Compressed Automatic Modulation Recognition Deep Learning Network Based on Bi-LSTM (CS-Bi-LSTM)

Hossam M. Kasem [1,2], Haithem S. Khallaf [3], and Sherief Hashima [4,5,*]

[1] Computer Science Department, Faculty of Engineering, Egypt Japan University of Science and Technology, Alexandria, Egypt
[2] Electronics and Electrical Communications Department, Faculty of Engineering, Tanta University, Egypt
[3] Reactors Department Nuclear Research Center, Egyptian Atomic Energy Authority, Cairo, Egypt
[4] Engineering Department, Egyptian Atomic Energy Authority, Cairo, Egypt
[5] Computational Learning Theory Team, RIKEN-AIP, Fukuoka, Japan
Emails: hossam.kasem@f-eng.tanta.edu.eg (H.M.K.); eng.h.khallaf@gmail.com (H.S.K.);
eng_shrif@yahoo.com (S.H.); sherief.hashima@riken.jp (S.H.)
*Corresponding author

***Abstract*—The revolutionary advancements of Fifth Generation (5G) technology have redefined wireless communication systems, establishing a critical platform for integrating Artificial Intelligence (AI) with modern telecommunications. This paper emphasizes leveraging deep learning to enhance Automatic Modulation Recognition (AMR) within the physical layer, especially in non-cooperative scenarios. Amid the increasingly complex and shared electromagnetic spectrum, AMR is crucial for efficient signal processing. Researchers address signal sparsity challenges by proposing Compressed Sensing (CS), enabling modulation identification in the compressed domain without full signal reconstruction. This paper introduces an innovative deep learning framework, CS- A Bidirectional Long Short-Term Memory (Bi-LSTM), combining CS with bidirectional long short-term memory networks. This architecture ensures high bandwidth signal acquisition via nonuniform low-rate sampling, excels in contextual feature extraction and addresses long-term dependencies. A SoftMax classifier further refines classification accuracy. Simulation results confirm that this groundbreaking approach surpasses existing deep learning models in AMR tasks, establishing a new benchmark for intelligent wireless communication systems.**

***Keywords*—Fifth Generation (5G), Deep Learning (DL), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BI-LSTM), Automatic Modulation Recognition (AMR), Compressed Sensing (CS)**

I. Introduction

Automatic Modulation Recognition (AMR) is crucial in future wireless communication, as it autonomously identifies diverse modulation types within signals, enhancing signal processing efficiency and robustness in dynamic environments. Its widespread adoption in commercial and military fields is attributed to its high efficiency and flexibility [1, 2]. Among the most popular implementations of this technology are Software Defined Radios (SDRs), which employ blind modulation detection to rapidly adapt to diverse communication systems without requiring extensive control. In a military context, automatic modulation classification is essential for securely receiving friendly signals and swiftly identifying hostile ones, even with minimal prior information. These demands enhanced real-time signal processing capabilities in addition to blind modulation identification schemes. This capability enables the prompt detection of unfriendly signals, which is essential for making intelligent decisions to protect friendly forces. Recognizing modulation is vital in identifying the source of wireless signals and supports informed decision-making for a self-aware, autonomous wireless communication system.

AMR serves as an intermediate step between detecting and demodulating a signal, enabling the extraction of information from a signal even without prior knowledge of its system parameters. This makes AMR a necessary prerequisite for demodulation at the receiving end of a signal [3]. Due to the growing demand for communication services, resources are becoming scarce, which may result in illegal organizations occupying the public spectrum and disrupting regular communication. To tackle this problem, AMR can be utilized to differentiate the modulation approach of interfered and legitimate user signals, investigate their characteristics, and manage spectrum resources accordingly.

AMR technology can be categorized into two main types: Likelihood-Based (LB) schemes [4] and feature-aided approaches [5]. LB-enhanced techniques have theoretically optimal performance but require a lot of computing power. On the other hand, feature-based

Manuscript received February 4, 2025; revised April 9, 2025, accepted April 18, 2025; published June 23, 2025.

methods depend on experts' expertise in feature extraction, which is unsuitable for increasingly complex communication systems. Therefore, Deep Learning (DL), a powerful machine learning approach, is used as an AMR method [6–8].

The LB technique selects the modulation type that maximizes a log-likelihood function based on the known probability distribution associated with the phase or amplitude of the received signals for the candidate modulation types. Achieving this task requires a lot of computing power. It involves a multiple-composite hypothesis-testing problem where a probabilistic model is built for the received signal. Classification of the modulation type is done by comparing Likelihood Functions (LFs) or likelihood ratios against a threshold. The Average Likelihood Ratio Test (ALRT) considers unknown quantities as random variables and averages them to compute the LF.

LB classification addresses a combination of hypothesis and testing problems where each incoming signal is assumed to have a modulation type. The Generalized Likelihood Ratio Test (GLRT) calculates the Probability Density Function (PDF) of incoming signals using maximum likelihood estimations based on unknown quantities to determine the most likely modulation type. Other techniques have been proposed in the literature, such as the Hybrid Likelihood Ratio Test (HLRT), quasi ALRT, and quasi HLRT. Researchers have utilized the LB approach for modulation recognition, with the likelihood method offering optimal performance, but it delivers high computational complexity. Hence, selecting threshold values and addressing modeling mismatches is crucial for accurate classification. The quasi-ALRT algorithm provides near-optimal performance for Frequency Shift Keying (FSK) and Phase Shift Keying (PSK)modulated signals at low Signal-to-Noise Ratio (SNR) but may not work well for Quadrature Amplitude Modulation (QAM) signals [9]. GLRT may struggle with nested constellations, while HLRT can have high time complexities due to numerous unknown parameters. Quasi-HLRT classifiers offer lower complexity with accurate estimators.

The feature-based AMR methods depend on experts' expertise in feature extraction, which is unsuitable for increasingly complex communication systems. Model-driven approaches mostly choose their features based on experience. Feature based method techniques lose specific original details while extracting some statistical features. This affects the categorization performance, especially in low-SNR circumstances. At the same time, the DL-based network may extract highly representative features from the source signals and incorporate feature extraction as part of the classifier training process. Consequently, it surpasses conventional feature-based method approaches in terms of classification performance. Therefore, based on AMR types, we propose to use a DL algorithm that overcomes the drawbacks of both approaches [10–13].

Recently, there has been a substantial increment in the amount of investigation on the utilization of DL for communication objectives, such as Convolutional Neural Networks (CNN) [14, 15], Long Short-Term Memory (LSTM) [16], and Recurrent Neural Networks (RNN) [17, 18], etc. DL techniques can offer more accuracy and flexibility in identifying distinct modulation types than conventional expert-dependent methods. Hence, this paper suggests using a Bi-directional LSTM (Bi-LSTM) network to detect modulations.

Long training time is the major challenge of Deep Neural Networks (DNN). For example, using three NVIDIA Tesla P100 GPU chips, a simplified CNN configuration in Ref. [19] would require about 40 minutes of training. Using these algorithms in realistic scenarios, as online training is necessary to adjust the network design with varying environmental conditions, which is challenging. Therefore, the challenge of mitigating the training time is a critical point in ensuring such algorithms' success. Furthermore, because most traditional AMR algorithms demand Nyquist-rate (or faster) samples of the incoming analog signal, front-end samplers may be severely burdened when the signal has a large bandwidth. This work proposes combining Compressed Sensing (CS) with a Bi-LSTM network to address these challenges. First, Bi-LSTM model has proven its ability to capture temporal dependencies in both forward and backward directions, which is crucial for effectively modeling time-series data such as modulation signals. It is known for its strong performance in sequential data analysis, particularly in scenarios where temporal patterns span multiple scales [20, 21]. Second, CS is a powerful technique to reduce the burden on the front-end sampler when dealing with high-bandwidth signals. CS takes a small number of random, nonuniform samples from the signal, which are then used to reconstruct the signal. This technique has been effective for signals with few significant Fourier coefficients within their high bandwidth. Our use of CS is motivated by its ability to retain temporal dependencies under certain conditions, as supported by the Restricted Isometry Property (RIP). RIP ensures that the distances between signal vectors are approximately preserved in the compressed domain, allowing sequential models like Bi-LSTM to operate effectively on the compressed data. Hence, the combination of CS and Bi-LSTM leverages the strengths of both components—efficiency from CS and robust temporal modeling from Bi-LSTM.

Hence, the proposed methodology in this paper introduces a novel integration of CS and Bi-LSTM networks to address the challenges associated with high-bandwidth signal processing. The Bi-LSTM network extracts critical features and temporal dependencies from the data. At the same time, the CS framework significantly reduces the computational burden on the front-end sampler by compressing the signal at the acquisition stage. Furthermore, the training efficiency is enhanced through dimensionality reduction and sub-sampling, which decreases the number of feature vectors in the datasets. The synergistic combination of CS and Bi-LSTM networks offers a robust solution for high-bandwidth signal analysis, enabling efficient data extraction while minimizing the load on the front-end sampler. This approach not only improves computational efficiency but also has the potential to achieve higher accuracy and substantially reduce training time. The key contributions of this paper are summarized as follows:

- Pioneering integration in AMR: This study is the first to integrate CS and Bi-LSTM networks in the field of AMR, establishing a novel framework for high-bandwidth signal processing.
- Efficient training framework: A cascaded architecture combining CS and Bi-LSTM models is proposed, which significantly reduces the training time while maintaining high performance.
- Superior performance: Extensive simulation results validate the effectiveness of the proposed CS-Bi-LSTM framework, demonstrating its superiority over existing recognition neural networks in terms of accuracy and computational efficiency.

These main contributions collectively advance the state-of-the art in high-bandwidth signal processing, offering a scalable and efficient solution for real-world applications.

The paper is structured as follows: Section I provides an overview of the relevant research. The signal model, data set, and programming environment used in this work are presented in Section II. Our suggested CS-Bi-LSTM network setup and its component parts are described in Section IV. Section V presents the findings of an experiment to measure the proposed approach's effectiveness. Finally, Section VI concludes the paper. Table I summarizes a list of abbreviations utilized in this paper according to its appearance.

II. Related Work

LB-ARM approach determines the modulation type by comparing the received signal's likelihood function with a known modulation pool [9, 22]. This technique has demonstrated high accuracy in various channel conditions [23]. However, specific parameters, such as carrier frequency, code rate, and channel parameters, must be known beforehand. In contrast, the feature-aided approach identifies the modulation type by collecting features from the delivered signal. Although this approach does not need any former signal information, its accuracy is lower than that of the LB method [24].

Feature-based AMR methods capture the characteristics of the incoming signal, and the modulation type can be recognized by either transforming these features to predetermined thresholds or inputting them into a pattern recognition program [25, 26]. Various features like simultaneous statistics, higher order statistics, time-frequency features, and sequential delay sampling statistics can be extracted from the signal [27], which are then used as inputs for classifiers like decision trees and support vector machines [28]. Although this approach is computationally simple, it may not perform well on non-linear problems. Furthermore, manually selecting the features can result in the classifier failing to recognize signals with different modulations, leading to decreased classification accuracy.

DL has gained remarkable focus in recent years as a robust machine-aided scheme that has attained impressive success in various fields, including image classification [14, 29, 30] and speech recognition [31]. DL enables machines to identify high-level attributes or features from scattered data representations by combining low-level features. DL aims to equip machines with the ability to analyze text, images, and sounds similarly to humans. DL uses activation functions to address nonlinear classification issues, and regularization can be employed to improve the model's robustness [32]. DL networks have a multi-layer architecture that automatically selects and optimizes features from signals, eliminating the need for laborious manual selection of data features. This can help reduce classification errors.

The success of DL algorithms in various applications can be attributed to the use of large data sets and mathematical models. Thus, Researchers employing CNN have used DL to automate AMR. Compared to expert-based approaches, the results demonstrated that this CNN had higher accuracy and more flexibility in recognizing different modulation types [19]. Simulation results have shown that a CNN is more precise and flexible in recognizing various modulation types than current expert-based approaches. To enhance feature propagation in deep neural networks, Residual Networks (ResNet) [33] and Densely Connected Networks (DenseNet) [34] were developed by creating shortcut paths between different layers. In Ref. [34], a ResNet design was suggested to identify 24 different forms of modulation, and the simulation results indicated that it was effective in classifying various modulations. Additionally, DenseNet was employed for modulation recognition in [35].

A Convolutional Long Short-Term Deep Neural Network (CLDNN) was discussed in Ref. [36], which combines CNNs, LSTM units, and conventional deep neural network architectures. RNNs are suitable for recognizing and learning sequences such as speech and handwriting, and the LSTM unit is a memory unit of an RNN. To address the problem of vanishing gradients in RNNs, LSTM was developed, which includes a forget gate in its memory cell to allow for long-term dependency learning. In Ref. [37], LSTM was proposed for use in neural networks, which resulted in high classification accuracy for various modulation formats. A CNN-LSTM [38] and multi-channel large kernel CNN (MCLCNN) [39] models for AMR were introduced with improved classification abilities. Tang *et al.* [40] proposed a Reparameterization Causal Convolutional Network (RepCCNet) for more accurate AMC performance. A single hidden layer NN with small trainable parameters for AMC in IoT applications is highlighted in Ref. [41]. A Two-Stream Transformer (TSTR) based model for AMR of underwater acoustics is addressed in Ref. [42] with high recognition results. Moreover, a Meta-Supervised Contrastive Learning (MSCL) scheme for few-shot open-set modulation classification is introduced in Ref. [43]. Furthermore, Xiao *et al.* [4] introduced a novel Masked Contrastive Learning with Hard Negatives (MCLHN) method to address the challenges of Automatic Modulation Classification (AMC) under limited labeled data. Extensive experiments on benchmark datasets demonstrate that MCLHN outperforms existing methods in both performance and generalization capability.

Notably, MCLHN achieves competitive results with just one labeled sample per modulation type under each SNR, rivaling the performance of other methods that require five to twenty times more labeled data.

In this study, Nonuniform Compressive Sampling (NCS) is explored to reduce the measurement burden for AMR. Instead of reconstructing a Nyquist-rate sample vector and applying a classical AMR algorithm, CS is employed to reduce training time. This approach significantly reduces the number of measurements required compared to traditional Nyquist rate AMR methods. Hence, this paper proposes combining CS with Bi-LSTM to enhance the accuracy of AMR classification with a limited number of measurements and reduce training time. The proposed network is compared to four existing architectures with higher classification accuracy.

TABLE I. LIST OF ABBREVIATIONS

Abbreviations	The Full Name
5G	Fifth-generation
CS	Compressed sensing
LB	Likelihood-based
ALRT	Average likelihood ratio test
HLRT	hybrid likelihood ratio test
MIMO	Multiple input multiple output
SNR	Signal-to-noise ratio
CNN	Convolutional neural networks
ResNet	Residual network
MCLCNN	Multi-channel large kernel CNN
MSCL	Meta-supervised contrastive learning
BPSK	Binary phase shift keying
BFSK	Binary frequency shift keying
AM-DSB	Amplitude modulation double-sided band
CR	Compression ratio
AI	Artificial intelligence
Bi-LSTM	Bidirectional long short-term memory networks
DL	Deep Learning
GLRT	Generalized likelihood ratio test
ML	Machine learning
FSK	Frequency shift keying
QAM	Quadrature amplitude modulation
RNN	Recurrent neural network
DenseNet	Densely connected network
RepCCNet	Reparameterization causal convolutional network
MCLHN	Masked contrastive learning with hard negatives
QPSK	Quadrature phase shift keying
CPFSK	Continuous-phase FSK
RIP	Restricted isometry property
SDA	Stacked denoising autoencoder
AMR	Automatic modulation recognition
SDR	Software defined radio
LFs	Likelihood functions
PDF	Probability density function
LIDAR	Light Detection and Ranging
PSK	Phase shift keying
LSTM	Long short-term memory
DNN	Deep neural network
CLDNN	Convolutional long short-term DNN
TSTR	Two-stream transformer
NCS	Nonuniform compressive sampling
CSI	Channel state information
WB-FM	Wide band frequency modulation
CO	Convex optimization

III. SIGNAL MODEL

In this paper, we propose utilizing DNN classifiers to identify the modulation type of wireless signals. These classifiers can be trained to incorporate features extracted from training datasets. We provide a popular formula for the delivered complex envelope as follows:

$$r(t) = s(t;u_i) + n(t), \tag{1}$$

where:

$$s(t;u_i) = a_i e^{j2\pi\Delta f t}\, e^{j\theta} \sum_{k=1}^{K} e^{j\Phi_k} s_k^{(i)} g(t-(K-1)T-\varepsilon T),\ 0 \le t \le KT \tag{2}$$

where $s(t;u_i)$ is the delivered signal's noise-free baseband complex envelope. $n(t)$ is the instant channel noise at moment t. a_i is the unknown signal amplitude, Δf is the carrier frequency offset. θ is the time-invariant carrier phase introduced by the propagation delay. Φ_k is the phase jitter. s_k^i , $0 \le k \le K$ defines K complex symbols collected from the i^{th} modulation format, T is the symbol period, ε is the normalized epoch for time offset between the transmitter and receiver.

$g(t) = P_{pulse}(t) \oplus h(t)$, $h(t)$ is the composite effect of the residual channel with $h(t)$ channel impulse response and denoting mathematical convolution, and $P_{pulse}(t)$ is the transmit pulse shape.

$u_i = \left\{a_i, \Delta f, \theta, \varepsilon, g(t), \{\emptyset_k\}_{k=1}^{K}, \left\{s_k^{(i)}\right\}_{k=1}^{K}\right\}$ is the multidimensional vector with deterministic unknown signal or channel parameters for the i_{th} modulation category. Finally, our target is identifying the modulation pattern i from the delivered signal.

A. Dataset Details

In this paper, we use the RadioML 2016.10b dataset, which is sparse, created in Ref. [19] as both our training and testing datasets. O'shea and West [44] thoroughly explain how the data was generated. Fig. 1 illustrates the process's summary.

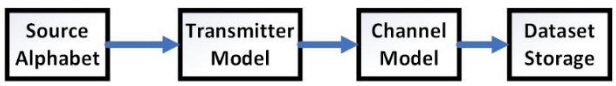


Fig. 1. Data generation framework.

To ensure equiprobable symbols and bits for digital modulations, the works of Shakespeare in ASCII are utilized as the input data. This is because Shakespeare's works contain a wide variety of characters, which can be used to represent symbols and bits of digital modulation. The ASCII code represents the symbols and bits, and although the original distribution may not be uniform, the data is then whitened using randomization techniques to ensure that the symbols and bits are equiprobable, meaning each has an equal chance of being chosen. A continuous voice signal is used as the input data for analog modulations. This signal mainly contains acoustic voice speech, with some interludes and off times. Specifically, these consist of Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), 8PSK, quadrature amplitude modulation QAM16, QAM64, Binary Frequency Shift Keying (BFSK), Continuous-Phase Frequency shift Keying (CPFSK), and pulse amplitude modulation PAM4 for digital modulations, and Wide Band Frequency Modulation (WB-FM), and Amplitude Double-Sided Band (AM-DSB) for analog modulations. Digital modulations are used to represent symbols and bits, while analog modulations are used to represent continuous voice signals. The modulation types chosen are chosen based on their ability to accurately represent the data and ensure that the symbols and bits are equiprobable. The RadioML 2016.10b datasets are divided evenly among the different modulation types. It contains two types of simulated channels: environmental noise, such as thermal noise and multipath fading, and device shortcomings, including sample rate offset, noise model, center frequency offset, and fading model. To obtain the finalized datasets, the output stream is first randomly divided into vectors created by down sampling the original data at a rate of 1M samples per second. Then, a sliding window of the same size as that used for acoustic waves is employed to extract 128 samples with a shift of 64 samples. This approach results in a dataset with 160,000 samples generated by the GNU-radio library originated in Ref. [44]. These samples are then divided into training and testing datasets. A rectangular windowing with 128 samples is used to form both datasets. The training examples, each consisting of 128 samples, are fed into the neural network as 2 × 128 vectors with real and imaginary components split in complex time samples, except for the original LSTM structure, which is fed in polar form (amplitude and phase). The input data is labeled with both the true SNR and modulation type. The SNR values of the samples are evenly spaced between 20dB and +18dB, with 2dB step size. This means that the data set is equally represented from all SNR values in the $-20, -18, -16, \ldots, 16, 18$ range. The percentage of samples from the testing data set that were successfully categorized is used to calculate classification accuracy.

B. LSTM

RNNs are a particular form of neural network appropriate for processing data with a temporal component. RNNs are enhanced with the idea of recurrent connections (also known as recurrent edges) to feed information back into previous levels (or into the same layer), unlike a normal multilayer network with feed-forward connections. This enables RNNs to capture the long-range time dependencies of the input data and use past knowledge to process the present input. When input is passed through a simplified RNN with a chain-like design (also known as memory cells), the node processes the activation outcomes of the present input vector and the prior state of the hidden nodes at each time step. This mechanism permits RNNs to capture the temporal dependencies of the input data, which is especially useful for tasks such as natural language processing, speech recognition, and time series forecasting. RNNs are particularly useful for tasks that require the model to remember information from the past. For example, in natural language processing, the model needs to remember the context of the sentence to predict the next word accurately. In speech recognition, the model must remember the previous words to recognize the current word accurately. In time series forecasting, the model must remember past values to predict future values accurately.

RNN [45] are among the most widely used neural networks for learning persistent features from time series data. LSTM is sub RNN type which is adept at recognizing long-term dependencies. Fig. 2 displays the block diagram of an LSTM cell. The core idea of an LSTM network is a memory cell, allowing the network to keep its state across different times. The LSTM layer at each time-step handles the information of the cell state by upgrading it with four main parts: an input gate to control how much of the cell state should be updated, a forget gate to decide how much of the cell state should be forgotten (or restart), a cell candidate to gather new information to the cell state, and an output gate to regulate how much of the cell state should be amended to the hidden state. The hidden state stores the output of the LSTM layer at that particular time-step, while the cell state retains information from prior time-steps.

In Fig. 2, the current moment can be represented by t. From the figure, the inputs of the memory cell are the input feature sequence x_t and the outcome sequence of the prior time h_t. Additionally, from the forgetting gate, The forgetting factor f_t is obtained as follows:

$$f_t = \sigma\left(W_f . [h_{t-1}, x_t] + b_f\right), \tag{3}$$

where h_{t-1}, b_f is the offset matrix, W_f is the connection matrix of x_t. σ is the sigmoid activation function, used to control the information-passing rate. It is mathematically formulated as follows:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{4}$$

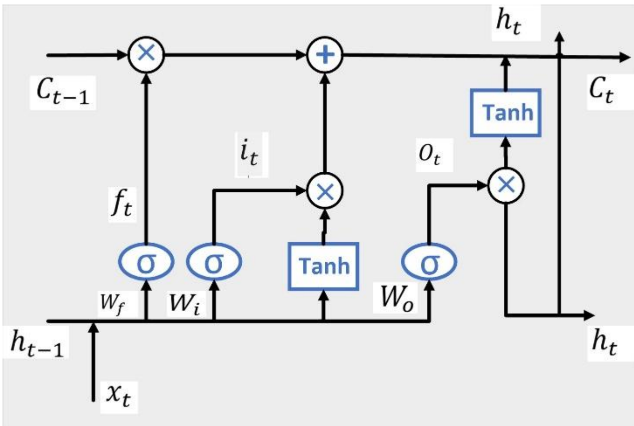


Fig. 2. LSTM.

The final value of σ is between 0 and 1. The input gate and memory status update information are

$$i_t = \sigma(W_i . [h_{t-1}, x_t] + b_t) \tag{5}$$

$$\widetilde{C}_t = tanh(W_c . [h_{t-1}, x_t] + b_c) \tag{6}$$

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C_t} \tag{7}$$

where $tanh$ is an activation function that creates potential values $\widetilde{C_t}$ and i_t is the output of the input gate. Also, the formula to determine the memory state $\widetilde{C_t}$ includes $\widetilde{C_t}$.

The most crucial part of these components is the memory state $\widetilde{C_t}$, which transfers data across the entire link while maintaining consistency. This guarantees that the information remains accurate for a prolonged duration. The output gate controlling element o_t, which is stated below, decides whether information h_t should be emitted or not.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_0) \tag{8}$$

$$h_t = o_t \times \tanh(C_t) \tag{9}$$

The output gate weight matrix is represented by W_o and the offset matrix is denoted as b_0. Compared to C_t, h_t contains more details about the present moment, making it a representation of short-term memory while C_t is a representation of long-term memory.

C. Compressed Sensing

Recently, compressive sensing has demonstrated great potential as an efficient technique for signal acquisition. The concept of CS is based on the idea that signals can be sampled at much lower rates than the Nyquist limit, while still allowing for recovery of sparse signals from these low-rate measurements [46]. Experimental results have shown that this is indeed possible.

In fact, CS can be applied to signals that are not precisely sparse but can be represented compactly enough using a sparse basis in some signal representation domains. This is because CS main concept relies on efficient representation of signals using a sparse basis. This means that instead of using a large number of coefficients to represent a signal, only a few coefficients are used to describe the same signal. This makes it possible to reduce the amount of data that needs to be transmitted or stored while still preserving the signal's essential characteristics. The wide spread of CS is because most communications and multimedia signals fall under the sparse or nearly sparse signals category and can consequently be exploited through the CS theory. This is because these signals can be represented sparsely, reducing the amount of processed data possible. Furthermore, CS theory also makes it possible to recover signals from incomplete or noisy data, especially useful in communication systems.

In addition, CS theory can also be used to improve the performance of signal processing algorithms. For example, CS theory can be used to reduce the computational complexity of algorithms by exploiting the sparsity of signals. This makes it possible to reduce the processing time and resources needed to handle signals, which can be especially beneficial in realtime applications. Overall, CS theory has become increasingly popular due to its ability to efficiently represent and process signals. This makes it possible to reduce the amount of data that needs to be transmitted or stored while still preserving the signal's essential characteristics. Furthermore, CS theory can also be used to improve the performance of signal processing algorithms, making it a powerful tool for various applications.

Consider signal $\boldsymbol{x}$ of length N (i.e., $\boldsymbol{x} \in R^N$), which can be expressed as sparse on a basis Ψ using techniques like the inverse Wavelet Transform (WT), inverse Discrete Cosine Transform (DCT), and inverse Discrete Fourier Transform (DFT). This signal can be represented as:

$$\boldsymbol{x} = \boldsymbol{\Psi}\mathbf{f}, \tag{10}$$

where $\mathbf{f}$ is sparse coefficients vector and $\boldsymbol{\Psi}$ is $N \times N$ matrix whose columns are orthonormal basis functions. We can claim that $\boldsymbol{x}$ is $k-$ sparse if only $k-$ non-zero coefficients are in the vector $\mathbf{f}$. By following the CS theory [46–48], the vector $\boldsymbol{x}$ can be precisely retrieved from M random observations over the measurement matrix as in Eq. (11).

$$\mathbf{y}_{CS} = \Phi \boldsymbol{x}, \tag{11}$$

where Φ is a $M \times M$ random measurement matrix (observation process) and $\mathbf{y}_{CS}$ is a M dimensional measurement vector with $M \ll N$. Typically, $M \geq const \times K \log\left(\frac{N}{K}\right)$ [46, 49]. Substituting Eq. (11) into Eq. (12) yields:

$$\mathbf{y}_{CS} = \Phi\boldsymbol{\Psi}\mathbf{f} = \mathbf{Af} \tag{12}$$

where $\mathbf{A}$ is a full rank sensing matrix and satisfies the Restricted Isometry Property (RIP) [49]. It has been shown that certain families of random matrices, such as those with appropriately dimensioned i.i.d. Gaussian or i.i.d. Bernoulli components, are highly likely to satisfy the RIP.

The main challenge of CS is to find a way to retrieve the initial signal $\boldsymbol{x}$ from the observed vector $\mathbf{y}_{CS}$ as stated in Eq. (12). Linear inverse problems have been extensively studied in engineering and mathematics. Since these problems are usually under-determined, extra regularizing constraints must be applied to obtain meaningful results. Sparsity constraints have become a popular type of regularization [50].

Now, assume that $\mathbf{f}$ is a sparse vector. By resolving: It is fairly simple to retrieve $\mathbf{f}$ from knowledge of $\mathbf{y}_{CS}$ as follows.

$$\hat{f} = \mathbf{argmin} \; \|\boldsymbol{f}\|_{\mathbf{0}} \quad s.t \;\; \mathbf{y}_{CS} = \mathbf{Af} \tag{13}$$

Because of its inevitable combinatorial search, this algorithm is NP-hard [50, 51]. The primary idea of [52] was to replace the l_0-norm with the adjacent l_1 convex norm. This results in the following minimization issue, known as Basis Pursuit [52]:

$$\hat{f} = \mathbf{argmin} \; \|\boldsymbol{f}\|_{\mathbf{1}} s.t \;\; \mathbf{y}_{CS} = \mathbf{Af} \tag{14}$$

Numerous recovery algorithms have been suggested lately, such as Convex Optimization (CO) and greedy algorithms. CO works by transforming the non-convex problem into a convex one in order to find an approximated solution, like basis pursuit [53], iterative splitting and thresholding [54], sparse reconstruction via separable approximation [55], and gradient projection for sparse reconstruction [56]. However, convex programming methods have high computational complexity, so much work has been done to reduce this complexity. Greedy algorithms have been proposed for CS reconstruction and are computationally simpler than convex optimization, such as matching pursuit [57] and orthogonal matching pursuit [58]. However, the reconstruction quality is lower than that of convex optimization.

In recent years, with the enormous success of deep learning, various techniques have been proposed for image compressive sensing reconstruction [59–63]. For instance, in Ref. [62], the authors proposed a Stacked Denoising Autoencoder (SDA), which can capture the statistical relations between various signal elements and thus refine signal recovery performance. Though, SDA has considerable computational complexity as the signal's dimension rises. The architecture of ReconNet, proposed by the authors in Ref. [63], contains complete connections between each successive layer. This weight-sharing method reduces computational complexity, allowing for faster CNN based reconstruction. Similarly, Zhang *et al.* [61] proposed ISTA-Net, inspired by iterative shrinkage-thresholding for CS reconstruction. Bo et al. [60] utilized a CNN configuration (FompNet) as a post-processing step for their fast orthogonal matching pursuit algorithm. These DL-based approaches are faster than conventional image CS ones thus motivated us to focus on employing DL-based algorithms for compression and recovery of original images.

IV. PROPOSED CS-BI-LSTM

The flowchart in Fig. 3 outlines our proposed CS-Bi-LSTM network model for AMR. The model is composed of four main components: an input layer, CS layers, Bi-LSTM layers, and a fully connected layer. The first step is to divide the received signal into I and Q channels. These signals are then fed into the network for processing. The initial stage is to compress the input using CS to reduce its dimensionality and consequently speed up the training time of the Bi-LSTM. Afterwards, the compressed signal is passed through the Bi-LSTM layers which extract features from it. Finally, we obtain an identified signal type.

A. Compressed Sensing Layer

The initial step of the suggested CS-Bi-LSTM framework is a compression network, as depicted in Fig. 4. This network takes in the I and Q channels as inputs and utilizes M neurons Fully Connected (FC) layer to shrink the size of the input vector. The CS sampling matrix Φ is viewed as the weights of this FC layer. Initially, its values are randomly initialized at the start of training. Unlike traditional CS methods that use fixed random projections, our model allows Φ to be learned during training. Specifically, the weights of this dense layer are updated iteratively through backpropagation during each training epoch, allowing the model to learn an optimal projection that enhances downstream classification performance [64, 65]. This sampling network learns the sampling matrix within training based on the I and Q channels provided. This process allows more vector structure information to exist in the CS observations due to utilizing characteristics from the input vector.

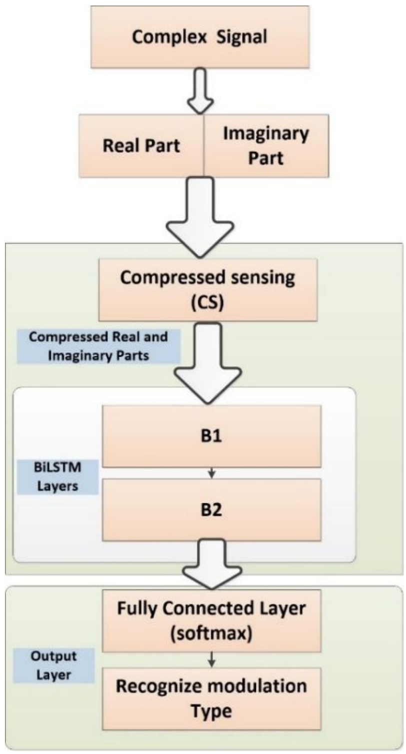


Fig. 3. CS-Bi-LSTM flow chart system model.

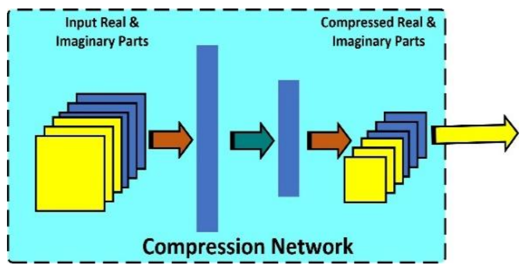


Fig. 4. Compression network.

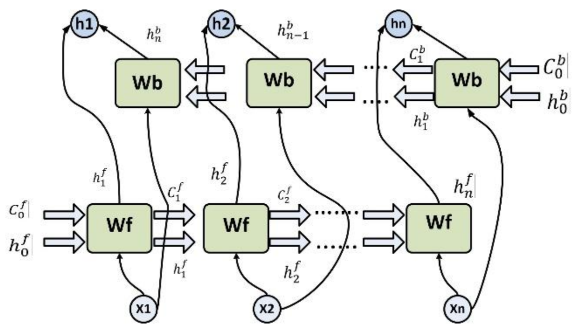


Fig. 5. Bi-LSTM Structural graph.

B. Bi-LSTM

The next step after the compressed sensing network is to employ Bi-LSTM model used to capture the contextual information within the input vectors and extract their features. As depicted in Fig. 2, Long Short-Term Memory (LSTM) only considers the data from the previous moment. Conversely, according to Huang *et al.* [66], Bidirectional LSTM (Bi-LSTM) considers both the former and subsequent moments. The structural operation graph of Bi-LSTM is presented in Fig. 5, where the Bi-LSTM takes the input sequence and reverses it before processing it similar to LSTM. Combining the forward and reverse LSTMs results in the final product, which enables consideration of contextual information.

where h_t is the final product of the Bi-LSTM, where $t = 1, 2, \dots, n$. Fig. 6 illustrates the combination of forward and reverse components to produce h_t. Hence, h_t is mathematically expressed as follows:

$$h_t = \left[h_t^f, h_t^b\right] \tag{15}$$

The fully connected layer maps the output attributes of the Bi-LSTM into a sparse space. The algorithm outputs the classification probability of the relevant modulation patterns after the network has been trained.

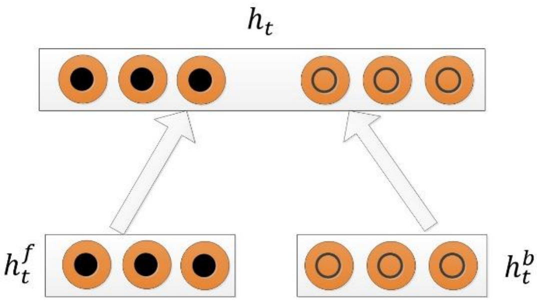


Fig. 6. Relation between h_t, h_t^f, and h_t^b.

V. Results and Discussions

Herein, experiments have been conducted to assess the performance of the suggested CS-Bi-LSTM framework. We present our experimental results and their analyses. Our CSBi-LSTM is compared with four different architectures that are more accurate than the CNN in Ref. [19], including: the CNN from Ref. [19], CLDNNs for modulation recognition, and optimized versions of ResNet from Ref. [67] and LSTM from Ref. [37]. Firstly, we compare our proposed framework with the mentioned networks without using CS. Then, we repeat the experiment by comparing our CS-Bi-LSTM with various existing networks. For a fair comparison, we combine CS with all the networks mentioned, then compare the results to our proposed model.

A. Experiment Setup [1]

Dataset Details: The proposed framework is trained and tested using the RadioML2016.10b sparse dataset. It consists of various digital and analog modulations, including BPSK, QPSK, 8PSK, QAM16, QAM64, BFSK, CPFSK, PAM4, WBFM, and AM-DSB. This provides a comprehensive set of modulations to test the framework's performance. The dataset is also well-structured and organized, making it easy to use. The details of the dataset are provided in Table II, which includes information such as the modulation type, SNR, and the number of samples. This makes it easy to select the appropriate dataset for training and testing. In addition, the dataset is publicly available, which makes it easy to access and use. This is important for research purposes, as it allows researchers to compare their results with those of other researchers easily.

TABLE II. Dataset Parameters.

Parameter	Value/Scheme
Modulations Number	11
Modulation formats	BPSK, QPSK, 16QAM, 8-PSK,64QAM, 4PAM, GFSK, CPFSK, WBFM, AM-SSB, AM-DSB
Signal length	128 complex samples
SNR range	−20:2:20
Total number of signals	165000
Channel specification	Selective multipath channel sample rate offset center frequency offset AWGN

TABLE III. Network Details

Network	Layer	Activation	Size
3 × CS network	Input	/	2×128×3
	Fully connected layer	Relu	(Batch size×768) ×1
	Fully connected layer	Relu	CR×(Batch size×768) ×1
4× Bi-LSTM	Bi-LSTM	ReLU	CR× (Batch size ×768) ×1
	Bi-LSTM	/	CR×(Batch size×768)×1
	Fully connected layer	Relu	
	Dense	Relu	6

Training Details: We employed Keras with TensorFlow as the backend for all our experiments. An 8 Tesla P100 GPUs server with 16 GB of memory was used for this purpose. All architectures employed the Adam optimizer, and the loss function was the category cross entropy function. All layers used ReLu activation functions, except for the final dense layer, which used a Softmax activation function. Except for the LSTM, which used a batch size of 400 and a learning rate of 0.0018, all designs had a batch size of 1024 and a learning rate of 0.001. Table III summarizes the network structure of both CS and Bi-LSTM networks. Besides, we used an early stopping approach to prevent the networks from being overfitted as the input vector is one dimension with the size equal to the product of compression ratio multiplied by Batch size multiplied by 768.

[1] Simulation codes are provided in [68]

Implementation Details: We first compare the performance of Bi-LSTM with the benchmark networks. In this experiment, CS is not combined with Bi-LSTM. This experiment aims to show the ability of using Bi-LSTM to classify modulation techniques. Then, we combine CS with the benchmark network and compare the performance with our proposed CS-Bi-LSTM network under various compression ratios, including 50%, 25%, 12.5%, and 6.25%.

Benchmarks Comparisons: The performance of our proposed framework is validated by comparing it with the existing state-of-the-art benchmarks. Specifically, we compare our proposed framework with existing Convolutional Neural Network (CNN) [36], Convolutional Long Short-term Memory

Deep Neural Network (CLDNN) [36], LSTM [69], and Deep ResNet [33]. Herein, we have used the same compression ratio for all compared models for fair comparison.

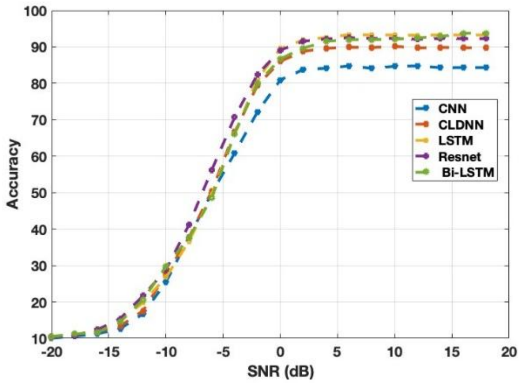


Fig. 7. AMR accuracy rates of CNN, CLDNN, LSTM, Resnet, and Bi-LSTM networks against different SNRs without CS.

B. Experiments on Effectiveness of Our Proposed Network C. Without Using CS

As shown in Fig. 7, we conclude that the CS-Bi-LSTM network's accuracy from −20 to 20 dB is significantly higher than that of the competing recognition techniques. The accuracy of the Bi-LSTM model is influenced by its primary structure. This is due to its ability to extract bidirectional time series features, effectively addressing gradient explosion and gradient vanishing issues. Additionally, Fully Connected (FC) deep neural networks successfully perform feature classification [70]. Our proposed deep neural network takes advantage of these networks' complementary and synergistic nature to extract and classify spatiotemporal features. This approach solves the limitation faced by traditional networks, which only extract single spatial or temporal features from sample signals [70].

D. CS-Bi-LSTM

The combination of CS and Bi-LSTM as suggested in section IV is a promising approach to improve the accuracy of existing AMR networks. By combining CS with Bi-LSTM, we can leverage the strengths of both techniques to create a more robust and accurate network. The CS technique decreases the overall network parameters, while the Bi-LSTM technique allows us to capture long-term dependencies in the data.

We tested both proposed and existing networks under different compression ratios of 50%, 25%, 12.5%, and 6.25%, which sample the wide-band signals at various values. The simulation results, shown in Figs. 8−10, and11 demonstrate that our proposed CS-Bi-LSTM has higher accuracy than the existing AMR networks across all compression ratios. This is because the CS-Bi-LSTM network can capture more complex patterns in the data due to its increased capacity. Additionally, the CS technique allows us to reduce the number of parameters in the network, which helps to reduce overfitting and improve.

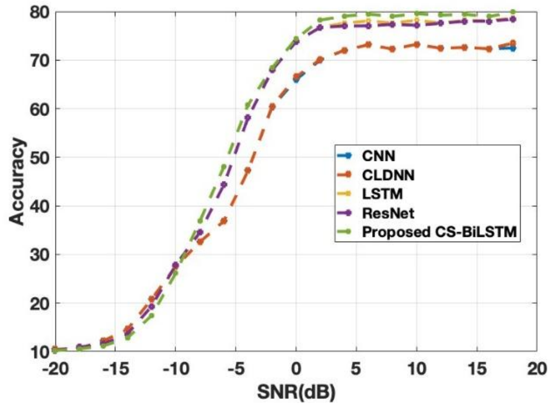


Fig. 8. AMR accuracy rates of CNN, CLDNN, LSTM, Resnet, and Bi-LSTM networks against different SNRs under 50% compression ratio.

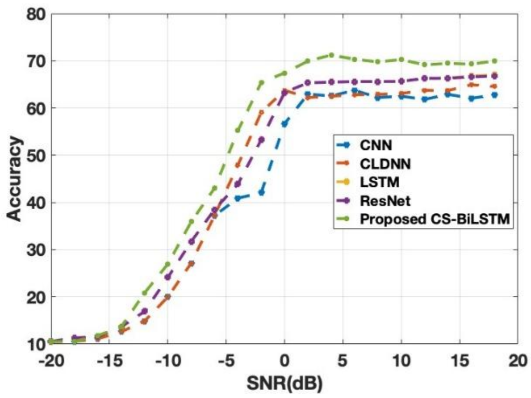


Fig. 9. AMR accuracy rates of CNN, CLDNN, LSTM, Resnet, and Bi-LSTM networks against different SNRs under 25% compression ratio.

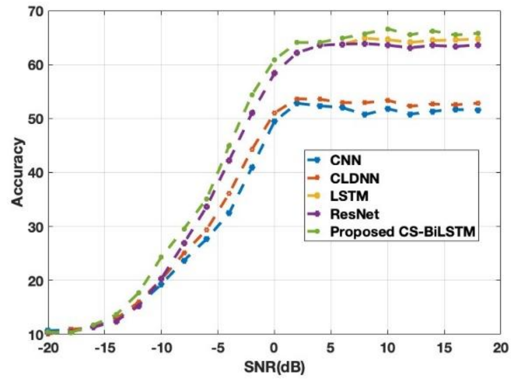


Fig. 10. AMR accuracy rates of CNN, CLDNN, LSTM, Resnet, and Bi-LSTM networks against different SNRs under 12.5% compression ratio.

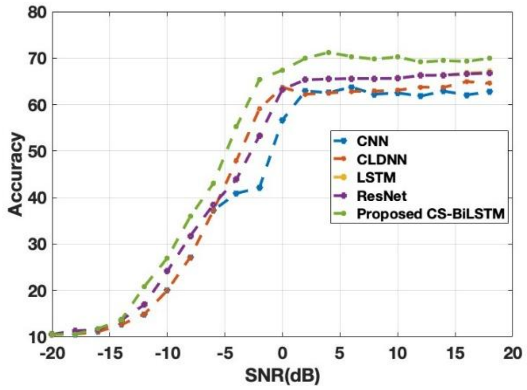
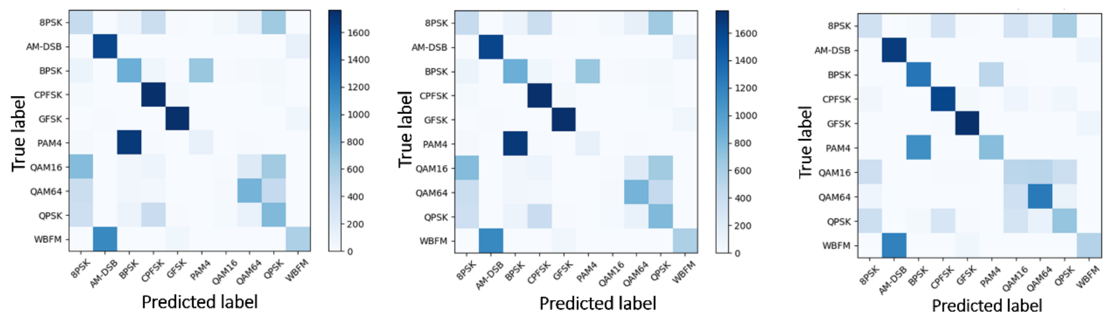


Fig. 11. AMR accuracy rates of CNN, CLDNN, LSTM, Resnet, and Bi-LSTM networks against different SNRs under 6.25% compression ratio.

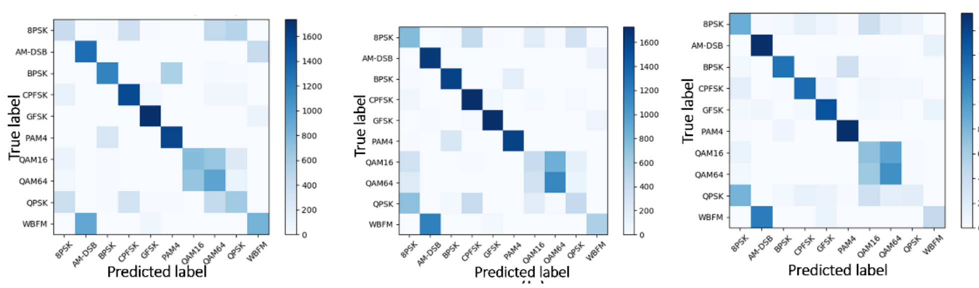
Generalization. We observe a comparable performance between LSTM and Bi-LSTM, surpassing that of alternative models. However, as the original dataset undergoes compression, the performance of LSTM exhibits a diminishing trend with escalating compression ratios. Conversely, Bi-LSTM experiences a marginal decrease in performance. This discrepancy can be attributed to the inherent advantage of Bi-LSTM, which processes sequential data bidirectionally (forward and backward), mitigating the impact of compression on its overall performance. The capacity of Bi-LSTM to operate in both forward and backward directions serves to constrain the adverse effects of compression, thus contributing to its resilience in such scenarios.

Hence, combining CS and Bi-LSTM is a promising solution to improve the accuracy of existing AMR networks. Simulation results demonstrate that our suggested CS-Bi-LSTM configuration is more accurate than the existing AMR networks across all compression ratios, which suggests that this approach is efficacious.

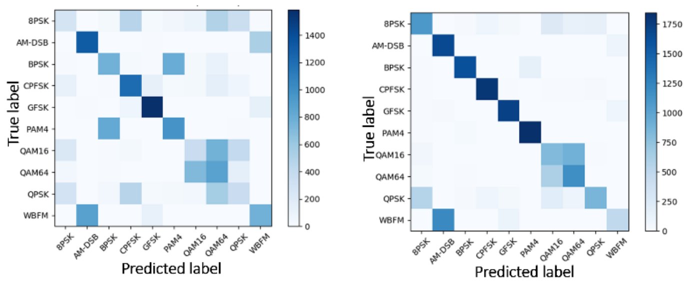


(a) (b) (c) (d) (e)

Fig. 12. The confusion matrix for our proposed framework and the existing various state-of-the-art networks with CR = 25%: (a) CNN, (b) CLDNN (c) LSTM (d) Resnet (e) Our proposed framework.



(a) (b) (c)



(d) (e)

Fig. 13. The confusion matrix for our proposed framework and the existing various state-of-the-art networks with CR = 50%: (a) CNN, (b) CLDNN (c) LSTM (d) Resnet (e) Our proposed framework.

The confusion matrices in Figs. 12 and 13 compare the performance of our proposed framework and the existing networks under compression ratio of 25% and 50%, respectively. Our proposed framework can accurately recolonize the modulation type at lower SNR and lower compression ratios than the existing framework.

As shown in Fig. 12, at compression ratio 25% with SNR = 18 dB, our proposed CS-Bi-LSTM framework can achieve a higher accuracy rate than the existing framework. indicating its efficiency and reliability as a valuable tool for modulation type identification in the future.

VI. Conclusion

This paper introduces the CS-Bi-LSTM network, a novel framework for Automatic Modulation Recognition (AMR), capable of accurately identifying 11 distinct types of modulated signals. The proposed architecture integrates Compressive Sensing (CS) layers to compress the input I and Q channels, followed by Bidirectional Long Short-Term Memory (Bi-LSTM) layers to extract contextual information and model temporal dependencies. This combination enables the network to effectively capture the temporal dynamics of signals, significantly enhancing recognition accuracy. To validate its performance, the CS-Bi-LSTM network is rigorously evaluated against state-of-the-art algorithms, including CNN, CLDNN, LSTM, and ResNet. The results demonstrate that the proposed network achieves superior recognition accuracy under high SNR conditions across various compression ratios. The CS layers play a critical role in reducing the volume of processed data, thereby improving computational efficiency without compromising performance. The CS-Bi-LSTM network represents a significant advancement in AMR, offering a robust and efficient solution for applications where accuracy and computational efficiency are paramount. Future research directions include extending this framework to support a broader range of modulation types, collecting our own over-the-air AMR dataset under varying SNRs and interference conditions for further validation, optimizing the network for real-time processing in dynamic environments, and exploring its applicability in next-generation wireless communication and signal processing. Additionally, further investigation into adaptive compression techniques and hybrid deep learning architectures could enhance the network's scalability and generalization capabilities. These advancements would solidify the CS-Bi-LSTM network as a versatile and powerful tool for next-generation AMR systems.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Hossam Kassem and Sherief Hashima conducted the research; Haithem Khallaf analyzed the data; Hossam Kassem and Sherief Hashima wrote the paper; all authors revised it; all authors had approved the final version.

Funding

This study is supported by JSPS KAKENHI grant number JP22H03649, Japan.

References

[1] Y. Feng, R. Duan, S. Li, P. Cheng, and W. Liu, "A dual-branch network with feature assistance for automatic modulation recognition," *IEEE Signal Processing Letters*, pp. 1–5, 2025.

[2] B. Wang, Z. Yuan, A. Li, J. Lu, and X. Zhang, "Hybrid driven model fusing deep learning and knowledge for automatic modulation recognition," *IEEE Internet of Things Journal*, p. 1, 2024.

[3] T. Fang, Q. Wang, L. Zhang, and S. Liu, "Modulation mode recognition method of non-cooperative underwater acoustic communication signal based on spectral peak feature extraction and random forest," *Remote Sensing*, vol. 14, no. 7, 1603, 2022.

[4] C. Xiao, S. Yang, Z. Feng, and L. Jiao, "MCLHN: Toward automatic modulation classification via masked contrastive learning with hard negatives," *IEEE Transactions on Wireless Communications*, vol. 23, no. 10, pp. 14304–14319, 2024.

[5] D. H. A. Nuaimi, I. A. Hashim, I. S. Z. Abidin, L. B. Salman, and N. A. M. Isa, "Performance of feature-based techniques for automatic digital modulation recognition and classification — A review," *Electronics*, vol. 8, no. 12, 1407, 2019.

[6] F. L. Luo, *Neural Networks for Signal Intelligence*, pp. 243–264. 2020.

[7] S. Ying, S. Huang, S. Chang, Z. Yang, Z. Feng, and N. Guo, "A convolutional and transformer based deep neural network for automatic modulation classification," *China Communications*, vol. 20, no. 5, pp. 135–147, 2023.

[8] K. Zhang, L. Xu, Z. Feng, and P. Zhang, "A novel automatic modulation classification method based on dictionary learning," *China Communications*, vol. 16, no. 1, pp. 176–192, 2019.

[9] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: Classical approaches and new trends," *IET communications*, vol. 1, no. 2, pp. 137–156, 2007.

[10] M. Leblebici, A. Çalhan, and M. Cicioglu, "CNN-based automatic modulation recognition for index modulation systems," *Expert Systems with Applications*, vol. 240, 122665, 2024.

[11] S. Mohsen, A. M. Ali, and A. Emam, "Automatic modulation recognition using CNN deep learning models," *Multimedia Tools and Applications*, vol. 83, no. 3, pp. 7035–7056, 2024.

[12] H. Han, Z. Yi, Z. Zhu, L. Li, S. Gong, B. Li, and M. Wang, "Automatic modulation recognition based on deep-learning features fusion of signal and constellation diagram," *Electronics*, vol. 12, no. 3, 552, 2023.

[13] D. Yi, D. Wu, and T. Hu, "A lightweight automatic modulation recognition algorithm based on deep learning," *IEICE Transactions on Communications*, vol. 106, no. 4, pp. 367–373, 2023.

[14] B. Tolba, M. Elsabrouty, M. G. A. Aguye, H. Gacanin, and H. M. Kasem, "Massive MIMO CSI feedback based on generative adversarial network," *IEEE Communications Letters*, vol. 24, no. 12, pp. 2805–2808, 2020.

[15] R. M. A. Makhlasawy, H. S. Ghanem, H. M. Kassem, M. Elsabrouty, H. F. Hamed, F. E. A. E. Samie, and G. M. Salama, "Deep learning for wireless modulation classification based on discrete wavelet transform," *International Journal of Communication Systems*, vol. 34, no. 18, e4980, 2021.

[16] S. Ramjee, S. Ju, D. Yang, X. Liu, A. E. Gamal, and Y. C. Eldar, "Fast deep learning for automatic modulation classification," arXiv preprint arXiv:1901.05850, 2019.

[17] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed lowcost spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, 2018.

[18] Q. Zhou, J. Qian, J. Tang, and J. Li, "Deep unrolling networks with recurrent momentum acceleration for nonlinear inverse problems," *Inverse Problems*, vol. 40, no. 5, 055014, 2024.

[19] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. International Conference on Engineering Applications of Neural Networks*, 2016, pp. 213–226.

[20] J. Bai, W. Zhu, S. Liu, C. Ye, P. Zheng, and X. Wang, "A Temporal Convolutional Network–Bidirectional Long Short-Term Memory (TCNBiLSTM) prediction model for temporal faults in industrial equipment," *Applied Sciences*, vol. 15, no. 4, 2025.

[21] S. Natha, F. Ahmed, M. Siraj, M. Lagari, M. Altamimi, and A. A. Chandio, "Deep BiLSTM attention model for spatial and temporal anomaly detection in video surveillance," *Sensors (Basel, Switzerland)*, vol. 25, 2025.

[22] J. Zhang, D. Cabric, F. Wang, and Z. Zhong, "Cooperative modulation classification for multipath fading channels via expectation maximization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6698–6711, 2017.

[23] Y. Kumar, M. Sheoran, G. Jajoo, and S. K. Yadav, "Automatic modulation classification based on constellation density using deep learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1275–1278, 2020.

[24] E. Nachmani, Y. Bachar, E. Marciano, D. Burshtein, and Y. Be'ery, "Near maximum likelihood decoding with deep learning," arXiv preprint arXiv:1801.02726, 2018.

[25] A. Ali and F. Yangyu, "Unsupervised feature learning and automatic modulation classification using deep learning model," *Physical Communication*, vol. 25, pp. 75–84, 2017.

[26] T. J. O'Shea, N. West, M. Vondal, and T. C. Clancy, "Semi-supervised radio signal identification," in *Proc. 2017 19th International Conference on Advanced Communication Technology (ICACT)*, 2017, pp. 33–38.

[27] M. Abu-Romoh, A. Aboutaleb, and Z. Rezki, "Automatic modulation classification using moments and likelihood maximization," *IEEE Communications Letters*, vol. 22, no. 5, pp. 938–941, 2018.

[28] X. Sun, S. Su, Z. Zuo, X. Guo, and X. Tan, "Modulation classification using compressed sensing and decision tree support vector machine in cognitive radio system," *Sensors*, vol. 20, no. 5, 1438, 2020.

[29] J. Jiang, H. M. Kasem, and K. W. Hung, "A very deep spatial transformer towards robust single image super-resolution," *IEEE Access*, vol. 7, pp. 45618–45631, 2019.

[30] H. M. Kasem, K. W. Hung, and J. Jiang, "Revised spatial transformer network towards improved image super-resolutions," in *Proc. 2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 2688–2692.

[31] H. Kasem and M. Elsabrouty, "Perceptual compressed sensing and perceptual sparse fast Fourier transform for audio signal compression," in *Proc. 2014 IEEE 15th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2014, pp. 444–448.

[32] M. Li, O. Li, G. Liu, and C. Zhang, "An automatic modulation recognition method with low parameter estimation dependence based on spatial transformer networks," *Applied Sciences*, vol. 9, no. 5, 1010, 2019.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[34] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[35] S. Shaik and S. Kirthiga, "Automatic modulation classification using DenseNet," in *Proc. 2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 2021, pp. 301–305.

[36] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4580–4584.

[37] N. E. West and T. O'shea, "Deep architectures for modulation recognition," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2017, pp. 1–6, IEEE.

[38] A. Kumar, M. S. Chaudhari, and S. Majhi, "Automatic modulation classification for ofdm systems using bi-stream and attention-based CNN-LSTM model," *IEEE Communications Letters*, p. 1, 2024.

[39] X. Hu, Z. Cai, and R. Zhou, "Automatic modulation classification based on the multi-channel large-kernel convolutional neural network," in *Proc. 2023 IEEE 23rd International Conference on Communication Technology (ICCT)*, 2023, pp. 259–265.

[40] N. Tang, X. Wang, F. Zhou, S. Tang, and Y. Lyu, "Reparameterization causal convolutional network for automatic modulation classification," *IEEE Transactions on Vehicular Technology*, pp. 1–8, 2024.

[41] Y. R. M. Llermanos and G. C. Briones, "Automatic modulation classification for low-power IoT applications," *IEEE Latin America Transactions*, vol. 22, no. 3, pp. 204–212, 2024.

[42] J. Li, Q. Jia, X. Cui, T. A. Gulliver, B. Jiang, S. Li, and J. Yang, "Automatic modulation recognition of underwater acoustic signals using a two-stream transformer," *IEEE Internet of Things Journal*, p. 1, 2024.

[43] J. Zhao, H. Wang, S. Peng, and Y.-D. Yao, "Meta supervised contrastive learning for few-shot open-set modulation classification with signal constellation," *IEEE Communications Letters*, p. 1, 2024.

[44] T. J. O'shea and N. West, "Radio machine learning dataset generation with gnu radio," in *Proc. GNU Radio Conference*, 2016, vol. 1.

[45] A. Graves, "Long short-term memory," *Supervised Sequence Labelling with Recurrent Neural Networks*, pp. 37–45, 2012.

[46] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[47] J. Liu, W. Zhao, and S. Li, "Compressive sensing empirical wavelet transform for frequency-banded power measurement considering interharmonics," *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–12, 2025.

[48] L. Shi and G. Qu, "An improved reweighted method for optimizing the sensing matrix of compressed sensing," *IEEE Access*, vol. 12, pp. 50841–50848, 2024.

[49] E. J. Candes *et al.*, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9–10, pp. 589–592, 2008.

[50] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[51] J. Skilling and S. Gull, "Algorithms and applications," in *Maximum Entropy and Bayesian Methods in Inverse Problems*, 1985, pp. 83–132.

[52] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[53] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[54] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 57, no. 11, pp. 1413–1457, 2004.

[55] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, 2009.

[56] M. A. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.

[57] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[58] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[59] H. Lu and L. Bo, "WDLReconNet: Compressive sensing reconstruction with deep learning over wireless fading channels," *IEEE Access*, vol. 7, pp. 24440–24451, 2019.

[60] L. Bo, H. Lu, Y. Lu, J. Meng, and W. Wang, "FOMPNET: Compressive sensing reconstruction with deep learning over wireless fading channels," in *Proc. 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, 2017.

[61] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1828– 1837.

[62] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in *Proc. 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1336–1343, IEEE, 2015.

[63] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "ReconNet: Non-iterative reconstruction of images from compressively sensed measurements," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 449–458.

[64] Y. Wu, M. Rosca, and T. Lillicrap, "Deep compressed sensing," in *Proc. 36th International Conference on Machine Learning*, 2019, pp. 6850–6860, 2019.

[65] Z. Yin, Z. Wu, W. Shi, G. Hu, and W. Lin, "Video compressed sensing via wavelet residual sampling and dual-domain fusion," *IEEE Transactions on Multimedia*, pp. 1–16, 2025.

[66] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," arXiv preprint arXiv:1508.01991, 2015.

[67] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[68] H. Kassem. Compressed-Automatic-Modulation-Recognition-Deep-Learning-Network. [Online]. Available: https://github.com/HossamMKasem/

[69] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM – A tutorial into long short-term memory recurrent neural networks," arXiv preprint arXiv:1909.09586, 2019.

[70] X. Zhang, Z. Luo, and W. Xiao, "CNN-BiLSTM-DNN-based modulation recognition algorithm at low SNR," *Applied Sciences*, vol. 14, no. 13, 2024.