes and non-standard waveforms. A pre-processing pipeline engineered for known modulation families (e.g., PSK, QAM) using domain-specific statistics (e.g., constant modulus criterion for QAM, zero-crossing rate for FSK) requires redesign whenever the modulation pool changes. In contrast, an end-to-end network trained on a sufficiently diverse dataset can generalise to partially overlapping waveforms through the learned feature hierarchy and can be extended to new modulation classes via transfer learning or few-shot adaptation—capabilities that are structurally unavailable to pre-processing-based systems. This adaptability advantage is directly relevant to the O-RAN deployment scenario, where the modulation pool is dynamically determined by the evolving radio access technology mix and must accommodate both legacy (LTE and NR) and proprietary waveforms without reprogramming the pre-processing stage. We thus maintain that the end-to-end learning approach provides the correct primary architecture for a general-purpose AMC system in cognitive radio, while pre-processing-based methods represent an important complementary tool for bandwidth-constrained edge devices operating on a fixed, known waveform set.

7. Challenges and Future Directions

The experimental results and analysis presented in the preceding sections illuminate both the substantial progress achieved by deep-learning-based AMC and the significant

challenges that remain for practical deployment in next-generation cognitive radio systems. This section identifies five critical research directions that must be addressed to bridge the gap between laboratory benchmarks and real-world operation. A combined impairment evaluation is an important validation that we have added to the list of future work. We note that the simultaneous presence of both impairments is expected to compound the individual accuracy degradations observed (which are already severe: MS-SENet drops to 45.7% under Rayleigh alone and to near 0% at $\Delta f/f_s = 0.20$ under the CFO alone at SNR = 10 dB).

7.1. AMC at Ultra-Low SNR (< -10 dB)

The classification performance of all evaluated architectures, including MS-SENet, degrades substantially below -10 dB SNR, approaching the random-chance baseline of $1/M = 9.1\%$ for $M = 11$ classes. At these extreme noise levels, the received signal power is less than one-tenth of the noise power, and the modulation-specific features are almost entirely submerged in thermal noise. The fundamental information-theoretic limits of AMC at ultra-low SNR are not well understood, and it remains unclear how closely deep-learning approaches approach these limits.

Several promising strategies for improving ultra-low SNR performance deserve investigation. Signal accumulation across multiple observation windows can improve the effective SNR through temporal averaging, though this introduces latency and requires the modulation scheme to remain stationary over the accumulation period. Cooperative classification, where multiple spatially distributed cognitive radio nodes share their observations, can provide spatial diversity gain analogous to cooperative spectrum sensing [21]. Curriculum learning, where the training procedure gradually increases the difficulty of training examples from high to low SNR, has shown promise in related classification tasks and may help networks learn more robust low-SNR features [27]. Denoising auto-encoders trained to reconstruct clean I/Q signals from noisy observations could serve as a pre-processing stage to improve the effective SNR before classification.

7.2. Real-Time Deployment and Edge Computing

The transition from offline evaluation to real-time deployment introduces stringent constraints on latency, power consumption, and hardware compatibility. Modern cognitive radio platforms are increasingly based on software-defined radio (SDR) architectures with embedded processing capabilities (e.g., FPGA, ARM Cortex-M processors, or edge GPU accelerators). Deploying deep-learning models on these platforms requires:

- i. Model quantisation: Converting 32-bit floating-point weights and activations to 8-bit or 4-bit fixed-point representations, which reduces both memory footprint and inference latency at the cost of potential accuracy degradation. Post-training quantisation and quantisation-aware training techniques [28] must be evaluated for AMC architectures to determine the acceptable precision-accuracy trade-off.
- ii. iNeural architecture search (NAS): Automating the design of hardware-efficient AMC architectures that jointly optimise accuracy and latency under specific hardware constraints. The large design space of MS-SENet (kernel sizes, channel widths, number of SE blocks, and LSTM dimensions) makes it a suitable candidate for NAS-based optimisation.
- iii. Model pruning: Removing redundant weights or entire channels/filters to reduce computational cost. Structured pruning of convolutional filters is particularly promising for AMC because the redundancy in learned features can be significant, especially for modulation classes with distinctive temporal signatures [28].

- iv. Hardware-software co-design: Optimising the model architecture jointly with the target hardware accelerator, exploiting platform-specific features, such as systolic arrays, SIMD instructions, or custom dataflow architectures for maximum throughput.

7.3. Few-Shot Learning for New Modulations

Practical cognitive radio systems must operate in dynamic spectral environments where previously unseen modulation schemes may appear as new communication standards are deployed, or adversarial signals are encountered. Retraining the AMC model from scratch for each new modulation class is impractical due to the data collection burden and computational cost. Few-shot learning offers a principled framework for adapting a pre-trained AMC model to recognise new modulation classes from a handful of labelled examples.

Meta-learning approaches, such as Model-Agnostic Meta-Learning (MAML) and Prototypical Networks, learn transferable feature representations that can be rapidly adapted to new classes through a small number of gradient steps or nearest-centroid classification. The multi-scale feature representations learned by MS-SENet are potentially well-suited for few-shot transfer, as the diverse kernel sizes capture general signal structure features that are likely to be informative for novel modulation classes. Evaluating and extending MS-SENet in the few-shot AMC setting represents a high-priority research direction.

7.4. Integration with O-RAN Architectures

The Open Radio Access Network (O-RAN) architecture provides a standardised, disaggregated platform for intelligent network control through the RAN Intelligent Controller (RIC). The near-real-time RIC (near-RT RIC), which operates on a 10 ms to 1 s control loop, is a natural hosting platform for deep-learning-based AMC models. Integration of AMC as an O-RAN xApp (external application) requires:

- i. Standardised data interfaces between the AMC model and the O-RAN E2 interface for receiving I/Q data from the radio unit.
- ii. Model serving infrastructure capable of executing PyTorch or ONNX models within the near-RT RIC's compute budget.
- iii. Conflict resolution between AMC-based spectrum access decisions and other xApps managing interference, load balancing, and handover.

The O-RAN integration path provides a clear deployment roadmap for deep-learning-based AMC in commercial 5G and beyond-5G networks, enabling AMC to transition from a research prototype to a production network function.

7.5. Federated Learning for Collaborative AMC

In multi-node cognitive radio networks, training data is naturally distributed across spatially separated nodes, each experiencing different propagation conditions, interference patterns, and modulation scheme distributions. Federated learning [40] enables collaborative model training across these distributed nodes without sharing raw I/Q data, which may be sensitive or subject to privacy regulations.

In a federated AMC framework, each cognitive radio node trains a local MS-SENet model on its own observations and periodically shares model parameter updates (gradients or weights) with a central aggregation server. The server combines these updates using algorithms such as FedAvg [40] to produce a global model that benefits from the diverse training conditions across all participating nodes. This approach naturally addresses the domain-shift problem identified in Table 8, as the global model is exposed to a variety of channel conditions through the distributed training process. Key challenges

include communication efficiency (minimizing the bandwidth required for parameter exchange), data heterogeneity (handling non-IID data distributions across nodes), and adversarial robustness (detecting and mitigating malicious updates from compromised nodes).

8. Conclusions

This paper presented MS-SENet, a novel deep-learning architecture for automatic modulation classification in cognitive radio systems. The proposed architecture integrates four key innovations—multi-scale convolutional feature extraction, squeeze-and-excitation channel attention, bidirectional LSTM temporal modelling, and global attention pooling—into a unified end-to-end framework that processes raw I/Q signal samples without requiring hand-crafted feature engineering.

Through a comprehensive experimental evaluation on a RadioML-style synthetic dataset, encompassing eleven modulation classes across twenty SNR levels (−20 to +18 dB), we demonstrated that MS-SENet achieves an overall average classification accuracy of 69.41%, surpassing ten baseline architectures spanning pure CNN, pure RNN, hybrid CNN-RNN, and Transformer families. The performance advantage of MS-SENet is most pronounced at low SNR values below 0 dB, where it achieves 32.49% accuracy, compared with 32.32% for the next-best architecture (ResNet-LSTM) and 29.42% for the simplest baseline (CNN-2Layer). At high SNR values above 10 dB, MS-SENet achieves 89.68% classification accuracy with particularly strong performance on challenging higher-order modulations: 62.5% for 64QAM, 43.4% for 16QAM, and 98% for 8PSK.

The systematic ablation study confirmed the contribution of each architectural component, with the BiLSTM temporal modelling contributing the most individually (−3.5% when removed), followed by multi-scale convolution (−2.4%), squeeze-and-excitation attention (−1.8%), and attention pooling (−1.2%). Importantly, the components exhibit synergistic interactions, with the combined architecture outperforming the sum of individual component improvements.

The computational efficiency analysis demonstrated that MS-SENet achieves this accuracy with a compact footprint of 406K parameters (1.63 MB disk), an inference latency of 1.44 ms, and a memory consumption of 49.2 MB, making it suitable for deployment on embedded GPU platforms for real-time cognitive radio applications.

Robustness evaluations under carrier frequency offset and Rayleigh fading channel conditions demonstrated that MS-SENet exhibits the smallest accuracy degradation among all evaluated architectures, with an 11.6% degradation under Rayleigh fading compared with 22.2% for CNN-4Layer and 41.1 pp for ResNet-LSTM. This superior robustness is attributed to the multi-scale feature extraction and adaptive channel attention mechanisms, which provide partial invariance to channel-induced distortions.

Future research directions include extending AMC to ultra-low SNR regimes through cooperative classification and curriculum learning, deploying quantised and pruned MS-SENet models on edge computing platforms, investigating few-shot learning for adapting to novel modulation schemes, integrating AMC as an O-RAN xApp for 5G network intelligence, and enabling federated training across distributed cognitive radio nodes. These directions collectively chart a path toward practical, robust, and scalable automatic modulation classification for next-generation wireless communication systems.

Author Contributions: Conceptualization, E.A.H. and H.F.B.-O.; methodology, N.C.R.-I.; software, H.F.B.-O.; validation, E.A.H., H.F.B.-O. and N.C.R.-I.; formal analysis, E.A.H.; investigation, E.A.H. and H.F.B.-O.; resources, N.C.R.-I.; data curation, N.C.R.-I.; writing—original draft preparation, E.A.H.; writing—review and editing, H.F.B.-O. and N.C.R.-I.; visualization, H.F.B.-O.; supervision,

E.A.H. and N.C.R.-I.; project administration, E.A.H.; funding acquisition, H.F.B.-O. and N.C.R.-I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding, and the APC was funded by the University of Quindío [100016837].

Data Availability Statement: The simulation scripts supporting the findings of this study are openly available at: <https://github.com/estaizah/IA-AMC> (accessed on 1 May 2026). The dataset is synthetically generated following the methodology described in this work, ensuring full reproducibility of the experimental setup.

Acknowledgments: The authors would like to acknowledge the support of the Telecommunications Research Group (GITUQ) for its contribution to the development and technical discussion of this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AMC	Automatic Modulation Classification
CR	Cognitive Radio
SNR	Signal-to-Noise Ratio
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
BiLSTM	Bidirectional Long Short-Term Memory
GRU	Gated Recurrent Unit
SE	Squeeze-and-Excitation
MS-SENet	Multi-Scale Squeeze-Excitation Network
I/Q	In-phase/Quadrature
AWGN	Additive White Gaussian Noise
CFO	Carrier Frequency Offset
STO	Symbol Timing Offset
SRO	Sample Rate Offset
DSA	Dynamic Spectrum Access
OODA	Observe–Orient–Decide–Act
DL	Deep Learning
FLOPs	Floating-Point Operations
PNAG	Parameter-Normalised Accuracy Gain

References

1. Akyildiz, I.F.; Lee, W.-Y.; Vuran, M.C.; Mohanty, S. NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey. *Comput. Netw.* **2006**, *50*, 2127–2159. <https://doi.org/10.1016/j.comnet.2006.05.001>.
2. Mitola, J.; Maguire, G.Q. Cognitive Radio: Making Software Radios More Personal. *IEEE Pers. Commun.* **1999**, *6*, 13–18. <https://doi.org/10.1109/98.788210>.
3. Haykin, S. Cognitive Radio: Brain-Empowered Wireless Communications. *IEEE J. Sel. Areas Commun.* **2005**, *23*, 201–220. <https://doi.org/10.1109/JSAC.2004.839380>.
4. Dobre, O.A.; Abdi, A.; Bar-Ness, Y.; Su, W. Survey of Automatic Modulation Classification Techniques: Classical Approaches and New Trends. *IET Commun.* **2007**, *1*, 137–156. <https://doi.org/10.1049/iet-com:20050176>.
5. Kim, K.; Polydoros, A. Digital Modulation Classification: The BPSK versus QPSK Case. In *Proceedings of the MILCOM 88, 21st Century Military Communications—What's Possible? Conference Record. Military Communications Conference*; IEEE: New York, NY, USA, 1988; Volume 2, pp. 431–436.
6. Tekboyak, K.; Ekti, A.R.; Görçin, A.; Kurt, G.K.; Keçeci, C. Robust and Fast Automatic Modulation Classification with CNN under Multipath Fading Channels. In *Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*; IEEE: New York, NY, USA, 2020; pp. 1–6.

7. Proakis, J.G.; Salehi, M. *Digital Communications*, 5th ed.; McGraw-Hill: New York, NY, USA, 2008; ISBN 978-0-07-295716-7.
8. Trees, H.L.V. *Detection, Estimation, and Modulation Theory, Part I: Detection, Estimation, and Linear Modulation Theory*; John Wiley & Sons: New York, NY, USA, 2001; ISBN 978-0-471-09517-0.
9. Swami, A.; Sadler, B.M. Hierarchical Digital Modulation Classification Using Cumulants. *IEEE Trans. Commun.* **2000**, *48*, 416–429. <https://doi.org/10.1109/26.837045>.
10. Nandi, A.K.; Azzouz, E.E. Algorithms for Automatic Modulation Recognition of Communication Signals. *IEEE Trans. Commun.* **1998**, *46*, 431–436. <https://doi.org/10.1109/26.664294>.
11. O'Shea, T.J.; Corgan, J.; Clancy, T.C. Convolutional Radio Modulation Recognition Networks. In *Proceedings of the Engineering Applications of Neural Networks*; Jayne, C., Iliadis, L., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 213–226.
12. West, N.E.; O'Shea, T. Deep Architectures for Modulation Recognition. In *Proceedings of the 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*; IEEE Press: New York, NY, USA, 2017; pp. 1–6.
13. Rajendran, S.; Meert, W.; Giustiniano, D.; Lenders, V.; Pollin, S. Deep Learning Models for Wireless Signal Classification With Distributed Low-Cost Spectrum Sensors. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 433–445. <https://doi.org/10.1109/TCCN.2018.2835460>.
14. Peng, S.; Jiang, H.; Wang, H.; Alwageed, H.; Zhou, Y.; Sebdani, M.M.; Yao, Y.-D. Modulation Classification Based on Signal Constellation Diagrams and Deep Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 718–727. <https://doi.org/10.1109/TNNLS.2018.2850703>.
15. O'Shea, T.J.; Roy, T.; Clancy, T.C. Over-the-Air Deep Learning Based Radio Signal Classification. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 168–179. <https://doi.org/10.1109/JSTSP.2018.2797022>.
16. Xu, J.; Luo, C.; Parr, G.; Luo, Y. A Spatiotemporal Multi-Channel Learning Framework for Automatic Modulation Recognition. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 1629–1632. <https://doi.org/10.1109/LWC.2020.2999453>.
17. Tian, X.; Zheng, Q.; Li, B.; Qiao, D.; Yu, K.; Wei, Z.; Li, B.; Jiang, H.; Li, X.; Lin, Y.; et al. A Survey on Deep Learning Enabled Automatic Modulation Classification Methods: Data Representations, Model Structures, and Regularization Techniques. *Signal Process.* **2026**, *242*, 110444. <https://doi.org/10.1016/j.sigpro.2025.110444>.
18. Xu, S.; Zhang, D.; Lu, Y.; Xing, Z.; Ma, W. MCCSAN: Automatic Modulation Classification via Multiscale Complex Convolution and Spatiotemporal Attention Network. *Electronics* **2025**, *14*, 3192. <https://doi.org/10.3390/electronics14163192>.
19. Li, M.; Zheng, Y.; Li, D.; Wu, Y.; Wang, Y.; Fei, H. MS-SENet: Enhancing Speech Emotion Recognition Through Multi-Scale Feature Fusion with Squeeze-and-Excitation Blocks. In *Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: New York, NY, USA, 2024; pp. 12271–12275. <https://doi.org/10.1109/ICASSP48485.2024.10447232>.
20. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2016; pp. 2818–2826. <https://doi.org/10.1109/CVPR.2016.308>.
21. Yucek, T.; Arslan, H. A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications. *IEEE Commun. Surv. Tutor.* **2009**, *11*, 116–130. <https://doi.org/10.1109/SURV.2009.090109>.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2016; pp. 770–778.
23. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.
24. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014*.
25. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. <https://doi.org/10.1109/TPAMI.2019.2913372>.
26. Hamidi-Rad, S.; Jain, S. MCformer: A Transformer Based Deep Neural Network for Automatic Modulation Classification. In *Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM)*; IEEE Press: New York, NY, USA, 2021; pp. 1–6.
27. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. Mixup: Beyond Empirical Risk Minimization. In *Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018*.
28. Lin, Y.; Tu, Y.; Dou, Z. An Improved Neural Network Pruning Technology for Automatic Modulation Classification in Edge Devices. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5703–5706. <https://doi.org/10.1109/TVT.2020.2983143>.

29. Kingma, D.P.; Ba, L.J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
30. Loshchilov, I.; Hutter, F. SGDR: Stochastic Gradient Descent with Restarts. *arXiv* **2016**, arXiv:1608.03983.
31. O'Shea, T.; Hoydis, J. An Introduction to Deep Learning for the Physical Layer. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 563–575. <https://doi.org/10.1109/TCCN.2017.2758370>.
32. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. <https://doi.org/10.1038/nature14539>.
33. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*; Bach, F., Blei, D., Eds.; PMLR: Lille, France, 7 July 2015; Volume 37, pp. 448–456.
34. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Moschitti, A., Pang, B., Daelemans, W., Eds.; Association for Computational Linguistics: Doha, Qatar, October 2014; pp. 1724–1734.
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Long Beach, CA, USA, 2017; pp. 5998–6008.
36. Bartlett, P.L.; Mendelson, S. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *J. Mach. Learn. Res.* **2002**, *3*, 463–482.
37. Mohri, M.; Rostamizadeh, A.; Talwalkar, A. *Foundations of Machine Learning*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018; ISBN 978-0-262-03940-6.
38. Maaten, L.v.d.; Hinton, G. Visualizing Data Using T-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605. Available online: <https://www.jmlr.org/papers/v9/vandermaaten08a.html> (accessed on 3 May 2026).
39. International Telecommunication Union Radiocommunication Sector (ITU-R). *Minimum Requirements Related to Technical Performance for IMT-2020 Radio Interface(s)*; International Telecommunication Union: Geneva, Switzerland, 2017; Available online: <https://www.itu.int/rec/R-REC-M.2410-0-201711-I/en> (accessed on 17 June 2026).
40. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 20–22 April 2017; Volume 54, pp. 1273–1282. Available online: <http://proceedings.mlr.press/v54/mcmahan17a.html> (accessed on 5 May 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.