Automatic Modulation Classification for Low-Power IoT Applications

Yasmín R. Mondino-Llermanos, and Graciela Corral-Briones

***Abstract*—The Internet of Things (IoT) has swiftly become one of the most important technologies in recent years. Radio spectrum access represents a stern challenge for the IoT as a consequence of the increased use of connected devices. This is particularly true for IoT devices operating in the unlicensed band where the huge demand for wireless connectivity will require techniques that use the spectrum efficiently. Avoiding training sequences enables a more efficient spectrum usage and has the additional advantage of reducing the power consumption of IoT devices, but it requires modulation identification mechanisms. This paper presents a simple yet efficient method to classify received signals according to their modulation type. We propose the application of a single hidden layer neural network with a small number of trainable parameters for performing the classification between seven different modulation types. The designed classifier achieves a maximum accuracy of 95% when the signal-to-noise ratio (SNR) of the input data is 12 dB, and in the presence of multi-path fading, sample rate offset and carrier frequency offset.**

Link to graphical and video abstracts, and to code: https://latamt.ieeer9.org/index.php/transactions/article/view/8267

***Index Terms*—Internet of things, Radio spectrum access, Automatic modulation classification, Feature extraction, Mutual information, Artificial neural network**

I. Introduction

The reduction of the communication overhead is known to improve the spectrum efficiency and transmit power consumption. Both of these characteristics are of paramount importance for enabling wireless internet of things (IoT) networking on a massive scale. For this purpose, modulation identification techniques that do not require training or pilot sequences for data retrieving are of particular interest [1].

Until this day, there has been a wide variety of deep neural networks and convolutional neural networks developed that try to accomplish this specific task. Lately, there have also been contributions focused in reducing the complexity of the networks to make them easier to deploy on edge devices [2]. This paper aims to propose a simple neural network of only one hidden layer, that can provide a classification system for signals highly corrupted by the channel. The designed model is intended to have a low complexity with a small number of trainable parameters, making it suitable for applications on low-power devices with limited computational resources. Moreover, the reduced amount of parameters would make it suitable for situations when there is a reduced amount of available training data.

Yasmín R. Mondino-Llermanos and Graciela Corral-Briones are with National University of Cordoba, Cordoba, Argentina (e-mails: yasmin.mondino@mi.unc.edu.ar and graciela.corral@unc.edu.ar).

In this paper, Section II gives an overview of previous papers that make use of neural networks and feature extraction to generate classification algorithms. Section III describes the dataset used throughout this work. Section IV presents all the extracted features and their formulae, while Section V presents the ultimately selected ones. The characteristics of the neural network used for the classification are shown in Section VI, and Section VII provides the experimental results of it. Finally, Section VIII concludes this paper and gives an overall evaluation.

II. Related Work

Automatic modulation classification (AMC) techniques can be classified into traditional techniques which include decision-theoretic approaches and feature-based approaches, along with advanced deep learning based approaches [3].

In [4], the authors carry out a survey of deep learning architectures used for modulation classification in wireless communications, such as deep neural networks (DNNs), recurrent neural networks (RNNs), long short-term memory networks (LSTMs) and convolutional neural networks (CNNs). Their study describes the fundamental concepts of each of these architectures and provides a list of implementations found in the literature. It highlights not only the used architecture, but also the modulation types, the channel and the best results obtained by each of these classifiers. Most of these algorithms exhibit a good performance, achieving an accuracy of over 90% at around 10 dB; however, they consume a lot of computational resources. Consequently their implementation in devices with limited computing capability, low memory and small storage space, becomes an extremely complicated task.

In recent years, new works have aimed to generate small architectures that can be suitable for IoT devices that do not have the capability to process large numbers of parameters. In [5] the authors develop a novel deep learning classifier for IoT-based systems that combines the feature extraction capabilities from two-dimentional information of CNNs and the capability to extract sequential correlations within time series data of the LSTMs and the gated recurrent unit (GRU) based RNN. The model outperformes other preexisting models [6], [7] but without demanding a larger training and computational requirement. In [8] the authors propose the use of depthwise and separable depthwise convolutional neural network architectures that can be implemented in devices that are constrained by power and area. The developed architectures achieve similar performances to the conventional

CNN approaches [9] but with a significant reduction of parameters. In [10], the authors propose an AMC method that combines residual learning and squeeze-excitation blocks that achieves an even higher reduction of parameters while improving the performance in high SNR scenarios.

Even though researchers are continuously presenting new alternatives for developing lower complexity deep learning based classifiers, the current architectures still require tens or hundreds of thousands of parameters [10]. For this reason, we decide to implement a featured-based classification method because these classifiers are known for offering good performances with a reduced computational complexity. Furthermore, they can be improved by applying machine learning algorithms, as their application can provide a simpler way to implement the decision making process and help in reducing the dimensions of the features set [3].

Artificial neural networks (ANNs) are useful for classification problems because they have a flexible structure that make them easy to implement. In addition, ANNs can adapt and learn to work with complicated signals [11]. They have been widely used for implementing AMC systems; for instance, the authors of [12] and [13] developed different classifiers that make use of an ANN with seven input features to perform the classification between eight and five modulation schemes, respectively. The employed features are derived from the instantaneous amplitude, frequency, and phase of the signals. In [14] it is presented another AMC method that implements an ANN and makes use of a different set of statistical features to classify between seven digital modulations. The input features of this model do not only include information about the instantaneous amplitude, phase and frequency of the received signal but also, higher order statistics.

Most of the existent studies are limited to simulations and not evaluated on actual hardware, this situation generated that some authors started to design their own ANN based classifiers and implement them on a software defined radio (SDR) testbed, like the ones presented on [15] and [16]. Each of these models made use of a different set of input features, but both of them exhibited a good performance in the classification of two different groups of seven digital modulation schemes in real-time. Despite achieving good results with their architectures, the papers only present the theoretical capabilities of each of their selected features to distinguish different modulation schemes. However, they do not present the selection criteria that was applied in order to choose the features over other ones. In our work, we take into consideration a set of 32 features and then use the mutual information algorithm to perform the selection of the most relevant ones. Another important aspect to consider is that even though the authors do not make use of a synthetically generated dataset, the configuration of their experimental setups do not generate a dataset of signals with all the characteristics that are considered in the dataset used within this work.

The use of the feature selection method is based on the work by Lee et al. [17], which presents an improved version of previously developed ANNs for AMC. The improvement is mainly focused on the implementation of the mutual information algorithm to select the best subset of input features and reduce the complexity of the model. However, while in [17] the authors utilise it in an scenario of signals affected only by additive white Gaussian noise and Doppler shift, we present an implementation with signals that are more in line with the reality of wireless signals.

III. Dataset

All the experiments within this study are made using the RadioML 2016.10A dataset [18]. This dataset consists of synthetically generated signals using GNU Radio. The signals correspond to 11 different modulation types. Consisting of eight digital; BPSK, QPSK, 8PSK, 16QAM, 64QAM, GFSK, CPFSK, PAM4 and, three analogue modulations; WB-FM, AM-SSB, AM-DSB. Nevertheless, only seven of the digital modulation types are used in the classifier presented within this paper, as the classification is performed between BPSK, QPSK, 8PSK, 16QAM, GFSK, CPFSK and PAM4. This subset of modulation types was selected as it was deemed the most suitable ones for IoT devices, of those present in the dataset. High order modulations, such as 64QAM, are more prone to error than lower-order modulations. Due to their sensitivity to noise, they require to be transmitted over short distances or with a significantly high power level, which is not feasible for remote IoT devices [19]. WB-FM consumes a large bandwidth so its use is recommended for applications in which spectral efficiency is not important [20], whereas IoT devices require an efficient use of the available spectrum. AM-DSB also requires a large bandwidth and it does not present a high bandwidth efficiency. Whilst, the spectral efficiency is higher with the use of AM-SSB, the hardware implementation of this modulation scheme is complex and it demands expensive equipment [21].

Each of the signal data examples included in the dataset has a length of 128 samples and a SNR value from -20 dB to 20 dB. For each combination of modulation type and SNR value, the dataset contains 1000 signal data examples. It is important to note that all of the signals have been affected by additive white Gaussian noise, selective multi-path fading, sample rate offset and carrier frequency offset. These channel specifications make it a more realistic dataset of wireless signals, compared to others that have been widely used in automatic modulation classification methods and that only include additive white Gaussian noise.

IV. Feature Extraction

The correct feature selection is essential in this work, as the main objective is to design a neural network with a small number of inputs, that has a reduced computational complexity which, produces a reliable modulation classification. The selection is performed between 32 different features that have been used in previous works and that are enumerated below.

A. Time Domain Features

1) Standard deviation of the absolute value of the normalised centered instantaneous amplitude

$$\sigma_{aa} = \sqrt{\frac{1}{N}\left[\sum_{i=1}^{N} a_{cn}^2(i)\right] - \left[\frac{1}{N}\sum_{i=1}^{N}|a_{cn}(i)|\right]^2} \quad (1)$$

where a_{cn} can be formulated as

$$a_{cn}(i) = \frac{a(i)}{m_a} - 1 \quad (2)$$

where m_a is the average value of the instantaneous amplitude over the length of the signal N.

2) Standard deviation of the normalised signal amplitude

$$\sigma_{v} = \sqrt{\frac{1}{N}\left[\sum_{i=1}^{N} a_{v}^2(i)\right] - \left[\frac{1}{N}\sum_{i=1}^{N}|a_{v}(i)|\right]^2} \quad (3)$$

where a_v can be formulated as

$$a_v(i) = \sqrt{\frac{a(i)}{m_a}} - 1 \quad (4)$$

3) Standard deviation of the instantaneous phase

$$\sigma_{dp} = \sqrt{\frac{1}{N}\left[\sum_{i=1}^{N} \phi^2(i)\right] + \left[\frac{1}{N}\sum_{i=0}^{N}\phi(i)\right]^2} \quad (5)$$

where $\phi(i) = angle(z(i))$, being $z(i)$ the complex signal.

4) Standard deviation of the absolute instantaneous phase

$$\sigma_{ap} = \sqrt{\frac{1}{N}\left[\sum_{i=1}^{N} \phi^2(i)\right] + \left[\frac{1}{N}\sum_{i=1}^{N}|\phi(i)|\right]^2} \quad (6)$$

5) Standard deviation of the absolute value of the normalised centered instantaneous frequency

$$\sigma_{af} = \sqrt{\frac{1}{N}\left[\sum_{i=1}^{N} f_N^2(i)\right] + \left[\frac{1}{N}\sum_{i=1}^{N}|f_N(i)|\right]^2} \quad (7)$$

where f_N can be formulated as

$$f_N(i) = \frac{f(i)}{m_f} - 1 \quad (8)$$

where m_f is the average value of the instantaneous frequency $f(i)$ over the length of the signal N.

6) Standard deviation of the change in the instantaneous phase

$$\sigma_{\Delta\phi} = \sqrt{\frac{1}{N}\left[\sum_{i=1}^{N} \Delta\phi^2(i)\right] - \left[\frac{1}{N}\sum_{i=1}^{N}\Delta\phi(i)\right]^2} \quad (9)$$

7) Kurtosis of the normalised centered instantaneous amplitude

$$\mu_{42}^{a} = \frac{E\{a_{cn}^4(i)\}}{E\{a_{cn}^2(i)\}} \quad (10)$$

8) Kurtosis of the normalised centered instantaneous frequency

$$\mu_{42}^{f} = \frac{E\{f_{N}^4(i)\}}{E\{f_{N}^2(i)\}} \quad (11)$$

9) Ratio of the in-phase component and quadrature component of the signal power

$$\beta = \frac{\sum_{i=1}^{N} a_Q^2(i)}{\sum_{i=1}^{N} a_I^2(i)} \quad (12)$$

where a_I and a_Q are the in-phase and quadrature samples of the complex signal.

10) Skewness of amplitude

$$S_a = \left|\frac{\frac{1}{N}\sum_{i=1}^{N}(a(i)-m_a)^3}{\left(\frac{1}{N}\sum_{i=1}^{N}(a(i)-m_a)^2\right)^{\frac{3}{2}}}\right| \quad (13)$$

11) Peak to average ratio

$$PA = \frac{max|a|}{\frac{1}{N}\sum_{i=1}^{N} a(i)} \quad (14)$$

12) Peak to RMS ratio

$$PR = \frac{max|a|^2}{\frac{1}{N}\sum_{i=1}^{N}(a(i))^2} \quad (15)$$

B. Transformation Based Features

1) Maximum value of the power spectral density of the normalised centered instantaneous amplitude

$$\gamma_{max} = \frac{\max|DFT(a_{cn}(i))|^2}{N} \quad (16)$$

where $a_{cn}(i)$ is the normalised centered instantaneous amplitude defined by Equation 2.

2) Maximum value of the power spectral density of the square of the instantaneous amplitude

$$\Gamma_{max} = \frac{\max\left|DFT(a^2(i))\right|^2}{N} \quad (17)$$

3) Skewness of frequency

$$S_f = \frac{E\left[(R_i-\mu_i)^3\right]}{(\sigma_i)^3} \quad (18)$$

where σ^2 can be formulated as

$$\sigma^2 = E\left[(R_i-\mu_i)(R_i-\mu_i)^*\right] \quad (19)$$

where $\mu_i = E(R_i)$ and R_i is the Fourier transform of the signal.

C. Statistical Signal Characterisation Features

There are four Statistical Signal Characterisation (SSC) parameters [22] that can be obtained if each signal is considered as a set of consecutive segments. A segment is bound by consecutive maximum and minimum values of the waveform, and its amplitude is defined as $A_n = |a_n - a_{n-1}|$, where A_n is the amplitude of the nth segment, with a_n and a_{n-1} being the amplitudes at the ending and beginning of the segment. Additionally, the segment period is defined as $T_n = |t_n - t_{n-1}|$ where T_n is the period of the nth segment, with t_n and t_{n-1} being the elapsed time at the ending and beginning of the segment.

The SSC parameters are defined from the two previously mentioned segment characteristics as follows:

$$M_a = \sum_{i=1}^{N_s} \frac{A_i}{N_s} \qquad M_t = \sum_{i=1}^{N_s} \frac{T_i}{N_s} \tag{20}$$

$$D_a = \sum_{i=1}^{N_s} \frac{|A_i - M_a|}{N_s} \qquad D_t = \sum_{i=1}^{N_s} \frac{|T_i - M_t|}{N_s} \tag{21}$$

where A_i is the amplitude of the ith segment, T_i the period of the ith segment, N_s the number of segments in the signal, M_a the amplitude mean, M_t the period mean, D_a the standard deviation of the amplitude mean, D_t the standard deviation of the period mean.

To obtain good results, it is crucial to accurately detect the maximum and minimum values of the signals. This improves the detection of the signals segments, contributing to the correct calculation of these parameters. Various methods have been developed for this purpose, and in this paper, we opted to filter the signals prior to the calculation of these four parameters. The chosen filter is the Savitzky-Golay Smoothing and Differentiation Filter (“savgol filter”) [23]. This filter is often used to reduce high frequency noise in a signal due to its smoothing properties and to reduce low frequency signals using differentiation.

The chosen Savitzky-Golay filter calculates a polynomial fit of order 2 in a filter window of size 7 while it moves along the signal. The output of the filter at the center of each window is then given by the polynomial fit at the center point.

D. Higher Order Statistical Features

These features are derived from a combination of two or more High Order Moments (HOMs)

$$M_{pq} = E\left[z^{p-q}(z^*)^q\right] \tag{22}$$

being z the complex signal.

1) High Order Cumulants (HOCs)
   The formula of each of the considered HOC is shown in Table I. However, these HOCs are then rescaled as described in [24], with each cumulant raised to the power $\frac{2}{p}$, where p is the order of the cumulant.
2) Quotient between high order moments

$$v_{20} = \frac{M_{42}}{M_{21}^2} \tag{23}$$

$$v_{30} = \frac{M_{63}}{M_{21}^3} \tag{24}$$

V. Feature Selection

The selection of the features that are used to discriminate between the seven modulation schemes is performed by the application of the Maximum Relevance Minimum Redundancy (mRMR) criterion of the mutual information algorithm [25]. As it was mentioned in Section II, this criterion was previously used by [17] and proved to achieve good results.

The mutual information is a metric that can be used to evaluate the relevance of a feature to the modulation class. It is usually expressed as

$$I(x_i, c) = \int\int P(x_i, c) \log \frac{P(x_i, c)}{P(x_i)P(c)} dx_i dc \tag{25}$$

where x_i is the ith feature and c is the class label.

The mRMR criterion is a sub-optimal way of selecting the set of features that is less computationally demanding, as it performs an incremental feature selection. It tends to select

TABLE I
HIGH ORDER CUMULANTS AS COMBINATIONS OF HIGH ORDER MOMENTS

HOCs	**HOMs Expression**
C_{20}	M_{20}
C_{21}	M_{21}
C_{40}	$M_{40} - 3M_{20}^2$
C_{41}	$M_{40} - 3M_{20}M_{21}$
C_{42}	$M_{42} - \|M_{20}\|^2 - 2M_{21}^2$
C_{60}	$M_{60} - 15M_{20}M_{40} - 30M_{20}^3$
C_{61}	$M_{61} - 5M_{21}M_{40} - 10M_{20}M_{41} + 30M_{20}^2M_{21}$
C_{62}	$M_{62} - 6M_{20}M_{42} - 8M_{21}M_{41} - M_{22}M_{40} + 6M_{20}^2M_{22} + 24M_{21}^2M_{20}$
C_{63}	$M_{63} - 9M_{21}M_{42} + 12M_{21}^3 - 3M_{20}M_{43} - 3M_{22}M_{41} + 18M_{20}M_{21}M_{22}$
C_{80}	$M_{80} - 35M_{40}^2 - 28M_{60}M_{20} + 420M_{20}^2M_{40} - 630M_{20}^4$
C_{81}	$M_{81} - 21M_{20}M_{61} - 7M_{21}M_{60} - 35M_{40}M_{41} + 210M_{20}^2M_{41} + 210M_{20}M_{21}M_{40} - 630M_{20}^3M_{21}$

TABLE II
MOST RELEVANT FEATURES

Order of relevance	Feature symbol	Feature name
1	C_{20}	Cumulant of order 2
2	S_f	Skewness of frequency
3	σ_v	Standard deviation of the normalised signal amplitude
4	D_a	Standard deviation of the signal segments amplitude mean
5	M_a	Amplitude mean of the signal segments
6	S_a	Skewness of amplitude
7	$\sigma_{\Delta\phi}$	Standard deviation of the change in the instantaneous phase
8	C_{60}	Cumulant of order 6
9	M_t	Period mean of the signal segments

features with a high correlation with the class and a low correlation between themselves. It is usually expressed as

$$V(j) = I(x_j, c) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} I(x_j, x_i) \tag{26}$$

where S_{m-1} represents the set of selected features at the $(m-1)$th iteration.

In [17], it was shown that both criteria ranked the features in a different order of relevance depending on the scenario in which the signals were being transmitted. Since the signals that we use are affected by other channel effects, we perform a new ranking of features. Our dataset consists of 1000 signal data examples for each combination of modulation type and SNR value. However, not all of these examples will be used in the training and validation of the neural network, hence we can not make use of all the dataset for the feature selection either. Instead of that, we perform an aleatory selection of 40% of the signals and calculate the feature importance according to them [1]. This operation is then repeated 100 times with different aleatory selected signals of the dataset. Table II shows the 9 features most frequently ranked with the highest relevance.

VI. NEURAL NETWORK STRUCTURE

The neural network used throughout this work initially consists of four layers. The first layer, or input layer, has the same amount of nodes as input features. These input features correspond to the nine highest ranked parameters according to the mRMR criterion. Before the second layer, we perform a batch normalisation to standardise the input values. The second and third layers, or hidden layers, have 25 and 12 neurons each, with rectified linear activation function (ReLU). Although these numbers of hidden layers and neurons were originally selected to match the ones used in [16], both of them are reduced in the following tests. Finally, the output layer has the same number of nodes as modulation types for which the classification is performed. This is because each neuron applies the softmax activation function to generate a one-hot encoding output layer. For this classifier there are seven neurons in the output layer.

[1]All the calculations were performed making use of the package **mrmr_selection**, located at https://github.com/smazzanti/mrmr, written by Samuele Mazzanti

VII. RESULTS

All the neural network architectures within this work are trained, validated and tested with signals that have a SNR value between the range of -10 and 20 dB, from the dataset mentioned in Section III. However, before performing any tests, the dataset is aleatory divided into two groups; 40% of the data is selected for training, the remaining 60% is left for testing. From the training dataset, 80% is actually used for training, while the other 20% is used for validation. The description of each dataset is summarised in Table III.

TABLE III
DETAILS OF THE USED DATASETS

TRAINING DATASET	
Number of considered SNR values	15
Number of modulation schemes	7
Number of examples for each combination of SNR and modulation	320
Total number of examples	**33,600**
VALIDATION DATASET	
Number of considered SNR values	15
Number of modulation schemes	7
Number of examples for each combination of SNR and modulation	80
Total number of examples	**8,400**
TESTING DATASET	
Number of considered SNR values	15
Number of modulation schemes	7
Number of examples for each combination of SNR and modulation	600
Total number of examples	**63,000**

The training process of a neural network depends on the training examples that are used. This is the reason why every test is performed multiple times, each with different training and validation datasets, randomly separated. The results shown in every test are then the average results obtained from 50 differently trained neural networks, all tested with the same dataset.

All training processes make use of the Categorical Cross Entropy Loss function and the parameters presented in Table IV. Even though it is indicated that the maximum number of training epochs is 1000, all the training processes are stopped once the model performance stops improving on the validation dataset.

TABLE IV
TRAINING PARAMETERS OF THE NEURAL NETWORK

Optimizer					Batch	Max
Name	Learning rate	β_1	β_2	ϵ	size	epochs
Adam	0.005	0.9	0.999	$1e-07$	50	1000

A. Neural Network with Two Hidden Layers

The neural network is first designed with two hidden layers as mentioned in Section VI. The performance analysis of the classifier is done by the calculation of the accuracy of the model, defined as

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{27}$$

Only for this test, we decide to perform two times the training of the neural network making use of two different datasets. First, we use the previously described one and then, we include into the training dataset signals from the RadioML 2016.10A dataset that have a SNR value between -20 and -10 dB. Both of the trained models are then tested with an extended dataset that also includes signals between -20 and 20 dB. In Fig. 1 it is possible to see the average accuracy of the models for the test set of signal examples. This graphic evidences that the ANN is not able to learn from signals with a SNR lower than -10 dB, as the performance of both models is the same. It is for this reason that for the rest of this work, we only make use of signals with a SNR value in the range of -10 and 20 dB for both, training and testing. In Fig. 1 it is also possible to see that the accuracy of the model for signals of 0 dB is 72.87%, whereas for signals of 4 dB or more, it is higher than 90%.

Other important metrics to evaluate the performance of the model are the precision and recall of each class. The former finds out what fraction of the predicted signals as a certain modulation type actually belongs to that class. While the latter measures the ability of the model to predict all the members of a certain class. These metrics are defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP + FN}} \tag{28}$$

where we consider TP to be the number of correct predictions of a signal as part of one specific modulation type, FP is the number of incorrect predictions of a signal as a member of a class, and FN is the number of incorrect predictions of a signal as not part of a class.

The two previously mentioned metrics are captured in the F1 Score. This metric works as an indicator of the ability of the model to capture positive cases and be accurate with the cases it does capture. The F1 Score is formally defined as the harmonic mean of the precision and recall. The variation of the F1 score with the SNR value for each of the modulation types is shown in Fig. 2. It is possible to observe that for signals of 4 dB or more the F1 score is always higher than 0.8, despite the modulation.

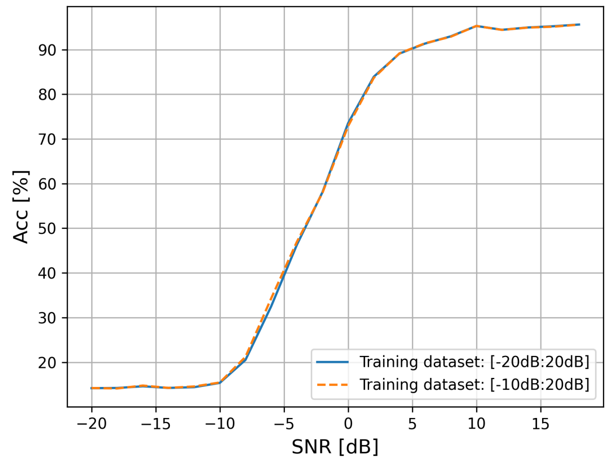

Fig. 1. Variation of the accuracy of the classifier with the SNR value, after being trained and validated with different datasets. The classifier consists of a four layer neural network, with two hidden layers of 25 and 12 neurons each.

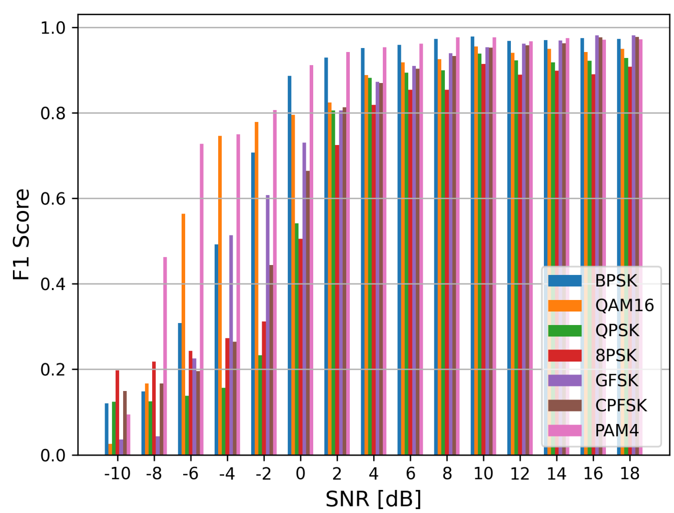

Fig. 2. Variation of the F1 Score with the SNR value for each modulation type. The classifier consists of a four layer neural network, with two hidden layers of 25 and 12 neurons each.

For the case of 18 dB, we show a confusion matrix in Fig. 3, in which the main errors are that of 8PSK missclassified mainly as QPSK. This is a common error present in previous works [5], [8], [9] and it can be explained due to the similarity between the constellation points between these modulations.

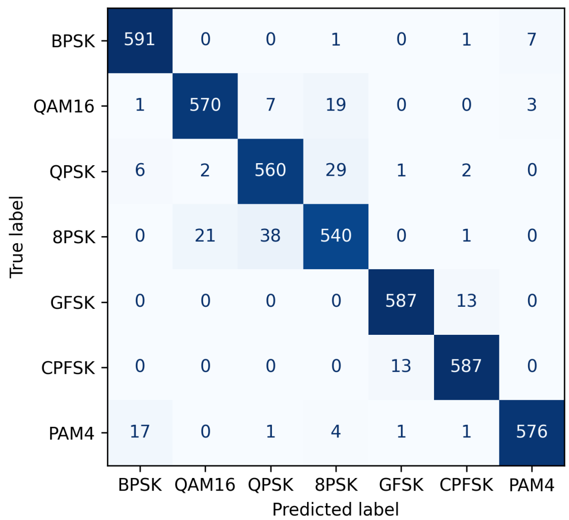

Fig. 3. Confusion matrix at SNR = 18 dB, generated with a classifier of two hidden layers of 25 and 12 neurons each.

B. Neural Network with One Hidden Layer

In this next test, we modify the number of hidden layers and neurons of the neural network. Table V contains all the tested configurations of the neural network with their corresponding average accuracy and F1 score across all SNR values on the test dataset. It also presents the number of coefficients that each model has associated according to its number of layers and neurons. This value is important as it is a basic indicator of the complexity of each model, i.e., a larger amount of coefficients implies a bigger memory requirement, as well as a higher number of multiplication operations during its use.

It is possible to see that the accuracy and F1 score results do not vary considerably with the elimination of the second hidden layer. This result is consistent with previously published analysis, such as [26], which establish that for nearly all problems the use of two hidden layers rarely improves the model. It is also possible to observe that there is not a significant variation between the performance of models with one hidden layer of 15, 20 and 25 neurons. However, we can see that choosing a model with one hidden layer and with a reduced number of neurons has associated a smaller number of parameters. For these reasons we decide to keep a neural network of only one hidden layer of 15 neurons for all of the remaining tests.

TABLE V
TEST RESULTS OF NEURAL NETWORKS WITH DIFFERENT ARCHITECTURE

1st hidden layer	2nd hidden layer	Number of coefficients	Average accuracy	Average F1 Score
25	12	689	72.08%	0.713
25	–	468	71.83%	0.711
20	–	383	71.60%	0.709
15	–	298	71.25%	0.704
10	–	213	70.31%	0.694

C. Variation of the Training Dataset

This next test is executed to determine what is the best performance that could be achieved with a limited training dataset. Table VI shows the variation of the average accuracy and F1 score obtained across all SNR values on the test dataset for the same model after it was trained and validated with a different dataset. The datasets that are described on the table refer to the combination of both training and validation datasets, while the test dataset remains always the same.

It is possible to see that the best performance is achieved with the originally selected dataset. As expected, the performance of the neural network deteriorates as the size of the training dataset is reduced. However, the reduction to half of the number of examples for each combination of modulation type and SNR value can be performed without producing a significant degradation in the overall performance of the classifier.

TABLE VI
TEST RESULTS OF NEURAL NETWORKS WITH DIFFERENT TRAINING

Dataset SNRs	Examples for each SNR-modulation	Dataset size	Average accuracy	Average F1 Score
$[0:20:2]$	400	42000	71.25%	0.704
$[0:20:2]$	200	21000	71.46%	0.710
$[0:20:2]$	100	10500	69.58%	0.689
$[0:20:2]$	50	5250	67.06%	0.664
$[0:20:2]$	10	1050	61.87%	0.611

D. Comparison with Other Classification Models

In this test we compare the performance of our classifier with others that were also designed with feature extraction and the use of an ANN. All of these classifiers were mentioned in Section II and they all have a different set of input features, hidden layers and neurons. It is important to notice that these classifiers were not originally designed for the same group of modulation schemes, but the comparison with our model can help in showing that we made a correct selection of input features. This is because we trained, validated and tested all the models in the same way. In Fig. 4 it is shown that all the models behave in a similar way for low SNR values, while our proposed model has a noticeable better performance on high SNR values.

E. Classification in the Presence of Interference

In the last test performed on the model, we intend to evaluate its performance in the presence of interference. We generate a new dataset of signals for testing the classifier. Every signal present in the original testing dataset will be combined with another signal from the dataset. Each of the interfering signals will have an aleatory selected modulation scheme and SNR value, higher or equal to 16 dB. Moreover, the interference signals will overlap the original signals during a percentage of their length, after being multiplied by an attenuation factor $\alpha \in [0, 0.1, \sqrt{10}/10, 0.5]$.

In Fig. 5 we can appreciate the degradation of the performance of the model in the presence of interference. It is

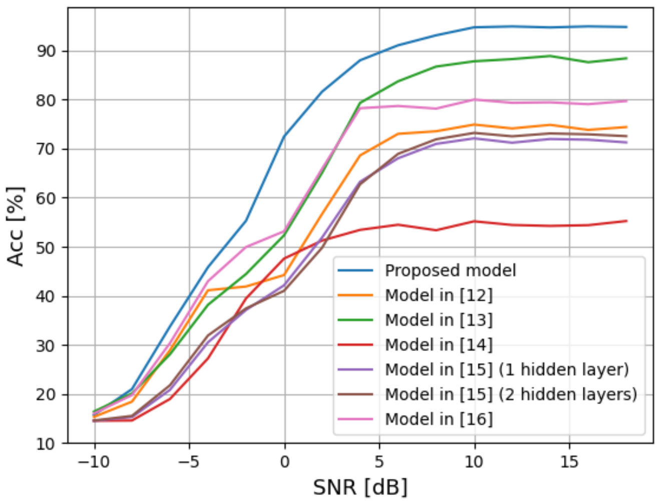

Fig. 4. Variation of the accuracy of different classifiers with the SNR value.

worth noticing that the performance of the classifier presents a greater degradation for signals with high SNR values, rather than for signals with an SNR closer to 0 dB, as the value of α increases. This behaviour was previously reported in [27] with the use of a different classifier.

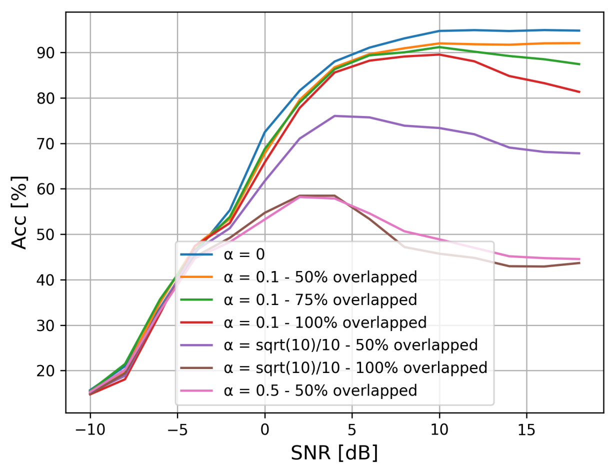

Fig. 5. Variation of the accuracy of the classifier with the SNR value of differently interfered signals.

VIII. Conclusion

The purpose of this paper was to propose the use of a single hidden layer neural network to classify signals of seven different modulation types. This architecture would be beneficial for IoT devices that do not have the capability to process a large number of parameters because of power or memory restrictions.

The primary difference with many previous studies that develop simple classification models comes from the use of a dataset in which the signals are considered to pass through multiple channel effects. Moreover, we perform a selection of input features through the application of the mRMR criterion of the mutual information algorithm.

The final neural network achieves an accuracy of more than 90% when the SNR of the input data is higher than 4 dB. Although this performance could still be improved upon, it is important to note that the model has a low complexity and small number of trainable parameters, allowing it to be trained with a reduced amount of training data examples. This characteristic will be of great use in future works where the intention will be not to use synthetically generated datasets, but signals obtained through the use of SDR devices.

References

[1] S. Hanna, C. Dick, and D. Cabric, "Signal processing-based deep learning for blind symbol decoding and modulation classification," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 82–96, 2022.

[2] B. Dong, Y. Liu, G. Gui, X. Fu, H. Dong, B. Adebisi, H. Gacanin, and H. Sari, "A lightweight decentralized-learning-based automatic modulation classification method for resource-constrained edge devices," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 24 708–24 720, 2022.

[3] M. Abdel-Moneim, W. El-Shafai, N. Abdel-Salam, E.-S. El-Rabaie, and F. Abd El-Samie, "A survey of traditional and advanced automatic modulation classification techniques, challenges and some novel trends," *International Journal of Communication Systems*, 05 2021.

[4] T. Huynh-The, Q.-V. Pham, T.-V. Nguyen, T. T. Nguyen, R. Ruby, M. Zeng, and D.-S. Kim, "Automatic modulation classification: A deep architecture survey," *IEEE Access*, vol. 9, pp. 142 950–142 971, 2021.

[5] C. Roy, S. S. Yadav, V. Pal, M. Singh, S. K. Patra, G. Sinha *et al.*, "An ensemble deep learning model for automatic modulation classification in 5g and beyond iot networks," *Computational Intelligence and Neuroscience*, vol. 2021, 2021.

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

[7] 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)*, 2017, pp. 695–700.

[8] M. Usman and J.-A. Lee, "Amc-iot: Automatic modulation classification using efficient convolutional neural networks for low powered iot devices," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2020, pp. 288–293.

[9] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," 2016.

[10] M. Z. Nisar, M. S. Ibrahim, M. Usman, and J.-A. Lee, "A lightweight deep learning model for automatic modulation classification using residual learning and squeeze–excitation blocks," *Applied Sciences*, vol. 13, no. 8, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/8/5145

[11] A. Hazza, M. Shoaib, S. A. Alshebeili, and A. Fahad, "An overview of feature-based methods for digital modulation classification," in *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, 2013, pp. 1–6.

[12] J. J. Popoola and R. van Olst, "Effect of training algorithms on performance of a developed automatic modulation classification using artificial neural network," in *2013 Africon*, 2013, pp. 1–6.

[13] A. Alarbi and O. Alkishriwo, "Modulation classification based on statistical features and artificial neural network," 05 2021, pp. 748–751.

[14] M. M. Roganovic, A. M. Neskovic, and N. J. Neskovic, "Application of artificial neural networks in classification of digital modulations for software defined radio," in *IEEE EUROCON 2009*, 2009, pp. 1700–1706.

[15] J. Jagannath, N. Polosky, D. O'Connor, L. N. Theagarajan, B. Sheaffer, S. Foulke, and P. K. Varshney, "Artificial neural network based automatic modulation classification over a software defined radio testbed," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[16] Z. Kaleem, M. Ali, I. Ahmad, W. Khalid, A. Alkhayyat, and A. Jamalipour, "Artificial intelligence-driven real-time automatic modulation classification scheme for next-generation cellular networks," *IEEE Access*, vol. 9, pp. 155 584–155 597, 2021.

[17] J. Lee, B. Kim, J. Kim, D. Yoon, and J. W. Choi, “Deep neural network-based blind modulation classification for fading channels,” in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, 2017, pp. 551–554.

[18] Deepsig-Inc. Rf datasets for machine learning. [Online]. Available: https://www.deepsig.ai/datasets

[19] G. Callebaut, G. Leenders, J. Van Mulders, G. Ottoy, L. De Strycker, and L. Van der Perre, “The art of designing remote iot devices—technologies and strategies for a long battery life,” *Sensors*, vol. 21, no. 3, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/3/913

[20] R. Holtzman, “Wideband frequency modulation applications and techniques.” *Microwave Journal*, vol. 58, 2015.

[21] S. Bharadwaj and R. Sharma, “A comparative study of different amplitude modulation signals,” 07 2021.

[22] H. L. Hirsch, “Statistical signal characterization-new help for real-time processing,” in *Proceedings of the IEEE 1992 National Aerospace and Electronics Conference NAECON*. IEEE, 1992, pp. 121–127.

[23] N. B. Gallagher, “Savitzky-golay smoothing and differentiation filter,” *Eigenvector Research Incorporated*, 2020.

[24] A. Abdelmutalab, K. Assaleh, and M. El-Tarhuni, “Automatic modulation classification based on high order cumulants and hierarchical polynomial classifiers,” *Physical Communication*, vol. 21, pp. 10–18, 2016.

[25] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[26] G. Panchal, A. Ganatra, Y. Kosta, and D. Panchal, “Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers,” *International Journal of Computer Theory and Engineering*, vol. 3, no. 2, pp. 332–337, 2011.

[27] P. Triantaris, E. Tsimbalo, W. H. Chin, and D. Gündüz, “Automatic modulation classification in the presence of interference,” in *2019 European Conference on Networks and Communications (EuCNC)*. IEEE, 2019, pp. 549–553.

Yasmín R. Mondino-Llermanos received the degree in electronic engineering from the National University of Cordoba (UNC), Cordoba, Argentina, in 2020. Since 2019, she has been part of the Digital Communications Laboratory, Department of Electronic Engineering, UNC. She is currently pursuing the Ph.D. degree at the same university. Her research areas are in the fields of signal processing, machine learning and satellite communications.

Graciela Corral-Briones received the Electrical and Electronic Engineering and Ph.D. degrees from the National University of Cordoba (UNC), Cordoba, Argentina, in 1991 and 2007, respectively. From 1991 to 1993, she was with the Center for Research in Materials, Cordoba, as a Research Fellow. From 1993 to 1996, she received a fellowship from CONICOR (Scientific and Technological Research Council of Cordoba) to develop analysers for communication protocols. Since March 1996, she has been with the Digital Communications Laboratory, Department of Electronic Engineering, UNC. Her research interests lie in the areas of wireless communications and signal processing, including multiuser detection, channel coding, and MIMO systems.