Efficient Squeeze-and-Excitation-Enhanced Deep Learning Method for Automatic Modulation Classification

Nadia Kassri[1], Abdeslam Ennouaary[2], Slimane Bah[3]
National Institute of Posts and Telecommunications, Rabat, Morocco[1, 2]
Mohammadia School of Engineers, Rabat, Morocco[3]

***Abstract*—The rapid proliferation of mobile devices and Internet of Things (IoT) gadgets has led to a critical shortage of spectral resources. Cognitive Radio (CR) emerges as a propitious technology to tackle this issue by enabling the opportunistic use of underexploited frequency bands. Automatic Modulation Classification (AMC), which serves as a technique to blindly identify modulation types of received signals, plays a pivotal role in carrying out several CR functions, including inference detection and link adaptation. Recent research has turned to Deep Learning (DL) networks to overcome the shortcomings of traditional AMC techniques. However, most existing DL approaches are impractical for resource-limited systems. To address this challenge, we propose a novel lightweight hybrid neural network for AMC that fuses Convolutional Neural Networks (CNNs) and Gated Recurrent Units (GRUs) layers, along with a customized Squeeze and Excitation (SE) block. The integration of CNNs and GRUs allows for the learning of both spatial and temporal dependencies in modulated signals, while the SE block recalibrates features by modeling interdependencies between CNN network channels. Our experimental results, using the RadioML 2016.10A dataset, clearly demonstrate the superior performance of our approach in effectively managing the tradeoff between accuracy and complexity compared to baseline methods. Specifically, our approach achieves the highest accuracy of 91.73%, surpassing all reference models while reducing the memory footprint by at least 45%. In future work, further investigation is warranted to differentiate modulations sharing temporal or frequency domain characteristics and enhance classification accuracy in high-noise environments.**

Keywords*—Cognitive radio; modulation classification; deep learning, convolutional neural networks; Gated Recurrent Units; squeeze and excitation

I. Introduction

Nowadays, we are witnessing a period marked by an exceptional proliferation of wireless applications and the emergence of radio technologies like the Internet of Things (IoT). This proliferation has indelibly altered the landscape of how we communicate, gather information, and interact with our environment. According to the most recent data available as of 2023, the global count of mobile devices is projected to attain 18.22 billion by 2025[1]. Simultaneously, the IoT ecosystem has seen explosive growth, with an estimated 15 billion connected devices globally in 2023, reshaping industries across the spectrum, from healthcare to agriculture [2].

This rapid expansion of IoT-connected devices is a testament to the ongoing digital revolution. Projections suggest that by 2030, the count of IoT-connected devices will reach a staggering 29.42 billion, highlighting the remarkable trajectory of this transformative technology [2].

However, this proliferation has brought about a unique set of challenges, one of the most pressing being the shared allocation of frequency bands. Notably, many of these devices, ranging from smartphones to environmental sensors, operate within the same spectral boundaries, exerting substantial pressure on this finite and invaluable resource. Consequently, we find ourselves in a congested spectrum landscape that necessitates innovative solutions to optimize its utilization while maintaining the dependable communication upon which modern society relies.

Cognitive Radio (CR) has emerged as a promising technology to address these challenges by enabling the opportunistic use of spectral bands underutilized by licensed users [3]. A CR, essentially a Software-Defined Radio (SDR), can actively monitor its surroundings, assess spectrum occupancy, and autonomously adjust its operational parameters to prevent disruptive interference with licensed users [3].

Implementing the aforementioned CR tasks through a centralized system can introduce latency issues and substantial network traffic due to data exchange among devices, exacerbating the challenge of spectrum scarcity. To effectively tackle this issue, edge computing and non-cooperative approaches are progressively becoming the preferred solutions, particularly in IoT applications. In these approaches, end-devices take on some or all of the computation-intensive CR functions, resulting in reduced communication overhead and quicker response times [4].

Spectrum sensing, a critical step in the cognitive cycle, involves exploring the radio environment, detecting available channels, and acquiring valuable data, such as the modulation types of sensed signals [3]. This modulation information is pivotal for detecting physical layer attacks and facilitating various CR tasks like link adaptation and dynamic spectrum access [5]. Automatic Modulation Classification (AMC) holds a central role in this process, involving two key stages: "Signal preprocessing" for extracting essential signal parameters like carrier frequency, symbol period, noise power, and signal power, followed by the "Application of a classification algorithm" to determine the modulation formats of detected signals [6].

In the realm of AMC, traditional techniques, including Likelihood-based (LB) and Feature-based (FB) methods, have long been foundational. LB methods approach modulation classification as a complex multiple-hypothesis testing scenario, relying on the calculation of likelihood functions and the application of predefined thresholds for classification decisions [5]. Conversely, FB approaches extract intricate features from intercepted signals and employ classifiers like K-Nearest Neighbor (KNN) and Support Vector Machines (SVM), leveraging handcrafted features such as wavelet transforms, cyclic statistics, and high-order cumulants [5], [7].

Traditional modulation classification methods face limitations, with LB approaches having high computational complexity and requiring signal knowledge, while FB schemes, although more practical due to lower computational complexity, may not ensure optimal accuracy [5], [8].

Motivated by the extraordinary success of deep learning (DL) networks in fields like computer vision and image recognition, recent research has turned to DL networks to address the limitations of traditional AMC methods. DL-based solutions represent a paradigm shift, operating as comprehensive learning systems that smoothly merge feature extraction and classification tasks. This innovative approach streamlines the automatic extraction of high-level features, removing the necessity for manually crafted features that frequently lack robust characterization [7].

Despite the remarkable promise of DL-based AMC techniques in achieving exceptional classification accuracy through extensive data utilization, their application to autonomous IoT end devices is hampered by a myriad of specific challenges. These challenges include but are not limited to the substantial energy consumption, demanding processing requirements, and extensive storage prerequisites inherent in many DL-based approaches [6]. Moreover, the resource-constrained nature of IoT devices exacerbates these challenges, with limited memory, real-time response constraints, modest computing power, and low battery life further complicating the implementation of AMC. This practical constraint severely restricts the deployment of DL-based AMC techniques in IoT networks, where operational efficiency and adaptability are paramount considerations. In such resource-constrained environments, the compatibility of DL-based AMC methods becomes even more precarious, underscoring the need for alternative solutions tailored to the unique constraints of IoT devices [4].

In this work, our primary focus centers on AMC, with a particular emphasis on its applicability to resource-constrained devices. Within this scope, we've developed a novel lightweight hybrid neural network that seamlessly integrates Convolutional Neural Networks (CNNs) for spatial feature mapping and Gated Recurrent Units (GRUs) for temporal feature extraction. To enhance accuracy while minimizing computational costs, we've incorporated a customized Squeeze and Excitation block after Convolutional Neural Network (CNN) layers. This block has been meticulously engineered to optimize feature extraction, improving model performance without overburdening computational resources.

It's worth noting that CNNs are renowned for their capacity to extract spatial features from input data, making them well-suited for capturing patterns and structures within modulation signals. On the other hand, GRUs excel at capturing temporal dependencies, allowing the model to discern sequential patterns and dynamics over time. By integrating these two architectures, our method capitalizes on the strengths of both CNNs and GRUs, enabling a comprehensive analysis of both spatial and temporal characteristics present in modulation signals [5].

The key contributions of this paper can be succinctly outlined as follows:

- We introduce a meticulously designed DL-based AMC scheme that prioritizes optimal accuracy and computational efficiency. This model seamlessly integrates a CNN block for intricate feature extraction and a GRU block to capture essential temporal dependencies.
- Our work includes the development of a finely tuned Squeeze and Excitation (SE) block, which enhances accuracy while keeping computational costs at a minimum.
- We conduct a rigorous performance assessment of our model, encompassing a comprehensive evaluation against state-of-the-art AMC techniques using prominent dataset, namely the RadioML 2016.10A dataset [9]. This evaluation incorporates critical factors such as inference time, training time, number of trainable parameters, and classification accuracy.

The following sections of this paper are carefully arranged to offer a systematic examination of our research. In Section II, we give an overview of related works in the field, offering valuable context for our contributions. In Section III, we meticulously detail the architecture of our suggested method, emphasizing its unique components and elucidating how they synergistically enhance the overall performance. In Section IV, we present the intricate implementation details and empirical results, offering a thorough comparison of our model's performance against state-of-the-art AMC techniques to assess its effectiveness. Finally, Section V summarizes the noteworthy contributions made by this work and delineates potential avenues for further research and exploration.

II. Related Work

The application of DL techniques in the context of AMC has garnered significant attention in recent research. This increasing interest is driven by the promising advantages that DL offers for the development of future communication networks.

DL architectures, encompassing CNNs, Recurrent Neural Networks (RNNs), Long Short-Term Memory Networks (LSTMs), and GRUs, have all contributed to this surge in interest [6], [8], [10].

RNNs, inherently suited for time series data, have grappled with the vanishing gradient problem, prompting the introduction of LSTMs and GRUs. These latter architectures employ internal mechanisms referred to as "gates" to regulate information flow, offering solutions to mitigate the vanishing gradient issue.

GRUs, distinguished by their efficiency through fewer training parameters, consume less memory and execute faster than LSTMs [5].

Moreover, bidirectional variants such as bidirectional GRU (BiGRU) and bidirectional LSTM (BiLSTM) have emerged, capable of capturing features in both forward and backward paths. This capability endows them with improved context-dependency compared to GRU and LSTM models, consequently enhancing the learning process's performance while incurring greater computational complexity [5].

Complementing the RNNs, CNNs stand out as prominent and successful DL networks that leverage convolution and pooling techniques to derive advanced features from data. CNNs excel particularly in computer vision tasks, where their adoption has catalyzed significant advancements [5], [11].

Table I lists relevant DL-based AMC methods along with their basic structures and implementation conditions.

An example of a Recurrent Neural Network (RNN) based AMC model is reported in study [12]. In this work, the authors proposed a novel AMC method using RNNs, which has demonstrated its capability to learn the temporal characteristics of received signals. This method directly utilizes raw signals with limited data length, eliminating the need for manual signal feature extraction. The proposed approach is compared with a CNN-based method, and the results highlight the superiority of the RNN-based approach, particularly when the Signal-to-Noise Ratio (SNR) exceeds -4dB. Furthermore, a comparative study evaluates various RNN structures, ultimately recommending a more efficient two-layer Gated Recurrent Unit (GRU) network. Numerical results illustrate that this recommended structure significantly enhances classification accuracy, improving it from 80% to 91%. However, although the study by the authors is significant, it neglects to consider training and inference times, which are important for evaluating a model's complexity and feasibility in real-time scenarios.

Additionally, another study in [4] introduced a GRU-based AMC model tailored for devices with limited resources. This model comprises a GRU layer succeeded by a SoftMax layer, designed following a comprehensive parameter study that considers metrics such as training set size, input vector length, layers count, and GRU cells number. The research also generated a unique dataset with over-the-air measurements of real radio signals collected using the resource-constrained SDR experimental platform MIGOU. All simulations were conducted using this dataset, showcasing the model's impressive results: a memory footprint of 73.5 kBytes, a 51.74% reduction compared to the baseline model, and a recognition accuracy of 92.4%. Although the proposed model generates few parameters and has a reduced memory footprint, it has not been evaluated under low SNRs, as the used MIGOU dataset contains SNRs with average values superior to or equal to 22dB. Moreover, the inference time is not considered in the evaluation.

In the research work presented in study [13], authors introduced a cost-efficient CNN-based AMC model known as MCNet, featuring a unique architecture with specific convolutional blocks utilizing various asymmetric convolution kernels. This design choice enables MCNet to effectively capture the intricate spatiotemporal signal correlations essential for accurate modulation classification. Additionally, strategically integrated skip connections within MCNet's architecture mitigate overfitting and address the vanishing gradient problem. These skip connections play a pivotal role in preserving crucial residual information across multi-scale feature maps. On the DeepSig dataset, MCNet achieves an overall accuracy rate exceeding 93% at 20 dB SNR. Despite the meticulously conceived CNN blocks and the use of innovative techniques to enhance the accuracy, this model fails to achieve a good balance between complexity and accuracy compared with other DL-based AMC methods [6].

Similarly, in another research effort in [14], the authors proposed a DL-based technique called ICAMCNet for classifying signal modulation with lower inference time, making it suitable for real-world networks that demand low-latency communications, like those beyond 5G. To achieve this goal, a reduced number of filters was employed to decrease computational time, and various layers were incorporated, including dropout and Gaussian noise layers, to enhance accuracy and mitigate overfitting. The ICAMCNet model achieved a highest accuracy of 91.70% and exhibited a latency of less than 0.01 ms when evaluated using the RML2016.10b dataset. However, despite its reduced inference time, the model has over one million trainable parameters resulting in a larger footprint, making it unsuitable for resource-constrained devices.

Furthermore, authors in [15] introduced a three-stream DL framework for Automatic Modulation Recognition, referred to as MCLDNN. This innovative approach efficiently extracts features from individual and combined in-phase/quadrature (I/Q) symbols by integrating one-dimensional (1D) convolutional, two-dimensional (2D) convolutional, and LSTM layers. When evaluated on the RadioML2016.10a dataset, MCLDNN surpasses other frameworks with SNRs above -4dB, achieving an impressive maximum accuracy of 92.95%. However, this outstanding accuracy comes at the cost of a larger number of trainable parameters, totalling 406,199, and superior inference and training times compared to several AMC models.

Another noteworthy hybrid DL-based AMC model, called PET-CGDNN, is introduced in study [16], leveraging phase parameter estimation and transformation. This model incorporates CNN and GRU layers for feature extraction, resulting in high recognition accuracy comparable to baseline models on the RML2016.10b dataset, achieving an average accuracy of 63.82% and the highest accuracy of 93.41%. Remarkably, it achieves this while reducing more than a third of its parameters. Moreover, PET-CGDNN demonstrates superior performance in terms of both training and test times when compared to benchmark models with similar recognition accuracy. This model strikes a good balance between accuracy and complexity, a balance we aim to surpass in our work.

TABLE I. DL-BASED AMC METHODS: BASIC STRUCTURE AND IMPLEMENTATION CONDITIONS

Model	Basic structure	Trainable parameters	SNR range(dB)	Frame length	Dataset/ modulations	Training and test sets (Numbers of vectors)	Channels	Hardware specifications
GRU2 [12]	GRU	151 179	−20: 2: 18	128	RadioML2016.10A dataset	Training:110k. Test: 110k.	AWGN. Center frequency offset. Selective multipath Rician fading. Sample rate offset.	NVIDIA GTX1080 GPU
GRU1 [4]	GRU	18 375	37 dB/ 22 dB (high SNR levels)	128	MIGOU dataset	Training: 2.2 million. Test: 2.2 million.	Multipath fading AWGN Frequency offset	Not mentioned
MCNet (6 M-blocks) [13]	CNN	142 000	-20: 2: 30	1024	RadioML2018.10A dataset	Training: 2 million. Test:500k.	AWGN Doppler shift Non-impulsive delay spread Symbol rate offset Carrier frequency offset Selective multipath Rician fading	NVIDIA GeForce GTX 1080Ti GPU, 16GB RAM, and 3.70-GHz CPU.
ICAMCNet [14]	CNN	1.2 million	-20: 2: 30	128	RadioML2016.10B dataset	Training:720k. Test: 480k.	AWGN Center frequency offset Selective multipath Rician fading Sample rate offset	12 GB GDDR5 VRAM, GPU 1xTesla K80, and 2496 CUDA cores.
MCLDNN [15]	CNN+LSTM	406 199	−20: 2: 18	128	RadioML2016.10A dataset	Training:132k. Test:44k.	AWGN. Center frequency offset. Selective multipath Rician fading. Sample rate offset.	NVIDIA GeForce GTX 1080Ti GPU.
PET-CGDNN [16]	CNN+GRU	72k	-20: 2: 30	128	RadioML2016.10B dataset	Training:720k. Test :240k.	AWGN Center frequency offset Selective multipath Rician fading Sample rate offset	NVIDIA GeForce GTX 1080Ti
Lightweight Backbone Network [11]	CNN	46k	-10: 2: 20	1024	RadioML2018.10A dataset	Training: 1 million. Test:250k	AWGN Doppler shift Non-impulsive delay spread Symbol rate offset Carrier frequency offset Selective multipath Rician fading	NVIDIA GeForce RTX 2080 Super GPU, 32 GB RAM, and 2.9 GHz CPU.
[17]	CNN+GRU	52.5k	−20: 2: 18	128	RadioML2016.10A dataset	Training:176k. Test:44k	AWGN Center frequency offset Selective multipath Rician fading Sample rate offset	NVIDIA Quadro T1000, 32 GB RAM, and Intel(R) Core (TM) i7-10850H CPU
SCNN [18]	CNN	96k	−10 :2: 20	128	BPSK, QPSK, 8PSK, PAM2, 2FSK, 4FSK, 8FSK, PAM4, PAM8, and 16QAM.	Trainig:60k. Test:100k.	AWGN Phase offset	NVIDIA GeForce GTX 1080Ti
RFNet128 [19]	CNN	137.3k	-20: 2: 30	1024	RadioML2018.10A dataset	Not mentioned	AWGN Doppler shift Non-impulsive delay spread Symbol rate offset Carrier frequency offset Selective multipath Rician fading	RTX A6000 GPUs and 48 GB VRAM
[20]	CNN+GRU	8 210	-20: 2: 30	128		Training: 1 million. Test:255k		NVIDIA QUADRO M600, 32 GB RAM, and CPU E5-2660 v4 @ 2.00GHz × 28

In research [11], a novel CNN AMC was introduced. This architecture incorporates a bottleneck layer and asymmetric convolution structures to minimize computational complexity, catering to real-time communication needs in CR networks. Evaluation using the RadioML 2018.01A dataset shows remarkable classification accuracy, especially in the -4 dB to 20 dB SNR range, with notable accuracies improvement of 5.52% and 5.92% at SNRs 0 dB and 10 dB, respectively. Additionally, their model significantly reduces trainable parameters by over 67% compared to MCNet and decreases signal processing prediction time by more than 54.4%. A comprehensive comparison with conventional models in study [11] highlights the effectiveness of their proposed architecture in handling AMC challenges in CR networks. It is worth noting that in this study, the authors did not re-implement all the models for comparison purposes. Instead, they relied on the results and values provided in the original papers, which does not guarantee a fair comparison due to potential disparities in implementation conditions.

Similarly, the paper in study [17] introduces a lightweight neural network (NN) built by merging a GRU layer and a set of convolutional blocks. The latter is meticulously designed using asymmetric filters to reduce computational complexity and SE blocks to enhance channel interdependencies. In this structure, skip connections are also incorporated to ameliorate accuracy and alleviate the vanishing gradient problem. Simulations on the RadioML 2016.10A dataset prove that this model surpasses baseline models in terms of accuracy while using a reduced number of trainable parameters. Despite the achieved performance, more efforts should be made to further reduce inference time.

In the study reported in study [18], the authors directed their efforts toward the implementation of decentralized learning methods for AMC by leveraging a separable CNN (SCNN). This SCNN approach was distinguished by its incorporation of model consolidation and a lightweight design, leading to the development of a significantly more efficient model in contrast to the centralized SCNN-based AMC approach. This enhanced model not only demonstrated improvements in training efficiency and a reduction in communication overhead but also maintained its classification performance. With a parameter count of 96 thousand, SCNN's training efficiency was estimated to be roughly N times greater than that of SCNN-based centralized learning, with N being the number of edge devices utilized. Remarkably, their model exhibited heightened accuracy in comparison to a standard CNN, while concurrently achieving a substantial reduction in both spatial and temporal complexities by up to 94% and 96%, respectively. While the SCNN model typically exhibits the lowest computational complexity and memory footprint, its classification accuracy consistently ranks lowest when compared to numerous DL-based AMC methods [16].

Furthermore, in the context of addressing the hardware resource demands of deep networks for AMC, an innovative approach called RFNet was presented in study [19]. The proposed RFNet introduces a Multiscale Convolutional (MSC) layer and utilizes Separable Convolution Blocks (SCB) to reduce network complexity, resulting in an efficient deep neural network solution for AMC. The RFNet family, including RFNet+, and RFNet++ that are built using pruning and quantization techniques, offers variations with fewer parameters and floating-point operations. The problem with pruning is that it often leads to degradation in accuracy and necessitates significantly longer training times. Nonetheless, these advancements hold promise for future AMC systems [20].

Similarly, for the same objective, in order to improve model compression and resource utilization, a novel iterative magnitude-based pruning approach combined with Quantization-Aware Training (QAT) was introduced in [20]. Simulation results using the RadioML 2018.01A dataset validate the effectiveness of the proposed approach in reducing DL model complexity while guaranteeing acceptable accuracy. The problem with this approach is the long training time.

This comprehensive review of the related works underscores the broad spectrum of approaches and innovations within the field of AMC, encompassing novel network architectures and endeavors to enhance hardware efficiency. However, it is notable that achieving an effective balance between classification accuracy and computational complexity remains a persistent challenge. Typically, models that excel in accuracy tend to exhibit heightened complexity, and conversely, those with low complexity often come at the cost of reduced accuracy. To address this pivotal issue and seek a more favorable equilibrium, our paper introduces a novel DL-based model for AMC, prioritizing both high accuracy and reduced computational complexity.

These advancements in AMC techniques may hold significant implications for various real-world applications. With the exponential growth of wireless devices and applications across industries such as healthcare and agriculture, the demand for efficient use of the frequency spectrum is increasing. AMC techniques play a crucial role in optimizing spectrum utilization, improving communication reliability, and enabling the deployment of innovative wireless technologies [21].

From a societal perspective, AMC contributes to bridging the digital divide by ensuring reliable connectivity in underserved areas and enabling access to essential services such as education and healthcare. Economically, AMC techniques can lead to cost savings through better utilization of spectrum resources, reduced interference, and improved network efficiency.

Moreover, the technological impacts of AMC extend beyond traditional communication systems. They pave the way for the development of advanced wireless networks, including 5G and beyond, as well as emerging technologies such as IoT.

Overall, the broad adoption of AMC techniques has the potential to revolutionize various sectors, driving innovation, improving quality of life, and fostering economic growth.

III. Signal Model and Proposed Method

A. Signal Model

Modulation classification serves as a core function in wireless communication systems, often framed as an N-class classification task, where each class represents a unique modulation scheme [4]. The received signal can undergo various

environmental changes as it travels through the radio environment. These changes encompass phenomena such as multipath fading and shadowing effects, which result from the signal's reflection, refraction, and scattering in the environment. These environmental effects introduce fluctuations in signal strength, potentially leading to signal distortion or loss. Consequently, for a transmitted signal x(t), the received signal y(t) can be expressed as follows:

$$y(t)=x(t)*h(t)+n(t) \quad (1)$$

In the above equation, h(t) signifies the channel gain, encapsulating all the effects experienced by the signal during its propagation, and n(t) denotes the Additive White Gaussian Noise (AWGN).

Within CR systems, radio receivers are capable of delivering received signals in an I/Q format. This I/Q format divides a signal into two elements, commonly referred to as the in-phase (I) and quadrature (Q) components [4]. These components can be expressed mathematically as:

$$I=A\cos\theta \quad (2)$$

$$Q=A\sin\theta \quad (3)$$

A and θ represent the instantaneous amplitude and phase of y(t). The I and Q components contain valuable details about the signal, including its frequency, phase, and amplitude. This information facilitates the identification of the modulation scheme employed in a particular communication signal.

B. Squeeze and Excitation (SE) Approach

The Squeeze-and-Excitation technique operates at the heart of feature recalibration, offering a nuanced solution to enhance the discriminative power of convolutional neural networks (CNNs). By directly capturing the relationships between different channels, SE dynamically adapts channel-wise feature responses during the network's forward pass. This adaptability proves crucial, especially when confronted with the inherent challenges of varying data patterns and input characteristics [22].

In essence, the "squeeze" phase involves compressing global information into a set of channel-wise descriptors, while the subsequent "excitation" phase utilizes these descriptors to recalibrate the importance of each channel's features. The result is a network that can dynamically emphasize the most salient features, contributing significantly to improved model performance. Fig. 1 illustrates the architecture of a typical SE block [22].

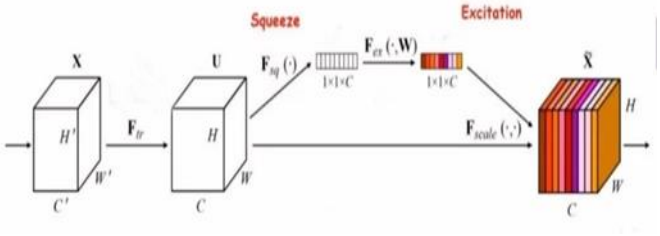

Fig. 1. Typical SE block [21].

An SE block serves as a computational unit that can be applied to a transformation $\boldsymbol{F_{tr}}$, mapping an input $\boldsymbol{X} = [x^1, x^2, \dots, x^{C'}] \in \mathbb{R}^{H' \times W' \times C'}$ to feature maps $\boldsymbol{U} = [\boldsymbol{u_1}, \boldsymbol{u_2}, \dots, \boldsymbol{u_C}] \in \mathbb{R}^{H \times W \times C}$. In the subsequent notation, we consider $\boldsymbol{F_{tr}}$ to be a convolutional operator and utilize $\boldsymbol{V} = [\boldsymbol{v_1}, \boldsymbol{v_2}, \dots, \boldsymbol{v_C}]$ to depict the learned set of filter kernels, with $\boldsymbol{v_c}$ represents the parameters of the c-th filter. Then, the output $\boldsymbol{u_c}$, corresponding to the output feature map produced by the c-th channel, can be expressed as follows [22]:

$$u_c = v_c * \boldsymbol{X} = \sum_{i=1}^{C'} v_c^i * x^i \quad (4)$$

where, * represents convolution operator and $v_c = [v_c^1, v_c^2, \dots, v_c^{C'}]$, $with\ v_c^i$ is a 2D spatial kernel representing a single channel of v_c, which operates on the corresponding channel $x^i\ of\ \boldsymbol{X}$.

1) Squeeze operation: The squeeze operation aggregates global information across spatial dimensions for each channel within U using a Global Average Pooling (GAP), transforming its C feature channels into a one-dimensional vector $z \in \mathbb{R}^C$. The c-th element of z can be computed as follows [22]:

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \quad (5)$$

2) Excitation operation: The excitation operation adaptively recalibrates the importance of each channel based on the channel-wise descriptor obtained from the squeeze operation. It should involve a gating mechanism to capture nonlinear interactions between channels and ensure a non-mutually-exclusive relationship and then to allow multiple channels to be emphasized simultaneously. To meet these requirements, the excitation phase generally use a gating mechanism with a sigmoid activation function [22].

To control model complexity and improve generalization, the gating mechanism is typically configured by creating a bottleneck using two fully-connected (FC) layers around the non-linearity. This entails a layer for reducing dimensionality with a reduction ratio r, followed by a ReLu activation, and subsequently, a layer to increase dimensionality, ultimately restoring the output to the channel dimension of the transformed result U. The reduction ratio r is often chosen through empirical studies.

The output of the excitation function $F_{ex}(.,W)$ can be expressed as follows [22]:

$$s = F_{ex}(z,W) = \sigma\big(g(z,W)\big)\sigma\big(W_2\delta(W_1 z)\big) \quad (6)$$

where, σ refers to the sigmoid activation, δ denotes the ReLU activation, $W_1 \in \mathbb{R}^{\frac{C}{r} \times C}$, and $W_1 \in \mathbb{R}^{C \times \frac{C}{r}}$.

3) Scale operation: The final scale operation combines the original feature map U with the recalibrated version s. The output for a given channel can be expressed as follows [22].

$$\tilde{x}_c = F_{scale}(u_c, s_c) = s_c u_c \quad (7)$$

where, $F_{scale}(u_c, s_c)$ refers to the channel-wise multiplication between the scalar s_c and the feature map u_c. The

resulted feature map across all channels $\tilde{X} = [\tilde{x}_1, \tilde{x}_{2,\dots,}\tilde{x}_C] \in \mathbb{R}^{H\times W\times C}$.

C. Proposed Method

In this subsection, we provide a detailed description of our proposed DL-based AMC method by focusing on the architectural choices and configurations made in its development. As shown in Fig. 2, the proposed model leverages a combination of CNN layers for feature extraction, a custom SE block for feature recalibration, and a GRU layer for temporal dependencies learning.

The input data consists of 2D representations of radio signals in the I/Q format. Each signal is shaped as a (128, 2, 1) vector, where '128' represents the number of samples, and '2' denotes the 'I' and 'Q' components of each sample. The initial processing step involves the use of a Zero Padding layer, which effectively pads the data with zeros to address spatial dimensions. Following this, we apply a Batch Normalization (BN) layer to standardize the data and improve convergence.

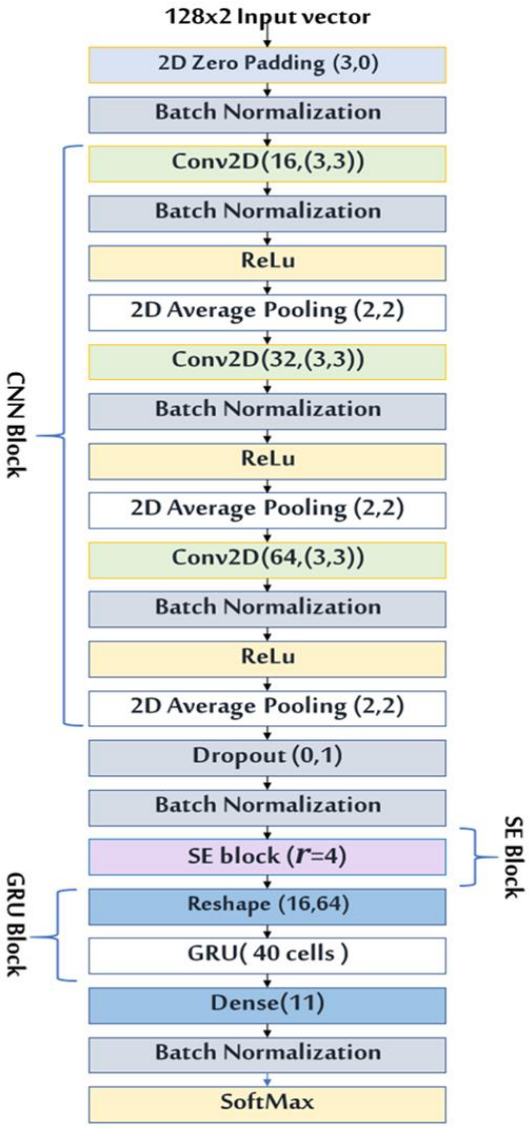

Fig. 2. Proposed method.

Afterward, our model incorporates three 2D convolutional (Conv2D) layers with the same kernel size of (3, 3) and different numbers of filters, which are 16, 32, and 64, respectively. After each convolutional layer, we apply a BN layer, a Rectified Linear Unit (ReLU) activation function, and an Average Pooling layer with a (2, 2) pool size to down-sample the feature maps. it's noteworthy that increasing the number of filters as we go in-depth allows the model to capture more complex and abstract features.

The cornerstone of our DL architecture is the Custom SE Block (see Fig. 3), strategically applied after the CNN layers. This block has been meticulously designed to improve feature recalibration and enhance the model's ability to prioritize the most informative aspects of the input data. The architecture of the SE block begins by globally averaging the input feature maps using a Global Average Pooling layer, resulting in a tensor with reduced spatial dimensions. This tensor is then reshaped to (1, 1, -1) vector, essentially converting it into a channel-wise representation. Subsequently, two Conv2D layers with 1x1 kernels are applied: the first reduces the number of channels using a reduction coefficient (r=4), followed by a BN layer and ReLU activation function; the second produces channel-wise attention scores through a sigmoid activation function. An additional step involves calculating the mean value of the attention scores, serving as a dynamic threshold. Scores exceeding this threshold are retained, while those falling below it are set to zero, ensuring that the model prioritizes the most informative data elements. These rectified attention scores are then element-wise multiplied with the original input tensor, ensuring that channels are selectively emphasized based on their learned importance. It's noteworthy that in our personalized SE block, the reduction coefficient is deliberately set to 4, a value determined through empirical experiments involving different numbers.

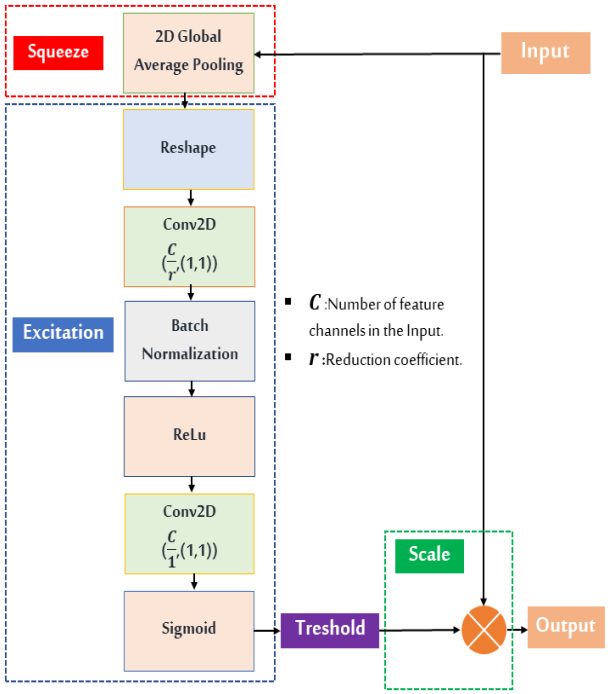

Fig. 3. SE block.

Subsequently, the output of the SE block is fed into the GRU block. The latter contains a GRU layer preceded by a reshape operation to adapt the input data to a 3D tensor. This block contributes to capturing the temporal dependencies within the data.

Finally, to map the signal features learned from the previous layers to the appropriate modulation format, a dense layer with 11 units, followed by a BN layer and a SoftMax layer, is applied.

It's important to note that while there are no definitive rules for selecting optimal hyperparameters such as kernel sizes and the number of filters, our decisions were guided by empirical experimentation and established best practices in CNN design. Tools like Optuna framework can be used for automating the search for the best parameter combinations.

IV. Experiments and Results

A. Dataset

In our experiments, we have utilized the RadioML 2016.10A dataset to assess the performance of our proposed model. This dataset comprises 220,000 modulated signals in the I/Q format, distributed across 20 distinct SNR levels and 11 unique modulation schemes [9]. For each (SNR, modulation) combination, there exists a subgroup of 1,000 signals.

Covering a wide SNR range from -20 dB to +18 dB, the RadioML 2016.10A dataset provides an extensive depiction of real-world signal propagation scenarios.

Each signal is organized into frames containing 128 samples, capturing the temporal characteristics inherent to the corresponding modulation type. These 11 modulation formats encompass both digital and analog classes, offering a wide spectrum of communication scenarios frequently encountered in practical applications.

Table II provides a concise summary of the key characteristics of the RadioML 2016.10A dataset.

In all experiments, it's important to highlight that the dataset was divided using a ratio of 3:1:1. Specifically, 60% of the data was assigned for training, while 20% was dedicated to validation, and the remaining 20% was preserved for testing.

TABLE II. RadioML 2016.10A Dataset

Parameter	Value /description
Number of modulation types	11
Modulation formats	**Analog:** AM-DSB, AM-SSB, WBFM.
	Digital: 8PSK, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK.
Signal format	I/Q format
SNR range	-20: 2: 18
Number of instances per modulation-SNR pair.	1,000
Global count of instances	220,000
Vector shape	2x128

B. Simulation Results and Analysis

Experiments were carried out using the following software and hardware setup: Python 3.9.7, Keras 2.7, and TensorFlow 2.7, executed on a workstation equipped with an Ubuntu 18.04.6 LTS Operating System. The workstation featured an Intel® Xeon(R) CPU E5-2660 v4 @ 2.00GHz × 28 Processor, 32 GB of RAM, and an NVIDIA Quadro M6000/PCIe/SSE2 Graphics Processing Unit (GPU) with Compute Unified Device Architecture (CUDA) support, significantly enhancing processing speed.

In all experiments, the Adam optimizer with a learning rate of 0.001 and the categorical cross-entropy loss function were employed. To prevent overfitting, a callback was implemented to cease training when the validation accuracy value showed no improvement for 12 consecutive epochs. Training was conducted over 100 epochs, with the learning rate reduced by 90% every 20 epochs.

1) Performance evaluation metrics: To rigorously evaluate the performance of our model, we have employed a range of commonly recognized metrics, like accuracy, precision, recall, and F1 score. Accuracy, as a fundamental measure, determines the model's classification ability by computing the ratio of correctly identified instances to the global count of vectors in the dataset. As for precision, it is defined as the ratio of correctly classified positive vectors to all vectors classified as positive, while recall assesses the percentage of actual positive instances that are correctly classified as positive. In applications where the cost of a false positive is high, precision is a critical metric, while recall is vital in scenarios where the cost of a false negative is high. To obtain a comprehensive evaluation of the model's performance, both precision and recall should be taken into account. The F1 score is a widely used metric that combines both precision and recall metrics to provide a single measure of the model's overall performance. It calculates the harmonic mean of precision and recall and is useful when both precision and recall are equally important.

In addition, in the realm of cognitive radio networks, the computational complexity of models is of paramount importance, particularly in real-time communication scenarios. To accurately assess this complexity, we consider three key metrics: the number of trainable parameters, test time, and training time.

2) Comparison with baseline models: In the first experiment, the performance of our model is evaluated through a comparative analysis with conventional models, including GRU2 [12], CLDNN [10], SCNN [18], MCLDNN [15], MCNet [13], and PET-CGDNN [16].

Based on the in-depth analysis detailed in Table III and visual data represented in Fig. 4, a clear and convincing pattern emerges from the comparison between our proposed model and the state-of-the-art models. This pattern underlines the ability of our model to achieve an optimal compromise between complexity and accuracy. Specifically, the proposed model surpasses all other models in classification accuracy, attaining an average accuracy rate of 62.08% and a maximum accuracy of 91.73%, while keeping the number of trainable parameters at the lowest level (39,003). Furthermore, our model demonstrates superior performance compared to all baseline models across recall, precision, and F1 score metrics. Notably, it achieves a

significant enhancement in recall from 0.57% to 49%, precision from 0.55% to 8.89%, and F1 score from 1.11% to 38.94%.

While the SCNN model excels in achieving an impressive inference time of 0.029 milliseconds (ms) per sample and boasts a minimal training duration of 15 seconds (s) per epoch, its accuracy falls short, averaging at 46.61% and peaking at 69.23%. Compared to this model, our proposed model achieves notably superior accuracy while saving 62.5% of trainable parameters and maintaining competitive training and inference times at 16 seconds per epoch and 0.038 milliseconds per sample, respectively.

In contrast, the MCLDNN model closely rivals our suggested model in classification accuracy, averaging at 61.64% with a peak of 91.45%. However, this comes at the expense of heightened complexity, demonstrated by a significantly larger number of trainable parameters at 406,199 and longer times for both making predictions (0.1 milliseconds per sample) and training (39 seconds per epoch).

Concerning the PET-CGDNN model, it achieves an acceptable average accuracy of 61.06% and a highest accuracy of 91.36%, accompanied by a moderate training time of 16 seconds per epoch. However, our proposed model outperforms PET-CGDNN by saving 45.7% of trainable parameters and reducing test time by 0.016 milliseconds per sample.

Regarding the MCNet, CLDNN, and GRU2 models, our model clearly outperforms them across all metrics.

In summary, our suggested method achieves an outstanding balance between accuracy and complexity. It stands out as the top choice by delivering the highest accuracy and the fewest trainable parameters, along with shorter training and inference times, making it an attractive solution for modulation classification applications.

3) Ablation study: The ablation study on our suggested method reveals valuable insights into the significance of each block within the architecture and the impact of varying reduction coefficients on its performance. As shown in Table IV, when we eliminate the GRU block, we observe a notable drop in average accuracy, from 62.08% to 60.32%. This result underscores the crucial role of the GRU block in improving the model's ability to learn temporal dependencies and patterns within the data, which is particularly important in modulation classification tasks. On the other hand, the computational complexity added by this block is deemed moderate. Specifically, it increases the inference time by 0.005 milliseconds/sample and the training time by two seconds per epoch. Furthermore, it adds 31.9% trainable parameters to the final model.

As for the SE block, its absence induces a drop in accuracy almost similar to the case of the absence of the GRU block (decreasing the average accuracy from 62.08% to 60.65%). However, the SE block enhances accuracy with minimal complexity cost, increasing the number of trainable parameters by only 2160 and the inference time by only 0.002.

The full proposed model with both the GRU and SE blocks demonstrates the highest average accuracy at 62.08% and the highest accuracy at 91.73%. These results underscore the importance of the complete architecture, where both the GRU and SE blocks work synergistically to achieve the best performance.

Table V demonstrates that reduction coefficients equal to or lower than 4 exhibit nearly identical accuracies, albeit with varying trainable parameters, which increase as the SE reduction coefficient decreases. Conversely, reduction coefficients greater than 4 yield a less significant decrease in trainable parameters but are accompanied by reduced accuracy. Consequently, a reduction coefficient of 4 is selected, offering an accuracy of 62.08% while maintaining a reduced number of trainable parameters (39,003).

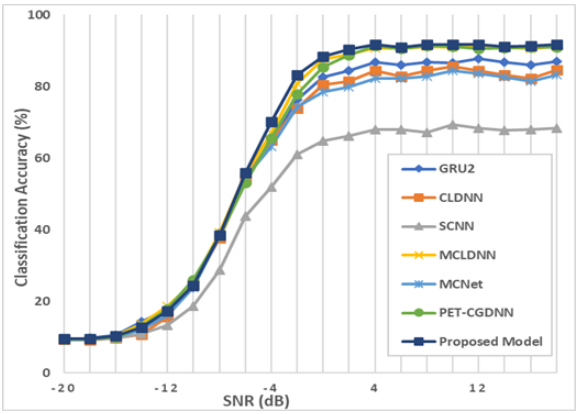

Fig. 4. Classification accuracy comparison between our model and the benchmark models.

TABLE III. PERFORMANCE COMPARISON BETWEEN OUR METHOD AND THE BENCHMARK APPROACHES

Model	Average accuracy (%)	Highest accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Trainable Parameters	Training time (s/epoch)	Inference time (ms/sample)
GRU2	58.97	87.72	82.35	48.81	61.29	151,179	25	0.071
CLDNN	57,15	85.54	79.17	46.14	58.30	167,243	18	0.047
SCNN	46.61	69.23	79.87	25.72	38.91	104,011	**15**	**0.029**
MCLDNN	61.64	91.45	84.74	50.16	63.02	406,199	39	0.1
MCNet	56.45	84.45	78.79	45.66	57.82	121,611	27	0.068
PET-CGDNN	61.06	91.36	86	49.45	62.79	71,871	16	0.054
Proposed Model	**62.08**	**91.73**	**86.48**	**50.45**	**63.73**	**39,003**	16	0.038

TABLE IV. IMPACT OF GRU AND SE BLOCKS ON MODEL PERFORMANCE

Model	Average accuracy (%)	Highest accuracy (%)	Trainable Parameters	Training time (s/epoch)	Inference time (ms/sample)
Without GRU block	60.32	89.41	26,547	14	**0.033**
Without SE block	60.65	90.32	36,843	**13**	0.036
Proposed Model	**62.08**	**91.73**	**39,003**	16	0.038

TABLE V. IMPACT OF VARYING REDUCTION COEFFICIENTS ON MODEL PERFORMANCE

Reduction coefficient	Avearage accuracy (%)	Trainable parameters
1	61.93	45,291
2	62.07	41,099
3	61.81	39,658
4	**62.08**	39,003
5	61.50	38,879
6	,61.69	38,217
7	61.66	38,086
8	61.63	37,824
9	61.47	**37,693**

4) Performance of our proposed scheme over 11 modulation formats: We evaluated the performance of our suggested method for 11 different modulations. The confusion matrix in Fig. 5, obtained at an SNR level of 4 dB, highlights our model's ability to accurately classify most modulation schemes, achieving a classification accuracy of over 97% for 7 modulation formats. However, differentiating between WBFM and AM-DSB presents a significant challenge. Notably, approximately 51% of WBFM signals are erroneously categorized as AM-DSB. This misclassification primarily results from the presence of overlapping silent intervals in both modulation types, where the carrier signal continues. Furthermore, the shared time-domain characteristics and similarities between AM-DSB and WBFM exacerbate the confusion.

Additionally, our model faces difficulty in distinguishing between QAM16 and QAM64. This challenge arises from the inclusion of the constellation points of QAM16 within QAM64, leading to confusion between these two types of modulation during the classification process.

It's worth noting that the misclassification of WBFM signals as AM-DSB signals, as well as the difficulty in distinguishing between QAM16 and QAM64, are prevalent issues in DL-based AMC methods, particularly when utilizing the RadioML2016.10a dataset. Table VI presents the misclassification rates of these signals by the baseline models and our proposed model. Both our model and the PET-CGDNN model demonstrate the lowest misclassification rates between QAM16 and QAM64. However, it is notable that almost all models exhibit a misclassification percentage slightly exceeding 50% when attempting to differentiate between WBFM and AM-DSB signals.

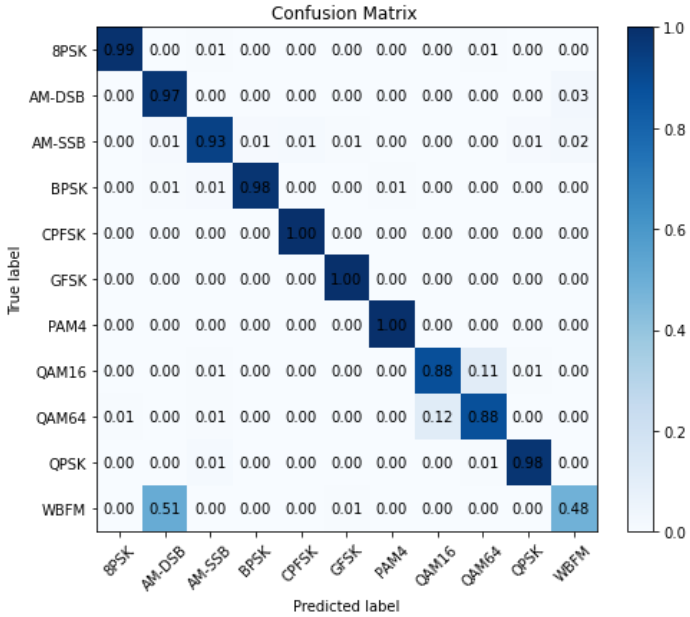

Fig. 5. Confusion Matrix of our model at 4 dB SNR.

TABLE VI. MISCLASSIFICATION RATES OF WBFM, QAM16, AND QAM64 SIGNALS

Model	WBFM signals misclassified as AM-DSB signals	QAM16 signals misclassified as QAM64 signals	QAM64 signals misclassified as QAM16 signals
GRU2	53 %	30 %	29 %
CLDNN	55 %	53 %	33 %
SCNN	62 %	35 %	24 %
MCLDNN	52 %	16 %	23 %
MCNet	**49 %**	59 %	23 %
PET-CGDNN	52 %	**10 %**	**13 %**
Proposed Model	51 %	**11 %**	**12 %**

V. CONCLUSION

This paper introduces a cutting-edge SE-Enhanced DL approach for AMC, seamlessly integrating CNN and GRU layers with a customized SE block to maximize accuracy and computational efficiency. It outperforms baseline models across multiple metrics. Notably, it achieves a peak accuracy of 91.73%, superior to that of all reference models while reducing memory footprint by at least 45%. Furthermore, our method showcases exceptional efficiency with rapid training and inference speeds, boasting an inference time of 0.033 ms/sample and a training time of 16 s/epoch, outperforming the majority of reference models in speed and performance. This combination

of heightened accuracy and reduced complexity positions our model as a viable solution for real-world implementation, especially in resource-constrained environments where memory space and processing time are critical factors.

However, our model faces challenges in accurately classifying certain signals, such as misclassifying WBFM signals as AM-DSB and distinguishing between QAM16 and QAM64 signals. Addressing these limitations and exploring techniques like pruning and quantization to further reduce model complexity while maintaining acceptable accuracy level, particularly in high-noise environments, constitute the objectives of our future work.

REFERENCES

[1] "Forecast number of mobile devices worldwide from 2020 to 2025 (in billions)." Accessed: Oct. 22, 2023. [Online]. Available: https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/

[2] "Number of IoT connected devices worldwide 2019-2023, with forecasts to 2030." Accessed: Oct. 22, 2023. [Online]. Available: https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/

[3] N. Kassri, A. Ennouaary, S. Bah, and H. Baghdadi, "A Review on SDR, Spectrum Sensing, and CR-based IoT in Cognitive Radio Networks," International Journal of Advanced Computer Science and Applications, vol. 12, no. 6, pp. 100–121, Autumn 2021, doi: 10.14569/IJACSA.2021.0120613.

[4] R. Utrilla, E. Fonseca, A. Araujo, and L. A. Dasilva, "Gated Recurrent Unit Neural Networks for Automatic Modulation Classification with Resource-Constrained End-Devices," IEEE Access, vol. 8, pp. 112783–112794, 2020, doi: 10.1109/ACCESS.2020.3002770.

[5] N. Kassri, A. Ennouaary, and S. Bah, "Lightweight Hybrid Deep Learning Scheme for Automatic Modulation Classification in Cognitive Radio Networks," in 2022 9th International Conference on Future Internet of Things and Cloud (FiCloud), IEEE, Aug. 2022, pp. 113–118. doi: 10.1109/FiCloud57274.2022.00023.

[6] T. Huynh-The et al., "Automatic Modulation Classification: A Deep Architecture Survey," IEEE Access, vol. 9, pp. 142950–142971, 2021, doi: 10.1109/ACCESS.2021.3120419.

[7] D. Zhang et al., "Automatic Modulation Classification Based on Deep Learning for Unmanned Aerial Vehicles," Sensors, vol. 18, no. 3, p. 924, Mar. 2018, doi: 10.3390/s18030924.

[8] Z. Zhu and A. Nandi, Automatic modulation classification: principles, algorithms and applications. 2015. Accessed: Apr. 26, 2022. [Online]. Available: https://books.google.com/books?hl=fr&lr=&id=AZtUDwAAQBAJ&oi=fnd&pg=PR11&dq=Automatic+Modulation+Classification:+Principles,+Algorithms+and+Applications&ots=ZdVUeXTLnW&sig=M3wy4yWYZMPjFCY6xYT5hkcB-JM

[9] T. O'shea, "Radio Machine Learning Dataset Generation with GNU Radio," 2016.

[10] "Deep architectures for modulation recognition". N. E. West and T. O'Shea, "Deep architectures for modulation recognition," in 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), IEEE, Mar. 2017, pp. 1–6. doi: 10.1109/DySPAN.2017.7920754.

[11] S.-H. Kim, J.-W. Kim, V.-S. Doan, and D.-S. Kim, "Lightweight Deep Learning Model for Automatic Modulation Classification in Cognitive Radio Networks," IEEE Access, vol. 8, pp. 197532–197541, 2020, doi: 10.1109/ACCESS.2020.3033989.

[12] D. Hong, Z. Zhang, and X. Xu, "Automatic modulation classification using recurrent neural networks," in 2017 3rd IEEE International Conference on Computer and Communications (ICCC), IEEE, Dec. 2017, pp. 695–700. doi: 10.1109/CompComm.2017.8322633.

[13] T. Huynh-The, C.-H. Hua, Q.-V. Pham, and D.-S. Kim, "MCNet: An Efficient CNN Architecture for Robust Automatic Modulation Classification," IEEE Communications Letters, vol. 24, no. 4, pp. 811–815, Apr. 2020, doi: 10.1109/LCOMM.2020.2968030.

[14] A. P. Hermawan, R. R. Ginanjar, D.-S. Kim, and J.-M. Lee, "CNN-Based Automatic Modulation Classification for Beyond 5G Communications," IEEE Communications Letters, vol. 24, no. 5, pp. 1038–1041, May 2020, doi: 10.1109/LCOMM.2020.2970922.

[15] J. Xu, C. Luo, G. Parr, and Y. Luo, "A Spatiotemporal Multi-Channel Learning Framework for Automatic Modulation Recognition," IEEE Wireless Communications Letters, vol. 9, no. 10, pp. 1629–1632, Oct. 2020, doi: 10.1109/LWC.2020.2999453.

[16] F. Zhang, C. Luo, J. Xu, and Y. Luo, "An Efficient Deep Learning Model for Automatic Modulation Recognition Based on Parameter Estimation and Transformation," Oct. 2021.

[17] N. Kassri, A. Ennouaary, and S. Bah, "Efficient Hybrid Neural Network for Automatic Modulation Recognition," 2024, pp. 347–359. doi: 10.1007/978-981-97-0744-7_29.

[18] X. Fu et al., "Lightweight Automatic Modulation Classification Based on Decentralized Learning," IEEE Trans Cogn Commun Netw, vol. 8, no. 1, pp. 57–70, Mar. 2022, doi: 10.1109/TCCN.2021.3089178.

[19] Mohammad Chegini, Pouya Shiri, and Amirali Baniasadi, "RFNet: Fast and efficient neural network for modulation classification of radio frequency signals," ITU Journal on Future and Evolving Technologies, vol. 3, no. 2, pp. 261–272, Sep. 2022, doi: 10.52953/XBPT2357.

[20] N. Kassri, A. Ennouaary, S. Bah, I. Hajjaji, and H. Dahmouni, "Pruning-Based Hybrid Neural Network For Automatic Modulation Classification In Cognitive Radio Networks," J Theor Appl Inf Technol, vol. 15, no. 3, 2024.

[21] Han'guk T'ongsin Hakhoe, IEEE Communications Society, Denshi Jōhō Tsūshin Gakkai (Japan). Tsūshin Sosaieti, and Institute of Electrical and Electronics Engineers, ICTC 2020 : the 11th International Conference on ICT Convergence : "Data, Network, and AI in the Age of 'Untact'" : October 21-23, 2020, Ramada Plaza Hotel, Jeju Island, Korea.

[22] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," Sep. 2017, [Online]. Available: http://arxiv.org/abs/1709.01507.