Deep Learning Models Applied in Automatic Modulation Classification of Radar Signals

Pedro de Figueiredo Abissamra, Sarah Negreiros de Carvalho Leite, Renato Machado, and Dimas Irion Alves

***Abstract*— Artificial Intelligence in Electronic Warfare has gained prominence, particularly for Automatic Modulation Classification tasks. Deep learning methods have demonstrated robustness and high accuracy in addressing this challenge. This study proposed and tested Long Short-Term Memory and Convolutional Neural Network architectures for Automatic Modulation Classification in radar signals. The LSTM model achieved 90% accuracy in classifying eleven modulation types at -2.66 dB SNR, while the CNN model reached the same accuracy at 1.50 dB SNR. Although the LSTM outperformed the CNN, it required higher computational resources and longer latency.**

***Keywords*— AMC, Convolutional Neural Networks, Long Short-Term Memory, Radar Signals.**

I. Introduction

There has been a growing demand for automation in Electronic Warfare (EW) systems. To address this need, integrating Artificial Intelligence (AI) into EW systems has emerged as a solution [1]. AI's ability to provide effective decision support, manage large volumes of data, and enhance decision-making processes allows for improved self-control, self-regulation, and self-actuation in these military systems [2].

Automatic Modulation Classification (AMC) or Automatic Modulation Recognition (AMR) consists of identifying the modulation scheme of a received signal without any prior information. In a non-cooperative environment, such as in EW, this becomes a critical challenge, which makes the AMC crucial for achieving tactical superiority, especially when dealing with the AMC of radar signals [3]. The challenge of AMC for radar signals is becoming an urgent problem in electronic countermeasure systems. As the electromagnetic environment on the battlefield becomes increasingly complex, the parameters of radar signals are also evolving accordingly. Conventional techniques are more likely to have poorer performance and higher computational complexity, especially under low signal-to-noise ratio (SNR) conditions [4]. Due to the high accuracy and robustness capabilities of deep learning techniques, they have begun to be widely used to solve AMR tasks [5].

Different deep learning techniques have been used in the context of AMC for radar signals. One example is the application of a Convolutional Neural Network (CNN) Le-Net-5 to classify eight types of radar modulations, achieving 96.52% accuracy under a -2 dB SNR condition [6]. The work presented in [7] proposed using an AlexNet to classify twelve radar modulations. The network accomplished the task with 97.58% accuracy at an SNR of -6 dB. In another early study, a coordinate attention model was introduced to a ShuffleNet, which could also distinguish between twelve classes with 98.14% accuracy at an SNR of -8 dB [8].

Recurrent Neural Networks (RNN) have been proposed for different applications [9], [10]. The study in [9] used RNNs to classify, denoise, and deinterleave pulse streams, while [10] focused on classifying radar emitters. Additionally, the authors in [4] developed a novel method that combined a shallow Convolutional Neural Network (CNN) with a bidirectional Long Short-Term Memory (Bi-LSTM) network to address the modulation classification task for radar signals. This combined approach achieved an impressive 95% accuracy in classifying eight modulation types under -10 dB Signal-to-Noise Ratio (SNR) conditions. More recently, an LSTM network capable of classifying twenty-three distinct types of radar modulations was introduced, achieving 90% accuracy at -2 dB SNR [11].

These advancements highlight the growing prevalence of deep learning techniques in electronic warfare, particularly for radar signal classification. This paper aims to evaluate the performance of LSTM and CNN architectures for automatic modulation classification (AMC) of radar signals.

This paper is organized as follows. Section II describes the dataset, the deep learning architectures used, and how the performance evaluation was executed. Section III presents the results obtained for each tested model, as well as a discussion about those results. Section IV concludes the paper and presents future work to be accomplished.

II. Methodology

This work applied two deep learning models, namely a Long Short-Term Memory and a Convolutional Neural Network, in the Automatic Modulation Classification task. For each model, a baseline architecture was chosen as a reference to verify the performance of the proposed architectures. The dataset used was the DeepRadar2022, which will be described next.

A. *DeepRadar2022*

DeepRadar2022 is a dataset created by [11]. It consists of modulated In-Phase and Quadrature (IQ) radar signals sampled at 100MHz. It has 21 modulation classes plus a noise and a non-modulated class, totaling 23 balanced classes. All data was created using Matlab, and signals with different SNRs were created for each class. The SNR varies from -12 dB to 20 dB with a step of 2 dB, summing up 782000 different

Pedro de Figueiredo Abissamra, Sarah Negreiros de Carvalho Leite, Renato Machado, and Dimas Irion Alves are with the Electronic Warfare Laboratory, Aeronautics Institute of Technology, São José dos Campos-SP. E-mails: pedro.abissamra.101808@ga.ita.br, sarah.leite@gp.ita.br, {rmachado, dimasirion}@ita.br.

signals with a (1024×2) shape. They are already separated into 60% for training and 20% for test and validation. The full dataset can be found in [12].

In this paper, due to hardware limitations, a reduced dataset with 11 classes was used: Linear Frequency Modulation, Frequency Modulation with Costas Code, Binary Phase-Shift Keying, Phase Modulation with Barker code, Phase Modulation with Frank code, Phase Modulation with P1 code, Phase Modulation with P2 code, Phase Modulation with P3 code, Phase Modulation with P4 code, Non-Modulation and Complex White Gaussian Noise (noise). The total number of training data was 224400; and for test and validation, this amount was 74800.

B. *Long Short-Term Memory*

The LSTM baseline architecture (referred to here as LSTM 1) was proposed by [11]. It consists of three stacked LSTM layers, a dense layer, and the classification layer beside the input layer. Each LSTM layer comprises 128 cells; the fully connected (FC) layer uses a softmax activation function with 11 output neurons, and the input layer has a (1024×2) shape. The total number of parameters is 331.659.

The proposed LSTM architecture (referred to here as LSTM 2) is similar to the baseline architecture. The only difference is that this new one has four stacked LSTM layers with 64 cells each. The input and the dense layer are the same as the previous, as shown in Figure 1. This new network has a total of 116.939 parameters.

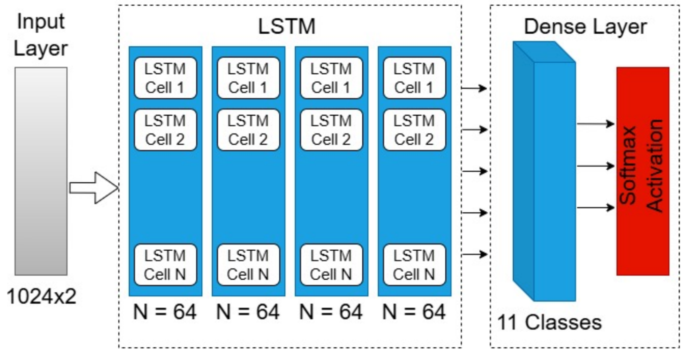

Fig. 1. Layers description of the proposed LSTM architecture (LSTM 2).

The training parameters were the same for both architectures. A total of 500 epochs were used with a batch size of 512 samples. The optimizer used was Adaptive Moment Estimation (ADAM) with a cyclical learning rate varying from 1.10^{-7} to 1.10^{-3}. These networks were trained on a personal computer equipped with an AMD Ryzen 7 3700x CPU, 32 GB RAM, and a 12 GB RTX3060 Nvidia GPU. The testing hardware was a server equipped with 104 x Intel Xeon Gold 5320 CPU and 1 TB of RAM.

C. *Convolutional Neural Network*

The CNN baseline architecture (referred to here as CNN 1) was proposed by [13]. It consists of seven one-dimensional convolution layers with 64 filters each and a kernel size of three. A max pooling layer is used between each convolution layer with a pool size of two. These layers are followed by two fully connected layers with 128 neurons and the output layer with 11 neurons. The ReLU activation function is used in all convolution layers. The dense layers employ a SELU activation function, whereas the output layer uses the Softmax activation function. The input layer has a (1024×2) shape, and the total number of parameters of CNN 1 is 158.155.

The proposed CNN architecture (CNN 2) consists of six one-dimensional convolutional layers, each with a distinct number of filters. In the first layer, there are 512 filters, and as the next layer arises, the number of filters is divided by two. This way, in the sixth layer, there are only 32 filters. As discussed in CNN 1, the kernel size is three, and a max pooling layer with a pool size of two is used between each convolutional layer. After the first and second max pooling layers, a dropout of 20% was inserted. Four fully connected layers were used. In the first layer, there are 256 neurons, while in the others, this number decays by a factor of two until 32 neurons in the last layer before the output, which has 11 neurons. A dropout rate of 20% was also applied after the first and second FC layers. The activation functions used were the same as in CNN 1. A description of this network is presented in Figure 2. The total number of parameters of the new network is 713.579.

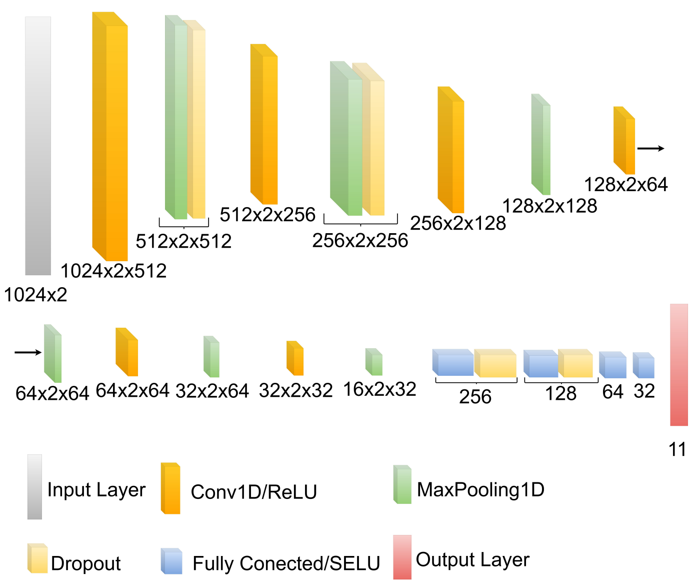

Fig. 2. Layers description of the proposed CNN architecture (CNN 2).

The same training parameters were used for both architectures. A total of 80 epochs were used with a batch size of 128 samples. The optimizer was ADAM with a cyclical learning rate varying from 1.10^{-7} to 1.10^{-3}. An early stopping monitor was also used with a patience value of 5 epochs, and the monitored parameter was the accuracy of the validation set. The training and testing hardware was a server equipped with 104 x Intel Xeon Gold 5320 CPU and 1 TB RAM (the same used for testing the LSTM).

D. Performance Verification

To gain a more comprehensive understanding of the models' and architecture's performance, both the baseline and the proposed networks were trained and tested. The LSTM network underwent three training and testing cycles due to its high computational demands and limited hardware availability. For the CNN network, five training and testing cycles were performed. After each cycle, all performance parameters (explained in Section III) were recorded, and their average values were calculated.

Once all tests were completed, a statistical analysis was conducted solely for the CNN models, given the limited observation quantity for the LSTM network, to ensure that the proposed CNN network performed better than the baseline method. The statistical test employed was the one-way ANOVA, with its approval criteria of a p-value of 0.05 or lower. In other words, if the calculated p-value of the performance parameters observations was equal to or lower than 0.05, an improvement with statistical significance could be noticed. To ensure that this evaluation method would be valid for these data (which respected a normal distribution), the Kolmogorov-Smirnov test was previously executed for all parameters in the evaluation, except for the number of epochs and latency per sample, which can vary depending on the computer in use.

III. Results and Discussion

The performance of all architectures was verified by their accuracy on the test set, number of epochs to train the model, latency (for a batch of 32 samples), and minimum (SNR) for a 90% accuracy. The SNR evaluation was done using two approaches. The first approach outputs the average accuracy (general accuracy) performance considering the assessment of all modulations, while the second considers each modulation separately. These approaches were tested for the LSTM and CNN architectures, as described and discussed throughout this section.

A. LSTM

The average results obtained for the LSTM 1 as well as for the LSTM 2 are presented in Table I. Observing these results, it is noticeable that, in general, the performance of the proposed model is similar to that of the baseline model, except for the latency per sample parameter, which is 27.03% lower in LSTM 2. This is a consequence of the reduced number of parameters compared to LSTM 1. However, although the total number of parameters of LSTM 2 is reduced by 65% compared to LSTM 1, the latency per sample does not follow this proportion. The probable reason for that is the addition of the fourth layer in the network.

Figure 3 presents the LSTM 2 general accuracy as a function of the SNR in one of the tests. The general SNR that gives a 90% accuracy is -2.50 dB. Figure 4 presents the SNR per modulation obtained for the same architecture in the same test. It can be seen that the minimum SNR required for the network to maintain 90% accuracy is 0 dB, and the modulation that limits this condition is the phase modulation with the P1 code.

TABLE I

Average performance test results of the LSTM architectures

Model	Accuracy (%)	Epochs	Latency/ Sample (ms)	SNR Modulations (dB)	SNR General (dB)
LSTM 1	89.67	500	29.23	1.33	-2.58
LSTM 2	89.33	500	21.33	0.83	-2.66

Figure 3 presents the LSTM 2 general accuracy as a function of the SNR in one of the tests done. The general SNR that gives a 90% accuracy is -2.50 dB. Figure 4 presents the SNR per modulation obtained for the same architecture in the same test. It can be seen that the minimum SNR at which the network can maintain 90% accuracy is 0 dB, and the modulation that limits this condition is the phase modulation with P1 code.

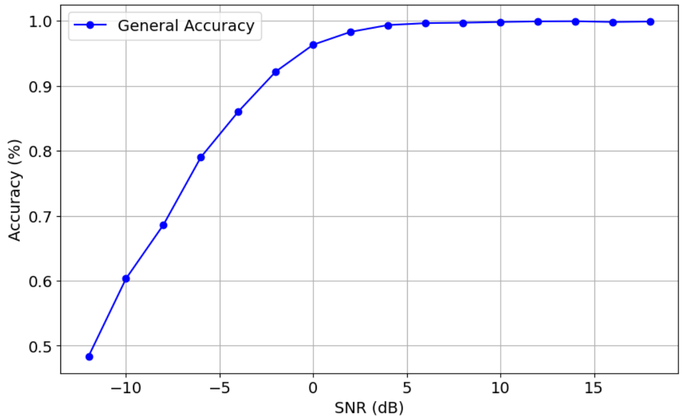

Fig. 3. General accuracy as a function of SNR of the proposed LSTM architecture.

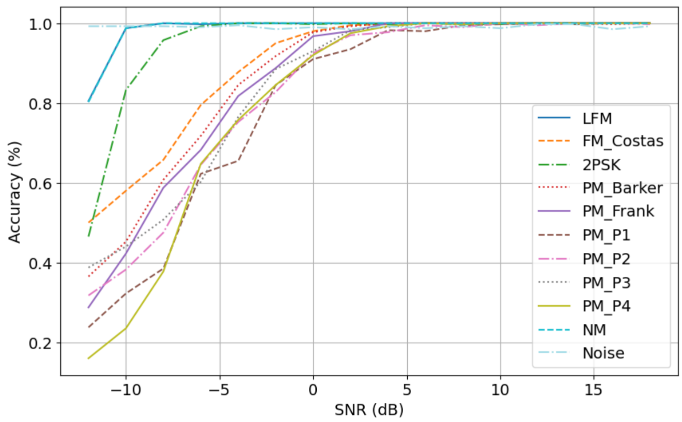

Fig. 4. Modulation accuracy as a function of SNR of the proposed LSTM architecture.

From Figure 5, it is possible to identify that the LSTM has more difficulty in classifying phase modulations, especially with Frank and P codes. In contrast, the LSTM excelled in the classification task for chirp, 2PSK, noise, and non-modulated signals, achieving 90% accuracy under a -9 dB condition.

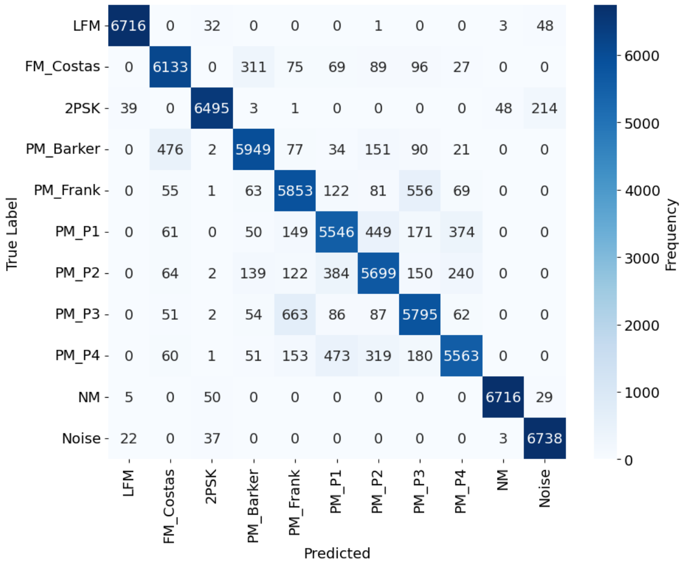

Fig. 5. Confusion matrix showing the misclassification errors of the proposed LSTM architecture.

B. CNN

The average results obtained for each performance parameter for both CNN models tested are presented in Table II. Analyzing it, it can be seen that the proposed network (CNN 2) was able to overcome 1,8% the baseline network in terms of accuracy. Consequently, the minimum SNR (general and per modulation) parameter is also reduced.

Comparing both SNR parameters, a 19% reduction (1.6 dB) was observed when examining the modulations separately, and a 35% reduction (0.70 dB) was noted when considering all modulations together. Nevertheless, when the number of epochs and latency per sample is observed, it is noticed that CNN 2 is heavier than CNN 1 once the number of epochs needed to train the network is almost twice that of CNN 1, and the latency parameter is 2.88 times higher than that of CNN 1.

TABLE II

AVERAGE PERFORMANCE TEST RESULTS OF THE CNN ARCHITECTURES

Model	Accuracy (%)	Epochs	Latency/ Sample (ms)	SNR Modulations (dB)	SNR General (dB)
CNN 1	82.00	27.80	0.39	8.60	2.30
CNN 2	83.80	56.20	1.04	7.00	1.50

The statistical test results were 0.0013, 0.0602, and 0.0139 for accuracy, SNR per modulation, and general SNR, respectively. As mentioned in Section II, the p-value approval criterion for this work is 0.05. Therefore, looking rigorously at the p-values presented, it is possible to verify that CNN 2 presented an improved performance compared to CNN 1 when observing the accuracy and the general SNR.

The CNN 2 result for general accuracy as a function of the SNR is shown in Figure 6. As we can see, the SNR that gives a 90% accuracy is 1 dB. The accuracy per modulation as a function of the SNR for the same test is presented in Figure 7. As a result, the minimum SNR at which the network can maintain 90% accuracy is approximately 7 dB. The modulation that limits this condition is the phase modulation with the P4 code. Figure 8 presents a confusion matrix obtained for this CNN architecture. The CNN also has more difficulty classifying phase modulations, especially with Frank and P codes. Similarly to the LSTM, the CNN also excelled in the classification task for chirp, noise, and non-modulated signals, classifying them with 90% accuracy under a -7.5 dB condition.

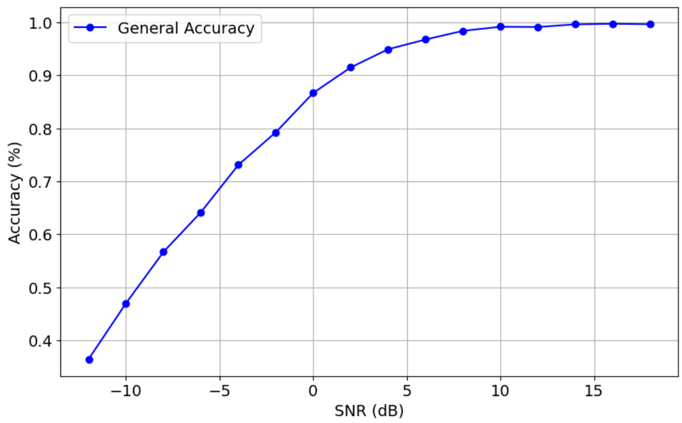

Fig. 6. General accuracy as a function of SNR of the proposed CNN architecture.

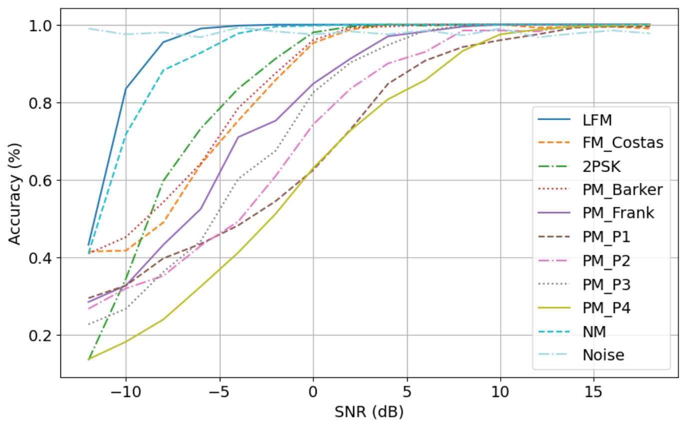

Fig. 7. Modulation accuracy as a function of SNR of the proposed CNN architecture.

C. Models Comparison

From the previous results, both models have demonstrated difficulty classifying phase modulations, especially in distinguishing between similar modulations, such as those with P code. Similarly to the LSTM, the CNN can easily classify noise, chirps, and non-modulated signals. LSTM was also

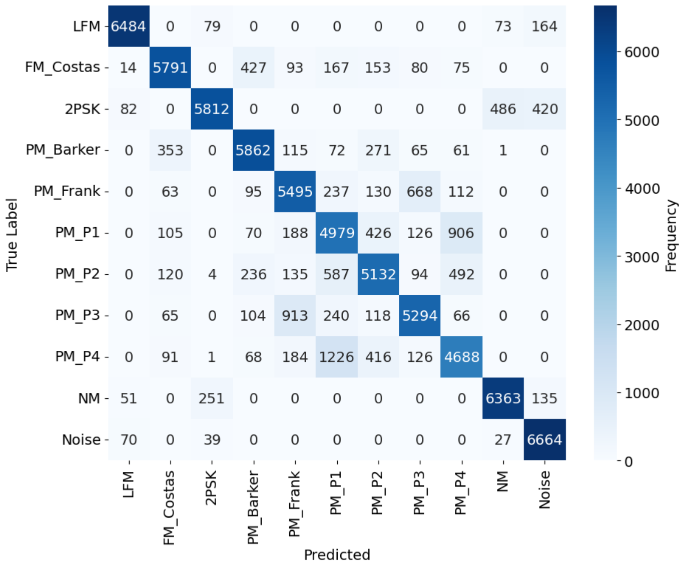

Fig. 8. Confusion matrix showing the misclassification errors of the proposed CNN architecture.

able to classify 2PSK modulations effortlessly. Comparing the results obtained from the two models tested, it is evident that the LSTM outperforms the CNN in terms of accuracy, making it capable of classifying modulations under worse SNR conditions. Nonetheless, this improvement incurs a high computational cost to train the model and a high latency time after training, compared to the CNN model. The choice of using an LSTM or a CNN will depend on the context of the application. On the one hand, if a more precise classification is needed and processing time is not a critical problem, the LSTM can be an appropriate model. On the other hand, if processing time is crucial and the application permits slightly neglecting accuracy, the CNN might be a good approach.

IV. Conclusions

This paper proposes an LSTM and a CNN architecture for the Automatic Modulation Classification task applied to radar signals. The performance of the proposed architectures was compared to that of the baseline architectures. The LSTM architecture can reduce latency time by 27%, while maintaining similar accuracy and SNR performance to the baseline architecture. Regarding CNN, the proposed architecture could improve accuracy and SNR performance, but at the cost of higher latency time.

LSTM has proven to be a powerful approach to this problem, although it demands high computational power for training and presents high latency time in the test stage. Convolutional Neural Networks have been demonstrated to be an efficient method for the AMC task. Besides having an accuracy slightly lower than that of the LSTM architectures tested, the CNN architectures had a latency time during the testing stage that was at least 95% lower than that of the faster LSTM architecture. The choice of using an LSTM or a CNN model must be made according to the application's needs. If it requires higher accuracy and processing time is not a concern, LSTM might be an excellent model. Alternatively, if processing time is a crucial demand, the CNN can be a good option, albeit at the expense of slightly compromising accuracy capability.

Acknowledgment

The authors thank the Electronic Warfare Laboratory of the Aeronautics Institute of Technology and the LOM Project for providing the facilities to carry out this work. This work was partially sponsored by SIATT, FINEP (under grant no. 01.22.0581.00), National Institute of Science and Technology (INCT-Signals) sponsored by Brazil's National Council for Scientific and Technological Development (CNPq) under grant no. 406517/2022-3, Ministry of Science, Technology and Innovation (MCTI), and by the São Paulo Research Foundation (FAPESP) under grant no. 20/09838-0 (BI0S—Brazilian Institute of Data Science).

References

[1] P. Sharma, K. K. Sarma, and N. E. Mastorakis, "Artificial intelligence aided electronic warfare systems- recent trends and evolving applications," *IEEE Access*, vol. 8, pp. 224 761–224 780, 2020.

[2] T. Singh and A. Gulhane, "8 key military applications for artificial intelligence in 2018," https://blog.marketresearch.com/8-key-military-applications-for-artificial-intelligence-in-2018, 2018, accessed: Apr. 2, 2025.

[3] G. Song, G. Jeon, and D. Yoon, "Deep learning-based automatic modulation classification for composite modulated radar signal using time-frequency image," in *2024 34th International Telecommunication Networks and Applications Conference, ITNAC 2024*. Institute of Electrical and Electronics Engineers Inc., 2024.

[4] S. Wei, Q. Qu, X. Zeng, J. Liang, J. Shi, and X. Zhang, "Self-attention bi-lstm networks for radar signal modulation recognition," *IEEE Transactions on Microwave Theory and Techniques*, vol. 69, pp. 5160–5172, 11 2021.

[5] B. E. Altuntas, O. Aksu, M. Y. Celik, and M. H. Durak, "Deep learning based automatic modulation recognition using gelu activation function," in *4th International Conference on Emerging Smart Technologies and Applications, eSmarTA 2024*. Institute of Electrical and Electronics Engineers Inc., 2024.

[6] G. Ruan and W. Wu, "Automatic recognition of radar signal types based on convolutional neural network," in *2019 International Conference on Image and Video Processing, and Artificial Intelligence*, R. Su, Ed. SPIE, 11 2019, p. 124.

[7] D. Li, R. Yang, X. Li, and S. Zhu, "Radar signal modulation recognition based on deep joint learning," *IEEE Access*, vol. 8, pp. 48 515–48 528, 2020.

[8] Y. Wen, X. Wang, G. Xu, and F. Wang, "Lpi radar signals intra-pulse modulation recognition based on ca-shufflenet," in *Proceedings - 2022 14th International Conference on Signal Processing Systems, ICSPS 2022*. Institute of Electrical and Electronics Engineers Inc., 2022, pp. 144–149.

[9] Z. M. Liu and P. S. Yu, "Classification, denoising, and deinterleaving of pulse streams with recurrent neural networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, pp. 1624–1639, 8 2019.

[10] X. Li, Z. Liu, Z. Huang, and W. Liu, "Radar emitter classification with attention-based multi-rnns," *IEEE Communications Letters*, vol. 24, pp. 2000–2004, 9 2020.

[11] V. Clerico, J. González-López, G. Agam, and J. Grajal, "Lstm framework for classification of radar and communications signals," in *2023 IEEE Radar Conference (RadarConf23)*. IEEE, 5 2023, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10149618/

[12] Kilian, "Deepradar: A radar signal dataset for deep learning," https://www.kaggle.com/datasets/khilian/deepradar, 2022, accessed: Apr. 2, 2025.

[13] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over the air deep learning based radio signal classification," 12 2017. [Online]. Available: http://arxiv.org/abs/1712.04578 http://dx.doi.org/10.1109/JSTSP.2018.2797022