Journal of Aeronautics and Space Technologies

17(2) (2024) 18-45

Journal Homepage: jast.hho.msu.edu.tr

RESEARCH ARTICLE

Depth Analysis in Deep Learning-Based Automatic Modulation Classification

Derin Öğrenme Tabanlı Otomatik Modülasyon Sınıflandırmasında Derinlik Analizi

Osman Kaya[1*], Tansal Güçlüoğlu[1], Hacı İlhan[1]

[1] *Yıldız Teknik University, Department of Electronics and Communication Engineering, 34220 İstanbul, Türkiye*

Received: *January 2, 2024* ***Revised:*** *February 1, 2024* ***Accepted:*** *February 6, 2024*

Abstract

Automatic Modulation Classification (AMC) is the process of determining the modulation type of a signal received by a communication system. Deep learning, a machine learning technique, has recently garnered significant attention due to its outstanding ability to classify intricate data structures. This study delves into the critical role of automatic modulation classification processes in both civil and military applications, utilizing Convolutional Neural Networks (CNN) as a deep learning approach. In this study, unlike other studies, the effect of changing the depth level of the network on the accuracy level was investigated.

Keywords: Modulation, Classification, Convolutional Neural Networks

Öz

Otomatik Modülasyon Sınıflandırması (AMC), bir iletişim sistemine gelen sinyalin modülasyon türünü belirleme sürecidir. Derin öğrenme, karmaşık veri yapılarını sınıflandırmak konusundaki üstün performansı nedeniyle son zamanlarda önemli ilgi çekmiş bir makine öğrenimi tekniğidir. Bu çalışma, otomatik modülasyon sınıflandırma süreçlerinin hem sivil hem de askeri uygulamalardaki kritik rolüne odaklanarak, derin öğrenme yaklaşımlarından biri olan Evrişimsel Sinir Ağları'nı (CNN) kullanmaktadır. Bu çalışmada, diğer çalışmalardan farklı olarak ağın derinlik düzeyinin değiştirilmesinin doğruluk düzeyine etkisi incelenmiştir.

Anahtar Kelimeler: Modülasyon, Sınıflandırma, Evrişimsel Sinir Ağları

*Corresponding Author
E-mail: osmnkaya11@gmail.com

1. INTRODUCTION

The process of adapting the characteristics of a transmission medium with a specific frequency electrical signal, known as the carrier of information, is referred to as modulation. It is also expressed as the alteration of the form of information before being incorporated into the communication process. The success of communication in a communication system is dependent on the demodulation of the modulation signal sent by the transmitter by the receiver. The prerequisite information needed to detect the type of modulation is often not available, especially in applications requiring confidentiality, primarily in military contexts. Automatic Modulation Classification (AMC) refers to the automatic recognition and categorization of the modulation format of a signal perceived by the receiver without the need for any prior information. The AMC process serves as an intermediate step between signal detection and demodulation for the receiver. How AMC is incorporated into a communication system is illustrated in Figure 1.

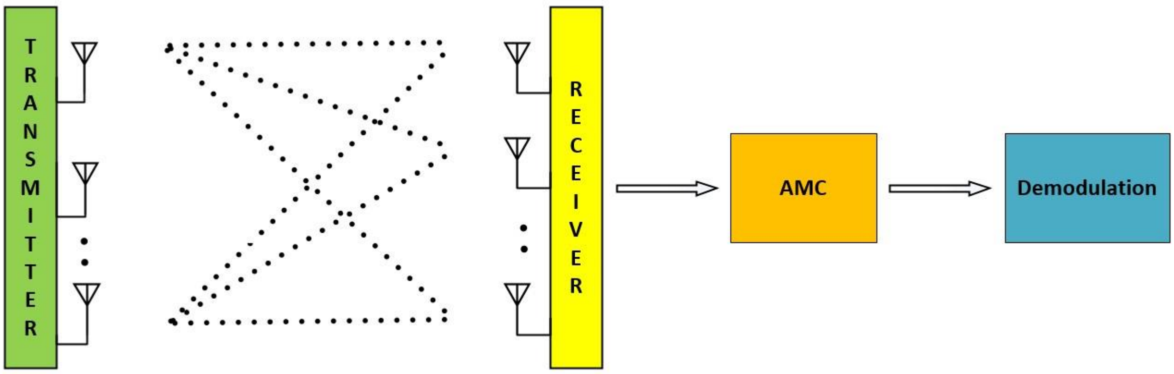

Figure 1. Block diagram of modulation classification in communication system.

The structure of AMC is expected to have high accuracy, low processing complexity, and the ability to serve multiple purposes. In the literature, studies in this regard, termed as traditional methods, are divided into two main categories: likelihood-based and feature-based [1, 2, 3]. Meanwhile, the deep learning approach is referred to as an advanced modulation classification method. The use of deep learning in AMC has been rapidly increasing in recent years. In modulation classification, likelihood-based methods such as Maximum Likelihood (ML), Average Likelihood Ratio (ALR), and feature-based methods utilizing characteristics like moments, Fourier transformation, and constellation points are referred to as traditional modulation classification methods [1-3]. In a study proposing a feature-based method aiming to extract constellation shapes from received information after noise and channel-induced distortions of the transmitted information [4], fuzzy c-means clustering was used for digital modulation classification. The study demonstrated that classification could be achieved using amplitude and phase modulation, initially employing ML classification for the latter [5]. To mitigate the high computational complexity of ML, a minimum distance (MD) classifier was suggested, and its performance was compared. Another study [6] focusing on the classification of phase shift keying (2/4/8-QPSK), quadrature amplitude modulation (16/64-QAM), and amplitude shift keying (4-ASK) modulations utilized local binary pattern as a feature. Extracted features were processed through a single hidden layer feedforward classifier. The proposed approach was mentioned to be faster compared to likelihood-based methods. In the realm of deep learning for modulation classification, studies involve classical deep neural network architectures such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long-Short Term Memory (LSTM), and hybrid structures combining these architectures. Some works not utilizing CNN in deep learning-based modulation classification studies are presented here. For instance, a study [7] employing Deep Belief Networks (DBN) classified frequency shift keying (FSK), PSK, minimum shift keying (MSK), and QAM modulations. The performance of DBN was observed to be higher compared to feature-based Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) methods. Another study [8] used Recurrent Neural Networks (RNN) to classify BPSK, QPSK, 8-PSK, and 16-QAM modulations under the presence of Rayleigh fading channel, white Gaussian noise, non-white Gaussian noise, and correlated non-white Gaussian noise. The study conducted in Python within the scope of this work observed the success of RNNs over traditional machine learning methods in scenarios where channel parameters were unknown. In a study utilizing LSTM networks on the RadioML.2016a dataset [9], the classification performance of IQ data and fourth-order cumulants obtained from IQ data was examined. The study investigated the impact of input data on LSTM, revealing close results for both types of data.CNN and hybrid networks containing CNN studies in the literature are presented in this section. A pioneering work in the field of automatic modulation classification (AMC), O'Shea et al. [10], compared radio modulation classification results achieved with feature-based methods to those using deep networks such as CNNs. The study demonstrated that CNNs trained on IQ data outperformed general machine learning algorithms in AMC. In another study treating communication signals as 2D data similar to images [11], the performance of a 2D CNN architecture on a dataset containing BPSK, QPSK, 8PSK, 16QAM, and 64-QAM signals in IQ format was examined. The classification performance of the classifier was evaluated for modulation recognition in Doppler channels with frequencies of 100 Hz and 300 Hz.A study incorporating 5 analog modulation types (FM, AMSSBWC/AMDSBWC, AMSSBSC/AMDSBSC) and 16 digital modulation types (OOK, ASK, PSK, APSK, QAM, GMSK, OQPSK) [12] compared the modulation classification performance of two different CNN structures to feature-based methods. In another work, principal component analysis (PCA) was employed to reduce data dimensions in low Signal-to-Noise Ratio (SNR) conditions for CNN, DenseNet, Convolutional Long-Short Term (CLDNN), LSTM, and ResNet architectures [13]. The study observed enhanced performance for these architectures with PCA preprocessing and uniform sampling, particularly at low SNR values.In a comparison study of four different deep learning architectures (CLDNN, CNN, ResNet, and DenseNet) for modulation classification [14], DenseNet demonstrated significantly better results than ResNet due to its strengthening of feature transfer. A study on classifying 10 different modulation formats in channels modeled with Rayleigh fading and white Gaussian noise [15] used stochastic features such as power spectral density from features like amplitude, phase, frequency. A decision tree classifier was employed for modulation classification. Another study [16] using LSTM structures evaluated the performance by using amplitude and phase data as input for classifying RadioML.2016a dataset symbols. A study [17] employing GRU architecture as input for RadioML dataset observed that using more than two layers did not significantly improve performance.A 5-layer CNN model was used to classify modulation formats such as 16/64-QAM, 2/4/8-FSK, 4/8-ASK, 2/4/8-PSK in a study conducted in MATLAB, achieving over 80% accuracy for SNR values above 10 dB [18]. In another study [19] comparing CNN, CNN-LSTM, and SVM structures, CNN-LSTM outperformed due to its ability to capture temporal relationships. A study [20] utilizing CNN and LSTM methods in serial and parallel structures outperformed SVM, with the serial architecture performing better than the parallel one. Another study [21] investigated CNN performance in different batch sizes and optimization methods using the RadioML.2018b dataset, achieving a maximum of 81% accuracy.A dataset containing 15 different digital modulation types was created in a study [22] by corrupting modulated signals with white Gaussian noise. CNN classification results were further improved by applying SVM to achieve a classification accuracy of 99.1%. Another study [23] employed CNN and GRU networks as inputs for high-order cumulant, SNR, and spectrum features to classify modulation types such as ASK, PSK, QAM. In a study using MATLAB-generated ASK, PSK, and QAM modulations, features such as frequency-offset suppressed constellation diagrams were extracted from IQ signals. The study trained GRU, RNN, and CNN networks on these features, demonstrating that the networks achieved similar performances and highlighting the positive impact of suppressing frequency offset.

The following are the contributions of this article to the literature:

- The effect of the level of depth of the network on the performance has been examined.
- The achievements of the synthetic data set produced with GNU and the natural data set have been shown separately.

In the second section, the "Aim and Objective" section will be discussed. In the third section, the "Data Set" section will be discussed. In the fourth section, the "Methodology" section will be discussed. In the fifth section, the "Simulation Results" section will be discussed. In the sixth section, the "Conclusion" section will be discussed. In the seventh section, the "Future Research" section will be discussed.

2. AIM and OBJECTIVE

For the receiver to demodulate the signal accurately, it needs to be aware of the different modulation types present in the data it is working on. This process, crucial in wireless communication systems, is referred to as AMC. It is also defined as the process of determining the modulation scheme of a specific radio signal without manual intervention. AMC is employed in various communication problems, such as efficient spectrum utilization, signal detection in the spectrum, identification and monitoring of enemy communication in military applications, counteraction against electronic warfare situations, managing network resources based on the classification of different traffic types in next-generation communication technologies, channel estimation, and error correction.As mentioned, in the realms of wireless communication, electronic warfare, and signal processing, there are classical and deep learning-based methods for automatic modulation classification that play a significant role in ensuring the efficient and effective operation of these systems. Within the scope of this study, an AMC process was implemented using deep neural networks, addressing challenges encountered in classical methods such as feature extraction, limitations with specific modulation sets, and adaptability issues to variable communication conditions like noise and fading channels. The goal of this study is to develop a high-performance AMC process using convolutional neural networks capable of learning complex features from raw data, handling various and large amounts of data, demonstrating high generalization ability to work with different datasets, and exhibiting robustness against noise and interference.

3. DATA SET

The RadioML2016.a dataset has been created for the development and evaluation of machine learning algorithms for the modulation classification of radio signals. This dataset is specifically designed to provide a realistic representation of radio signals commonly used in wireless communication systems. Like many other datasets in the field of wireless communication and signal processing, it was generated using the GNU Radio software. GNU Radio is a free and open-source software development tool that provides a wide range of signal processing blocks and tools to create software-defined radio systems. It allows for the easy configuration and combination of various modulation schemes, noise sources, and signal generators to create diverse radio signals. This software can be used to create software-defined radios in a simulation environment without the need for hardware.

During the creation of the dataset, time-varying multipath fading for channel impulse response, randomness in carrier frequency oscillation and sampling times, and incremental Gaussian white noise were employed. The generated signal sets were passed through a rigid channel model that added unknown scale, shift, expansion, and noise. These processes were carried out to accurately characterize the effects of radio channel impairments. Realistic, non-ideal effects such as thermal noise, oscillator drift, symbol timing offset, sample rate offset, carrier frequency offset, and phase difference were reflected in the data.

The initial part of the study focused on BPSK, QPSK, 64QAM, BFSK, CPFSK, PAM4, and AM-DSB modulations derived from the RadioML 2016.10a dataset. Each modulation type contains 1000 samples for each SNR value within the -20 to 18 dB range, with a step size of 2 dB. Each sample is in IQ format and consists of 128 sampling points. Thus, for each modulation type, there is a vector of size 2x128x1000 for every SNR value. With seven modulations, 20 different SNR values, and 1000 examples for each SNR value, the labeled dataset used in the study is of size 140,000 examples.

In the second part of the study, BPSK, BFSK, QPSK, QFSK, 16-QAM, and 64-QAM modulations were generated in MATLAB. This dataset incorporated a Rician multipath fading channel and additive white Gaussian noise (AWGN). Similar to RadioML2016.a, each modulation type contains 1000 examples for each SNR value within the -20 to 18 dB range. The size of the dataset generated in MATLAB is 2x128x100 for each modulation type, resulting in a dataset size of 120,000 examples.

Table 1 summarizes the characteristics of the datasets. The RadioML2016.a dataset includes labels for modulations, and labels for the dataset prepared in the second part were also created in MATLAB. The labels for both datasets were generated using one-hot encoding, representing the binary representation of categorical variables. In this representation, the label for each modulation type is a binary number with a length equal to the number of classification classes, with only one digit being 1, and the rest 0. In the RadioML2016.a dataset, with seven different modulations, the label length for modulations is 7. For example, the label vector for BPSK is [1 0 0 0 0 0 0], and for QPSK, it is [0 1 0 0 0 0 0]. In the second part of the study, with six modulations, the label for BPSK in the MATLAB dataset is [1 0 0 0 0 0], and for QPSK, it is [0 1 0 0 0 0]. The label vectors for each modulation type in the datasets are prepared in such a way that the position of the digit 1 varies without compromising the understanding.

Table 1. Characteristics of the data sets used in the study.

Data Set	Modulation Types	Sample Size	Data Set Size	SNR Range (dB)
RadioML 2016.10a	BPSK, QPSK, CPFSK, GFSK, 4-PAM, AM-DSB, QAM64	2x128	140000	-20:2:18
MATLAB	BPSK, BFSK, QPSK, QFSK,16 QAM, 64-QAM			

Modulation is the process of altering one or more properties of a high-frequency carrier signal, such as its amplitude, frequency, or phase, to transmit information over a communication channel. It is a crucial technology used in various communication systems, including radio, television, satellite, cellular, and fiber optic networks, to efficiently and reliably transmit information over long distances. The properties of a signal, such as amplitude, frequency, phase, and spectrum, play a significant role in the applications of communication systems. The amplitude of a signal determines the quality and power of the received signal. Frequency determines the bandwidth of the signal, which is crucial for determining data rates and preventing interference with other signals in the same frequency range. Phase determines the timing of the signal relative to a reference signal and is essential for synchronization between the transmitter and receiver.

The modulation scheme, frequency band, and channel coding scheme selection are influenced by the signal properties and their effects on the communication system. In modulation, the amplitude, frequency, and phase of the carrier signal are crucial as they determine the characteristics of the modulated signal carrying the information. The modulation process involves manipulating one or more of these properties to embed the information into the carrier signal. The selection of the modulation scheme and the values of amplitude, frequency, and phase used in modulation depend on various factors such as the bandwidth of the signal, the desired data rate, and the noise characteristics of the channel.

The characteristics of the modulated signal are carefully designed to optimize the performance of the communication system and ensure the reliable transmission of conveyed information. Below are discussed the features of the modulation schemes used in the scope of the study.

BPSK (Binary Phase Shift Keying): BPSK is a type of digital modulation scheme where the phase of the carrier signal is shifted to represent either 1 or 0. The phase of the carrier signal is shifted by 180 degrees depending on the value of the transmitted binary data. It is a relatively simple and efficient modulation technique with modest bandwidth requirements. BPSK is resistant to noise and interference, and its hardware implementation is straightforward. However, it is sensitive to phase ambiguity and fading, which can introduce errors in the demodulated signal. Additionally, it can transmit only one bit per symbol, limiting the data rate. It is commonly used in satellite communication, RFID (Radio-Frequency Identification), and wireless networks like Wi-Fi and Bluetooth.

QPSK (Quadrature Phase Shift Keying): QPSK is a digital modulation scheme where the phase of the carrier signal is shifted to represent two bits of binary data. The phase of the carrier signal is shifted by 90, 180, or 270 degrees depending on the values of the two transmitted binary data bits. It is more bandwidth-efficient than BPSK, as it can transmit twice as much data within the same bandwidth. QPSK also exhibits good resistance to noise and interference, making it suitable for many wireless communication standards. However, it requires more complex demodulation compared to BPSK, as it necessitates the use of two demodulators to recover the in-phase and quadrature components of the signal. QPSK is commonly used in satellite communication, digital cable TV, and cellular networks such as 3G, 4G, and 5G.

PFSK (Continuous Phase Frequency Shift Keying): CPFSK is a digital modulation scheme where the carrier signal's phase is shifted based on the frequency deviation of the input signal. It uses a continuous phase signal and a frequency-modulated carrier wave to represent digital information. The modulated signal's phase remains constant, but the frequency is slightly shifted to represent changes in the digital signal. CPFSK signals have a spectrum similar to frequency-modulated (FM) signals. It is an efficient modulation scheme in terms of bandwidth, can transmit data at higher rates compared to BPSK and QPSK, and exhibits good resistance to noise and interference. However, it can be affected by frequency deviations, potentially causing errors in the demodulated signal, and CPFSK requires more complex hardware compared to BPSK and QPSK.

GFSK (Gaussian Frequency Shift Keying): GFSK is a digital modulation scheme where the carrier signal's frequency deviation is modulated by a Gaussian-shaped pulse. The pulse shape determines the amount of frequency shift, while the modulated signal's amplitude and phase remain constant. GFSK signals have a spectrum similar to Gaussian-filtered frequency-modulated (FM) signals. It is resistant to noise and interference, bandwidth-efficient, and has a smooth frequency spectrum, making it useful for applications requiring low spectral spillover. However, GFSK requires more complex hardware than BPSK and QPSK, and it can be affected by frequency offsets and phase noise. Common applications of CPFSK and GFSK include wireless communication systems like Bluetooth and wireless LANs.

PAM-4 (Pulse Amplitude Modulation 4): PAM-4 is a digital modulation type that uses four amplitude levels to represent two bits at once. The amplitude levels are typically at -3, -1, 1, and 3 times the carrier amplitude, and the modulated signal has a single frequency equal to the carrier frequency. It is resistant to noise and interference, bandwidth-efficient, but sensitive to distortion and requires accurate timing and amplitude synchronization in the receiver. PAM-4 has limited data rates compared to other modulation schemes and is commonly used in fiber optic communication, DSL (Digital Subscriber Line) modems, and Ethernet networks.

AM-DSB (Amplitude Modulation with Double Sideband): AM-DSB is an analog modulation scheme where the carrier signal's amplitude is modulated by the input message signal. Modulating the carrier signal's amplitude produces two sidebands with equal amplitude, one above and one below the carrier frequency. The modulated signal's bandwidth is twice the bandwidth of the modulating signal. AM-DSB is a simple and easy-to-implement modulation scheme used in broadcasting and audio applications. It has limited data rates compared to other modulation schemes, inefficient use of bandwidth, is sensitive to noise and interference, and suffers from poor spectral efficiency.

QAM64 (Quadrature Amplitude Modulation 64): QAM64 is a type of digital modulation scheme where both the amplitude and phase of the carrier signal are modulated to represent multiple levels of data simultaneously. QAM64 uses 64 different combinations of amplitude and phase for each symbol, allowing it to transmit more data in a single symbol compared to BPSK or QPSK. It is an efficient modulation scheme that can transmit a large amount of data in a small bandwidth. However, it requires accurate amplitude and phase synchronization in the receiver and is sensitive to noise and interference. QAM64 is commonly used in digital cable TV, DSL modems, and cellular networks like 4G and 5G.

In general, the choice of modulation schemes depends on factors such as the transmission medium, bandwidth, power requirements, desired data rate, and error rate. Different modulation schemes have various advantages and disadvantages, and the selection of the modulation scheme depends on the requirements of the specific application.

4. METHODOLOGY

The performance of modulation classification has been investigated within the scope of the study through seven different deep neural network scenarios. The results obtained when two-dimensional CNN (Conv2D) and fully connected (dense) structures are used with different numbers of layers and filters have been compared. The characteristics of the scenarios used in the study are given in Table 2. The performance results for each scenario are explained in the Simulation Results section.

Table 2. Scenario features used in the study.

Scenario	Layer Characteristics	Number of Filters
1	Conv2D	32
2	Conv2D	64
3	Conv2D	128
4	Conv2D Conv2D	128 64
5	Conv2D Conv2D Conv2D	128 64 32
6	Conv2D Conv2D	256 64
7	Conv2D Conv2D Dense	256 64 256

Simulation in the scope of the study was conducted using the Keras library. In this section of the thesis, the features of the convolutional and fully connected layers of the library used in the study are discussed to provide a comprehensive understanding of the layer properties and simulations used.

In Keras, the Conv2D layer is typically a 2D convolutional layer used for image processing tasks such as image recognition, object detection, and segmentation. It operates with a series of filters that move across the input data, using sliding kernels of a specific size. Each kernel is a small weight matrix learned during the training process. During the convolution process, the kernels are applied to each part of the data, resulting in a feature map that highlights the locations where features are detected. Feature maps are further processed by additional layers to produce the final output of the neural network. The number of filters in a convolutional layer determines the number of feature maps produced as output from that layer.

On the other hand, the Dense layer in Keras is a fully connected layer, meaning each neuron in the layer is connected to every neuron in the previous layer. This layer takes the outputs from the previous layer and computes a weighted sum of these outputs for each neuron in the current layer. Each neuron has a set of weights and biases learned during the training process, determining the importance of each input in calculating the output. In summary, the Dense layer in Keras is a fully connected layer that calculates a weighted sum of inputs for each neuron in the layer using learned weights and biases during the training process. The output of each neuron passes through a non-linear activation function to add non-linearity to the output, and the final layer output is transferred to the next layer in the neural network.

For the filter kernels used in the study, the Glorot uniform initializer was employed. The Glorot initializer is a widely used method to initialize the weights of neural network layers. The main idea behind the Glorot initializer is to initialize the weights of a layer in such a way that the activations of the layer stay within a certain range during training. This helps avoid issues related to exploding or vanishing gradients that may occur when weights are initialized to very small or large values. The initializer samples weights from a uniform distribution within a certain range, determined by the number of input and output connections in the layer, as well as the activation function used in the layer.

The choice of kernel initializer is crucial for the backpropagation algorithm, which is used to update the weights of a neural network during training. Poor selection can lead to issues with gradients during backpropagation, resulting in the memorization of data, i.e., overfitting during the training period. Additionally, it can cause training to slow down and require higher memory demands. The selection of the number of filters affects the model's performance and should be carefully chosen based on the specific problem under consideration and the available resources. The same applies to increasing the number of layers; adding more layers does not always guarantee improved performance. However, it facilitates the learning of complex features and enhances the model's ability to extract meaningful information from the input. The optimal number of layers depends on the problem at hand. There is no definitive method for finding the best model in a problem involving deep learning; it is essential to experiment with different network architectures and hyperparameters to discover the most effective configuration.

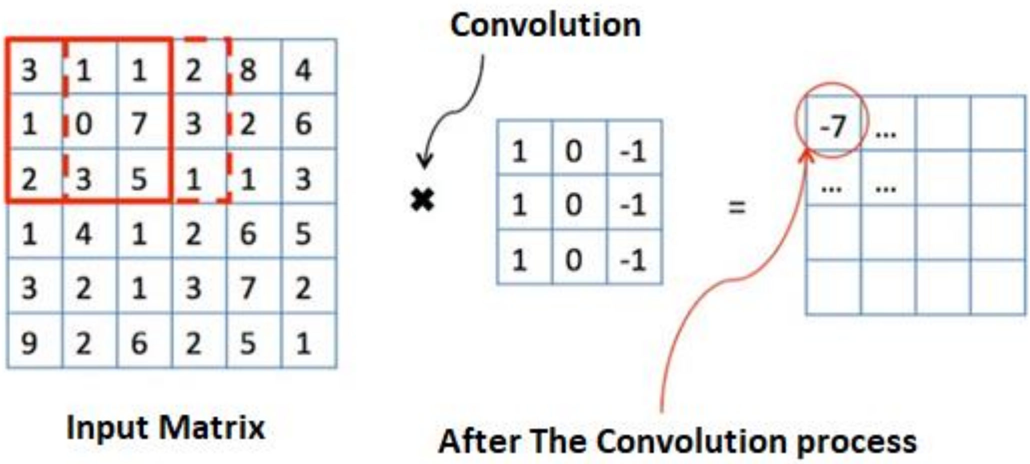

Figure 2. Convolution process[24].

Figure 2 illustrates the change in the first element of a 6x6 input matrix when convolution is applied with a filter of size 3x3. The kernel matrix is placed on the input matrix, and the convolution result is obtained by multiplying and then summing the values within the overlapping pixels. The filter slides over the input matrix, continuing this process. When the convolution operation is applied to the first element of the input matrix,the result is [(3*1)+(1*0)+(1*(-1))+(1*1)+(0*0)+(7*(-1))+(2*1)+(3*0)+(5*(-1))]=-7.The result obtained after the convolution operation is also referred to as a feature map. After the convolution process, the size of the matrix being worked on decreases. In some applications, to maintain the same size for the input matrix and the matrix obtained after convolution, zero-padding is applied by filling with zeros.

The size of the feature matrix obtained after a CNN layer is calculated as follows:

$$Post - convolution\ dimension = \left(\frac{n_1+2p-f_1}{s_1} + 1\right) x \left(\frac{n_2+2p-f_2}{s_2} + 1\right) \tag{1}$$

Equation 1 represents the dimensions of the input matrix (n_1, n_2), the convolution matrix size (f_1, f_2), the stride values (s_1, s_2) ,and the padding value p.

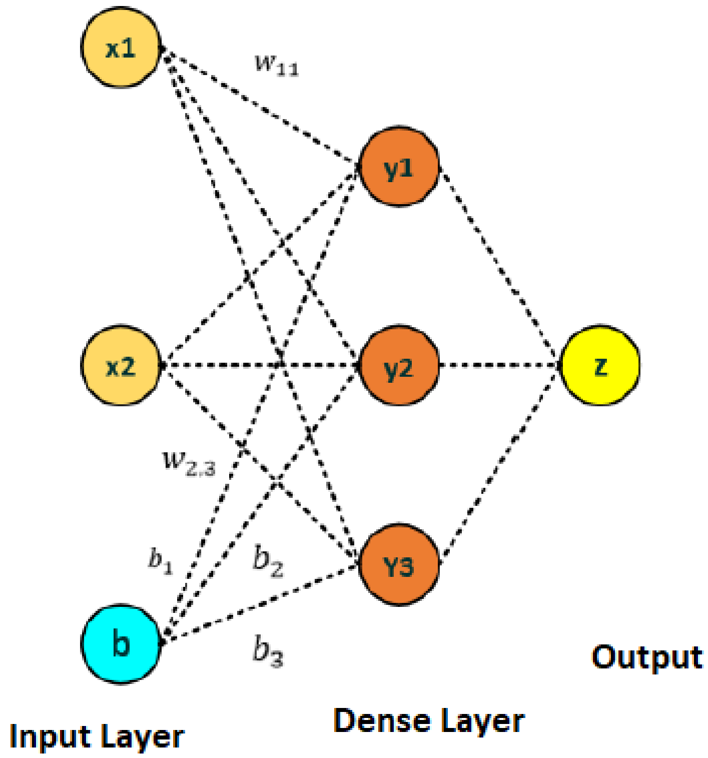

Figure 3. Fully connected layer [25].

Figure 3 illustrates a fully connected layer. The operation of a fully connected layer, used for learning the connection between the inputs and outputs of a neural network, involves two main processes. In the first step, during forward propagation, the input is multiplied by a weight matrix, and the result has a bias vector added to it. This computation is performed for each neuron in the fully connected layer. The output of each neuron is then passed through a non-linear activation function, adding a non-linear feature to the model. The layer's output is obtained by combining the outputs obtained from each neuron.

$$y = f(w * x + b) \tag{2}$$

$$\begin{matrix} y_1 \\ y_2 \\ y_3 \end{matrix} = f\begin{pmatrix} w_{1,1} & w_{2,1} & & & b_1 \\ w_{1,2} & w_{2,2} * & \begin{matrix} x_1 \\ x_2 \end{matrix} & + & b_2 \\ w_{1,3} & w_{2,3} & & & b_3 \end{pmatrix}$$

Equation 2 represents the activation function f, weight matrix w, and bias vector b. In the second step, during the backward propagation, the gradient of the loss function with respect to the layer's output is calculated. This gradient is propagated backward through the layer to compute the gradients of the weights and biases. These gradients are then utilized, with the aid of an optimization algorithm, to update the weights and biases. Within the scope of the study, ReLU has been employed as the activation function for fully connected layers. The Glorot initializer has also been utilized for the initial values of these layers.

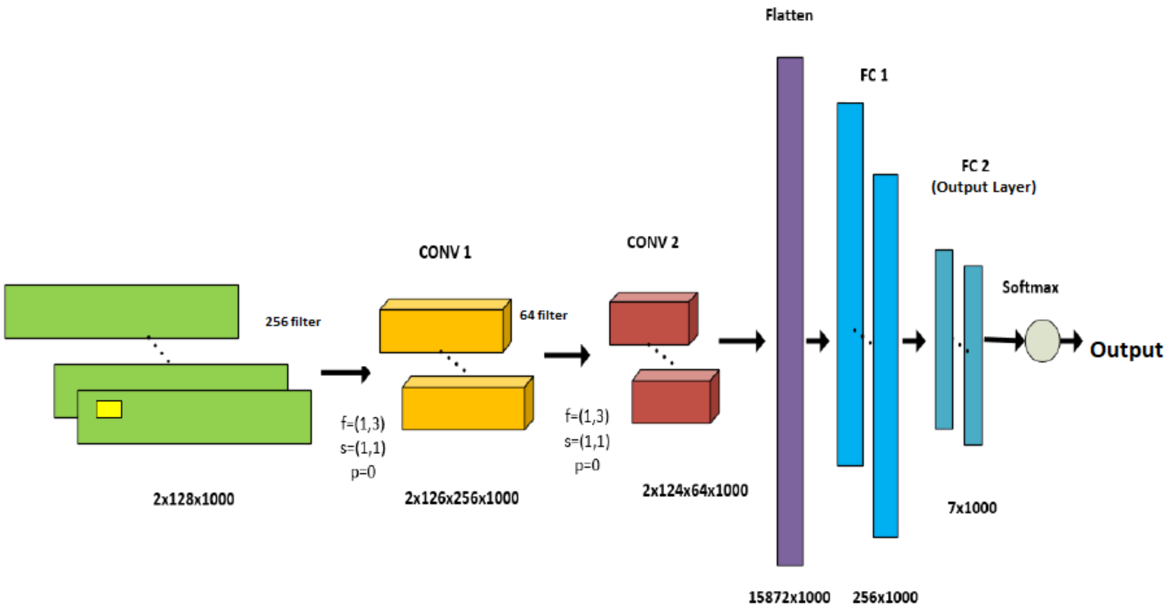

Figure 4. The network model used in scenario 7.

In Figure 4, the model used in Scenario 7 and the parameters of the model layers are shown. It generates input data consisting of $2x128x1000$ samples at any SNR value. After a convolutional layer with $(256,(1,3))$there are $2x126x256x1000$ parameters. Subsequently, with the following layer $(64,(1,3))$, $2x124x64x1000$ parameters are formed. Later, to be processed by the fully connected layer, the output of the second convolutional layer is sent to the flattening layer.

In the scope of the study, Rectified Linear Unit (ReLU) activation function was used for fully connected layers. The Glorot initializer was employed for the initial values of these layers. The purpose of the fully connected layer is to learn complex nonlinear relationships between input and output data by applying a weighted transformation to the input. The output of each neuron in the layer is a linear combination of the outputs of all neurons in the previous layer, weighted by learnable parameters. Finally, since the goal is to detect 7 different modulation types in the RadioML2016.a dataset, a fully connected layer with 7 units is used to reduce the output data size to 7x1000 (or 6x1000 for the dataset generated in Matlab). ReLU activation function is used in each layer. In deep learning models, gradients are calculated through the backpropagation algorithm based on the loss function, and convolutional matrix and fully connected layer weights are updated. The primary reason for using ReLU is to address the vanishing gradients problem that can occur when training very deep neural networks. Vanishing gradients occur when the gradient of the loss function with respect to the network weights becomes very small, preventing the weights from being updated and causing learning to stagnate. ReLU addresses this issue by adding a non-linearity property to the network, allowing it to learn more complex and sophisticated functions. ReLU leaves positive values unchanged while setting all negative input values to zero, making it a simple but effective activation function. The calculation of this non-linear activation function is straightforward and performs well in many deep learning applications. Finally, the obtained results are passed through the softmax function. The softmax function is commonly used in the output layer of deep learning models, especially for multi-class classification problems. In the study, since the goal is to detect one of several modulation classes, softmax assigns a probability value to each modulation class, and these probabilities are normalized to sum to 1. The detected modulation class is the one with the highest probability. Categorical cross-entropy is used as the loss function in the study. The fundamental idea of categorical cross-entropy is to compare the predicted class probabilities extracted by the model with the actual class labels. The loss function calculates the difference between these two distributions and produces a scalar value indicating how well the model is performing. The goal of the model is to minimize the categorical cross-entropy loss, meaning it attempts to make the predicted distribution as close as possible to the actual distribution. The lower the loss, the more accurately the model classifies input data. The training process aims to minimize the loss function, and the optimizer is an algorithm that updates the model parameters during training to minimize the loss function. The optimizer uses the gradients of the loss function with respect to the model parameters to determine how the parameters should be updated. In the study, the adaptive moment estimation (Adam) optimization function is utilized. The basic idea behind Adam is to adapt the learning rate for each parameter based on the past gradients. It maintains moving averages of both the first-order (mean) and second-order (variance) moments of the gradients and uses them to adjust the learning rate for each parameter. This feature enables Adam to converge faster than other optimization algorithms.

5. SIMULATION RESULTS

In the first part of the study, the RadioML 2016.10a Modulation dataset [69] was utilized. This dataset includes labeled data vectors for seven modulation formats: BPSK, QPSK, CPFSK, GFSK, 4-PAM, AM-DSB, and 64-QAM. Each modulation format was corrupted with random noise to have SNR ratios of -20 dB and +18 dB. The dataset consists of a total of 140,000 data vectors, denoted as $x_k \in R^{2x128}$, representing the components of the star constellation diagram, I and Q. For each SNR value, there are 1000 examples for each modulation format. Binary coding was used to create seven modulation class labels, turning the modulation recognition into a classification problem with seven categories.

In the second part of the study, six different modulation types, namely BPSK, BFSK, QPSK, QFSK, 16-QAM, and 64-QAM, were generated and labeled in the MATLAB environment. Similar to the first part, these were classified using a deep neural network. The generated dataset follows the same structure as the RadioML 2016.10a Modulation dataset, with 1000 examples for each modulation format corrupted with random noise in the SNR range of -20 dB to +18 dB. In this dataset, there are a total of 120,000 data vectors $x_k \in R^{2x128}$, representing the components of the star constellation diagram, I and Q. Sixty percent (60%) of the datasets were used for training, while the remaining forty percent (40%) were used for testing. Table 3 provides the characteristics of the datasets used in the study. The simulations of deep neural networks in the study were implemented using the Keras library in the Python programming language.

Table 3. Data sets and network characteristics used in the study.

Data set	RadioML.2016A
RadioML 2016.10a Modulation	BPSK, QPSK, 64-QAM, BFSK, CPFSK, PAM4, AM-DSB
Generated in Matlab	BPSK, BFSK, QPSK, QFSK, 16-QAM, 64-QAM
Data Format	In-phase/Quadrature (IQ)
Data Size	2x128
SNR Range	[-20dB, -18dB, ... , 18 dB]
Simulation environment/Library used	Python/Keras
Total number of samples (RadioML 2016.10a Modulation) Total number of samples (Generated in Matlab)	140000 vector 120000 vector
Number of training samples	60% of the data set
Number of test samples	40% of the data set

As stated in the study, the performance of a 2D Convolutional Neural Network (2D-CNN) structure with various depths and filter numbers was examined for modulation classification across seven different scenarios. The output layer for all scenarios includes a fully connected layer unit equal to the number of modulation types. In Scenario 1, only one CNN layer with 32 filters was used. In Scenario 2, the number of filters was increased to 64, and in Scenario 3, it was further increased to 128, investigating how the number of filters in a single-layer CNN structure affects modulation classification performance. In Scenario 4, a two-layer CNN structure with 128 and 64 filters was used. In Scenario 5, a three-layer CNN structure with 128, 64, and 32 filters was used, examining the impact of the number of layers on performance. In Scenario 6, a two-layer CNN with 256 and 64 filters was employed, and in Scenario 7, after a two-layer CNN structure with 256 and 64 filters, a fully connected layer with 256 units was used. Scenarios 4, 5, and the comparison of Scenarios 4, 6, 7 were used to investigate the effect of filter and layer numbers in multi-layer structures.

In this section of the simulation results, the simulation outcomes and classification performance of the studies conducted on the RadioML 2016a dataset will be discussed. All seven scenarios mentioned in the methodology section were applied to this dataset, and the results obtained were evaluated.

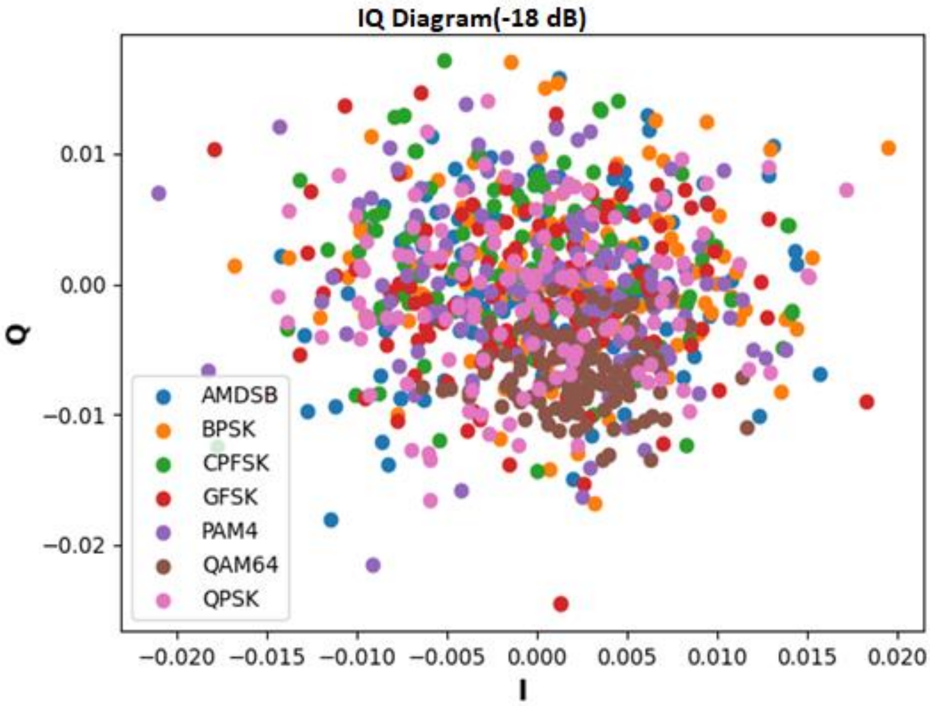

Figure 5. IQ diagram of the data set used at -18 dB.

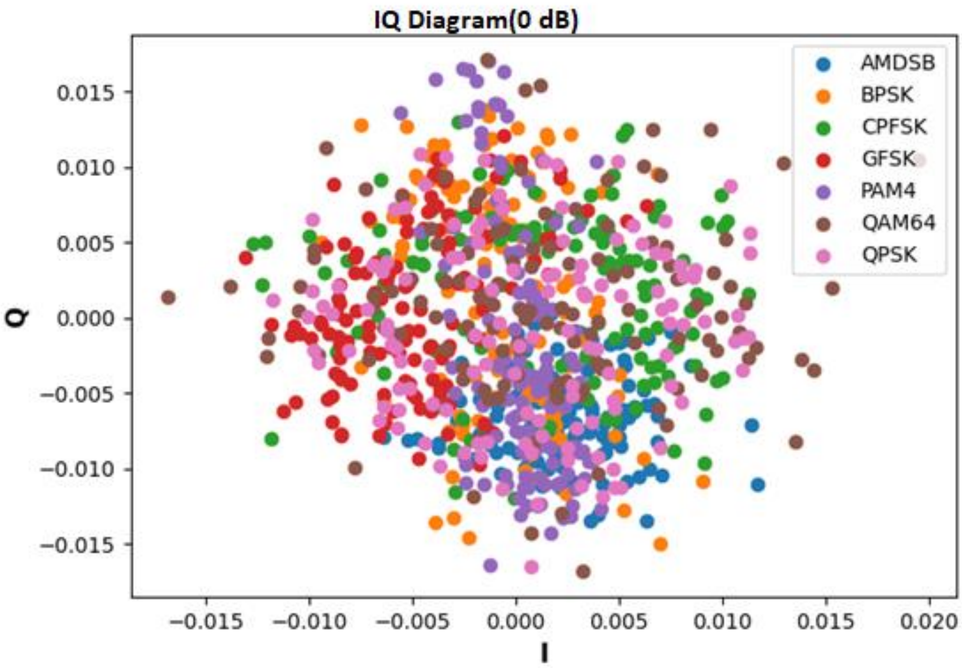

Figure 6. IQ diagram of the data set used at 0 dB.

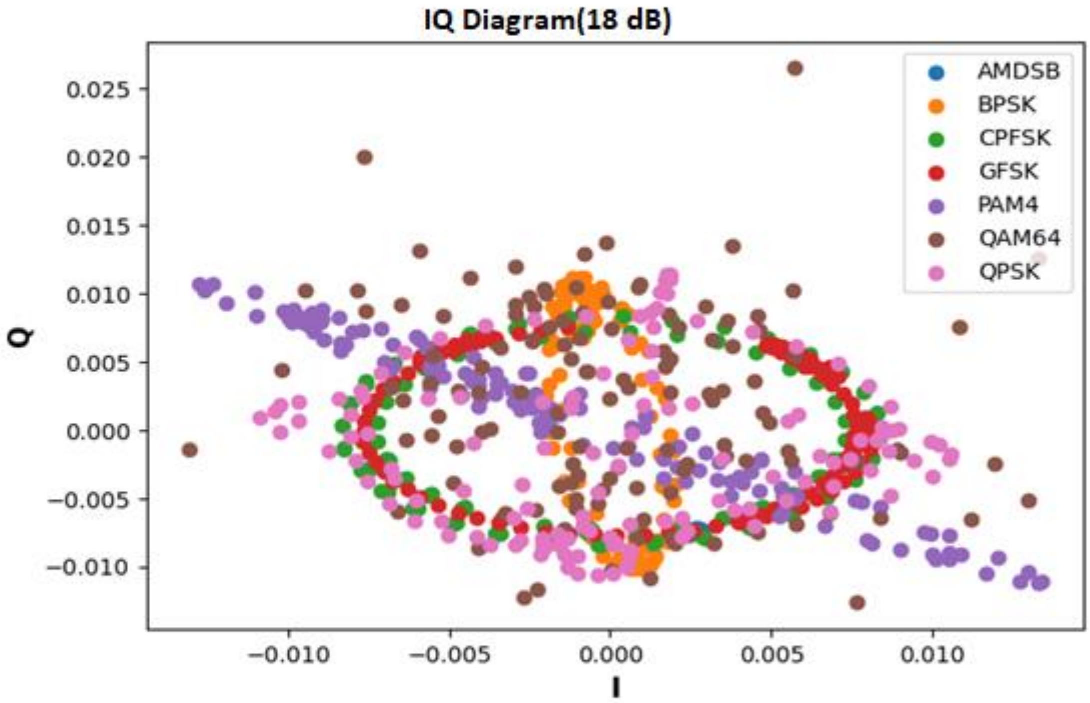

Figure 7. IQ diagram of the data set used at 18 dB.

Figures 5, 6 and 7 depict constellation diagrams for the sample space of the problem for different SNR values. Upon examination of these figures, it is observed that as the SNR value increases, different modulation classes become more distinct.

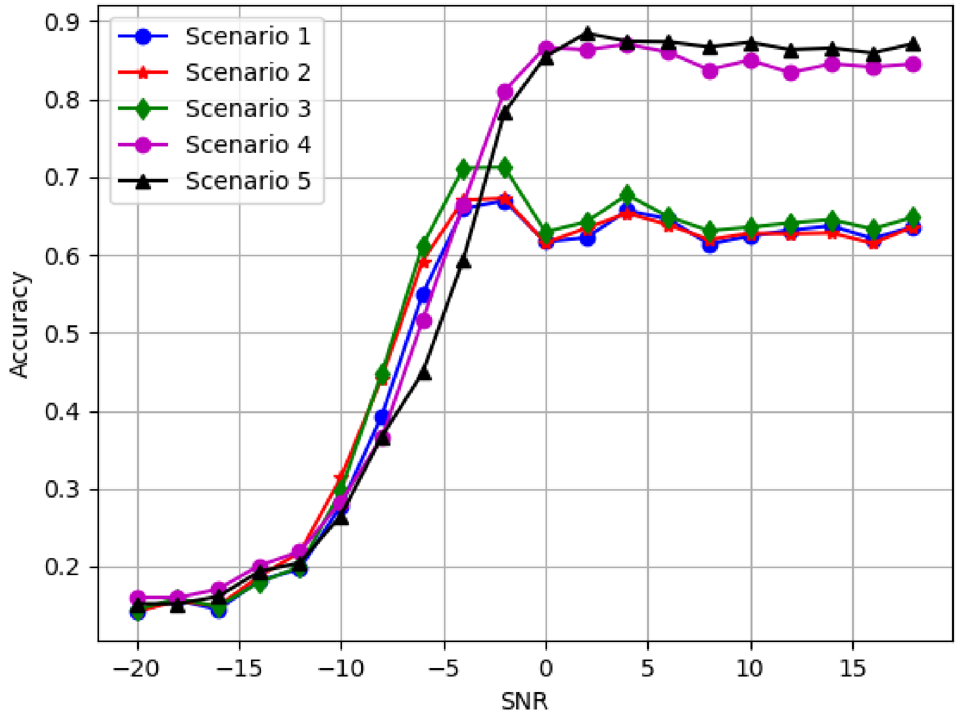

Figure 8. Performance analysis for different filter numbers and CNN network depth.

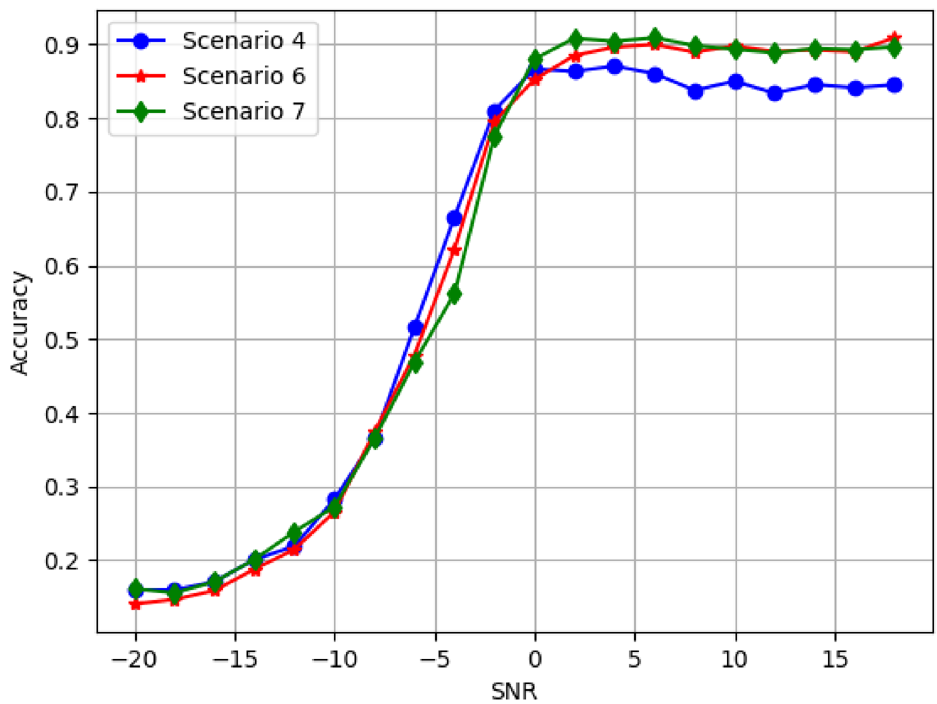

Figure 9. Performance analysis for different filter numbers and CNN network depth.

Scenarios 1, 2, 3, 4, and 5 are compared in Figure 8, and Scenarios 4, 6, and 7 are compared in Figure 9. When only one CNN layer is used, the accuracy remains at the 60% level. Due to a higher number of filters, the performance of Scenario 3 is higher compared to Scenarios 1 and 2. However, even with an increased number of filters, the achieved accuracy level indicates that the network's generalization ability is still low. Analyzing the accuracy graph of Scenarios 4 and 5, it is observed that the performance has improved compared to Scenarios 1, 2, and 3. The use of an additional convolutional layer in Scenario 5, where a third convolutional layer is employed, further enhances the performance. Examining Figure 9, the increase in the number of filters positively impacts the performance, as clearly observed in the accuracy graph of Scenarios 4 and 6. When examining the accuracy graph of Scenarios 6 and 7, the addition of a fully connected layer to the same system has positively influenced the performance.

From this part of the Simulation Results section onward, the modulation classification simulation results using CNN networks on the dataset generated in Matlab will be discussed. In this part of the study, the classification performance of Scenario 1 and Scenario 7, mentioned in Table 2, has been measured, and the results are presented.

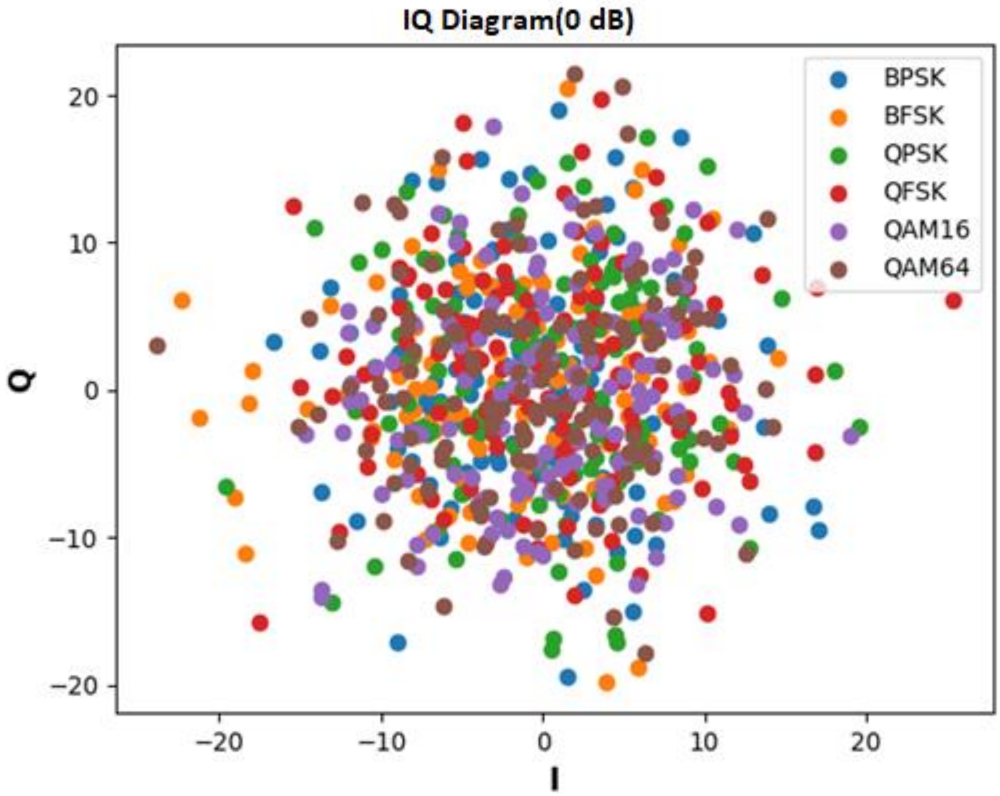

Figure 10. IQ diagram of the data set used at 0 dB (Generated in Matlab).

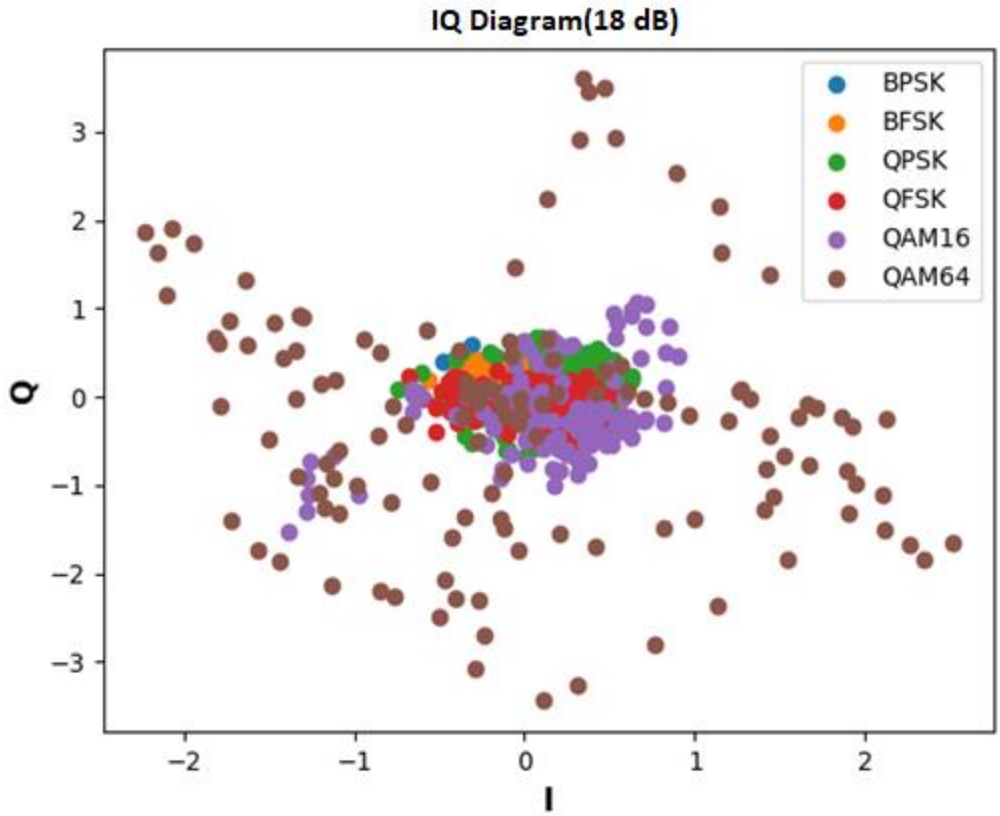

Figure 11. IQ diagram of the data set used at 18 dB (Generated in Matlab).

Figures 9 and 10 display constellation diagrams for the sample space at different SNR values. It is observed that the increase in SNR makes different modulation classes more distinct, similar to the RadioML.2016a dataset. Even at the 18 dB SNR level in Figure 10, most modulation types either overlap or are very close in proximity.

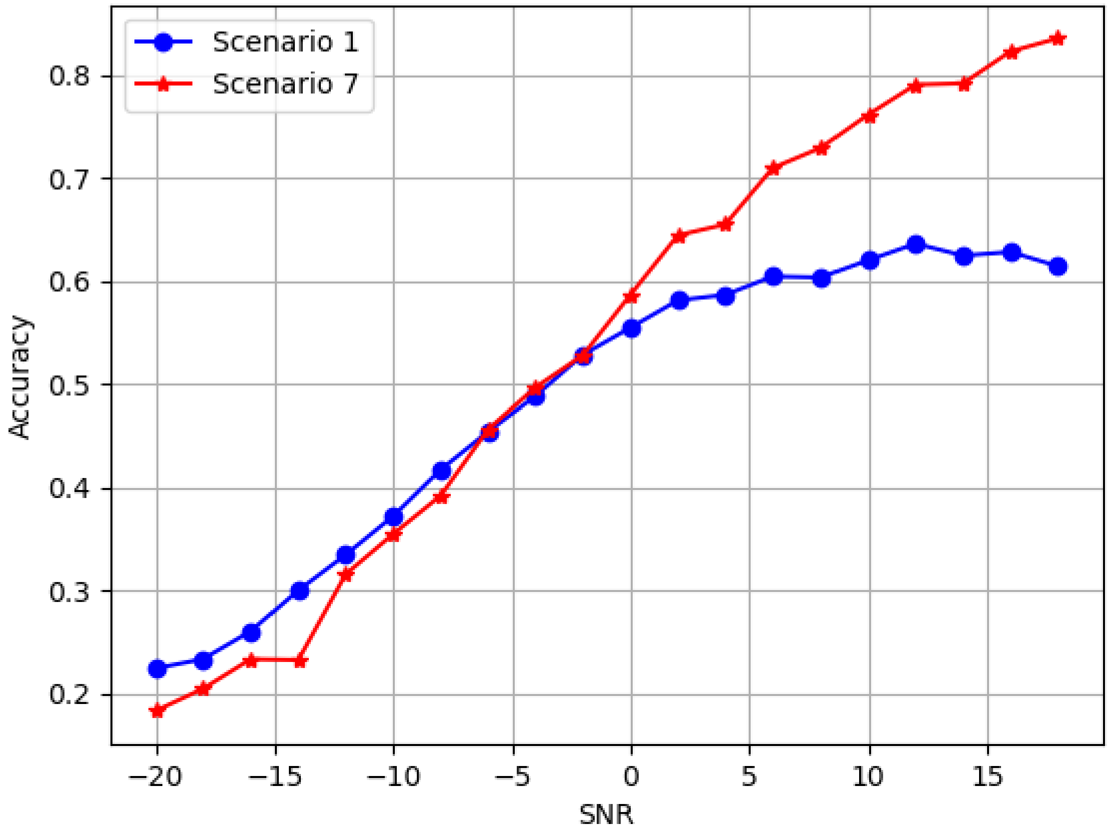

Figure 12. Performance analysis of the dataset generated in matlab in different scenarios.

The classification accuracy of Scenario 1 and 7 in the Matlab-generated dataset is compared in Figure 12. Similar to the initial part of the study, the single-layer CNN structure achieved a success level of only 60%, indicating that it is not suitable for this problem. In Scenario 7, however, the performance has exceeded 80%. The impact of the increase in the number of layers and filters on solving the problem is clearly observed.

6. CONCLUSION

Within the scope of the study, the performance of seven CNN architectures with different numbers of filters and layers was examined for the modulation classification problem on the RadioML.2016a dataset. The impact of the SNR level on distinguishing modulations in the problem space and the difficulty of the problem even at the highest SNR level in the dataset were discussed. Complexity matrices, which clearly present the impact of the SNR level and modulations that are easily confused or easily distinguishable, were provided in the simulation results of the seven scenarios used in the next step. The complexity matrix, indicating the bit-error rate of a modulation, does not show how robust the used network structure is when working with unseen data, i.e., whether it produces similar results to the training results. Therefore, in the next step, accuracy graphs of the scenarios were examined. When the complexity matrices were examined, it was clearly seen that a single-layer CNN structure was not successful in solving the problem even when the number of filters was increased and at high SNR levels. An improvement in performance in the two-layer CNN structure compared to the single-layer CNN was clearly observed. It was observed that the use of a low number of filters in two-layer and three-layer CNN structures produced similar results, and the three-layer structure was slightly more successful. Then, the effects of the number of filters in the two-layer CNN structure and the addition of a fully connected layer after two CNN layers were examined. As expected, an increase in the number of filters improved performance. The structure with a fully connected layer showed the highest performance in classifying 64-QAM. This indicates that the fully connected layer is successful in processing the data received from the CNN layer for the studied data. An increase in the SNR level and the number of filters also positively affected the performance in these structures. When the complexity matrices were examined, it was observed that the AM-DSB, BPSK, CPFSK, GFSK, and PAM4 modulation types were classified with a success rate of 90% at high SNR levels, while the CNN was not as successful in distinguishing 64-QAM and QPSK modulations from each other. The reason for this is thought to be that these two modulation types have similar star cluster diagrams due to the influence of noise, and the network model cannot track amplitude and phase changes.

Accuracy graphs for each scenario and information about the average classification performance of the network structure have been provided. While the accuracy of the network in single-layer structures remains around 60%, it reaches 80-90% in two and three-layer structures. Two-layer CNN structures with a high number of filters and structures with an additional fully connected layer have achieved performance above 90%. This indicates that the generalization performance of the network improves with the number of layers and filters when dealing with new data. In the second part of the study, the performance of the CNN architecture on BPSK, BFSK, QPSK, QFSK, 16-QAM, and 64-QAM modulations generated in Matlab was examined within the scope of Scenarios 1 and 7. It was observed that the SNR level makes the problem space more complex or simpler. It was seen that the single-layer CNN had very low classification success in this problem space, and the multi-layer structure was more successful. The performance of the multi-layer structure increased by around 20%. In this dataset, except for QFSK and 16-QAM modulation types, the multi-layer network could not achieve success even at the highest SNR value, reaching almost 80%. The reason for this is thought to be that channel conditions have deteriorated the generated data more than those used in RadioML.2016a. Another reason is that the dataset size used in the second part of the study has decreased by approximately 15% compared to the first one. When considering the fact that deep networks learn better with a large amount of data, increasing the dataset size may also be beneficial.

Within the scope of the study, the AMC process was studied using Convolutional Neural Networks (CNNs). For this purpose, seven and six different modulation types in two separate datasets, each with twenty different SNR values, were worked on. The findings were compared by applying the used datasets to seven different CNN scenarios. As a result of the study, it was observed that using Convolutional Neural Networks for automatic modulation classification achieved high success. It was clearly seen that increasing the number of CNN layers, the number of filters in the layers, and adding an extra fully connected layer before the output layer positively affected the performance.

7. FUTURE RESEARCH

In future studies, AMC is a valuable tool for military applications. It has a usage area in determining potential threats and increasing communication reliability. In future studies, AMC is a method that is open to development in order to be used effectively in these areas. AMC can be used to classify the signals of military technologies such as drones or unmanned aerial vehicles (UAVs). This would be a significant step in detecting and disabling enemy UAVs. Additionally, AMC can be more effective when combined with multiple antenna technologies. These technologies can determine the direction and power of the signal by using multiple antennas. Thus, more accurate classification results can be provided. In conclusion, future studies promise hope for military communication and electronic warfare applications with further development and improvement of AMC technologies.

REFERENCES

[1] S. J. Kim and D. Yoon, "Automatic modulation classification in practical wireless channels, *2016 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2016, pp. 915-917,* doi: 10.1109/ICTC.2016.7763329.

[2] 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, p. 137, 2007.

[3] D. Zhang et al., *Automatic Modulation Classification Based on Deep Learning for Unmanned Aerial Vehicles*, Sensors, **18**(3), p. 924 (2018).

[4] B. G. Mobasseri, *Digital modulation classification using constellation shape*, Signal Processing, **80**, 251-277 (2000).

[5] M. D. Wong and A. K. Nandi, *Semi-blind algorithms for automatic classification of digital modulation schemes*, Digital Signal Processing, **18**, 209-227 (2008).

[6] A. Güner, Ö. F. Alçin, and A. Şengür, "*Automatic digital modulation classification using extreme learning machine with local binary pattern histogram features*," Measurement, vol. **145**, pp. 214-225, 2019.

[7] W. Li, Z. Dou, C. Wang and Y. Zhang, "Signal Modulation Classification Based on Deep Belief Network," *2019 IEEE Globecom Workshops (GC Wkshps)*, Waikoloa, HI, USA, 2019, pp. 1-6

[8] S. Hu, Y. Pei, P. P. Liang and Y. -C. Liang, "Robust Modulation Classification under Uncertain Noise Condition Using Recurrent Neural Network," *2018 IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates, 2018, pp. 1-7

[9] M. Zhang, Y. Zeng, Z. Han and Y. Gong, "Automatic Modulation Recognition Using Deep Learning Architectures," *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Kalamata, Greece, 2018, pp. 1-5

[10] O'SHEA, Timothy J.; CORGAN, Johnathan; CLANCY, T. Charles. Convolutional radio modulation recognition networks. In: *Engineering Applications of Neural Networks: 17th International Conference, EANN 2016, Aberdeen, UK, September 2-5, 2016, Proceedings 17*. Springer International Publishing, 2016. p. 213-226.

[11] B. Kim, J. Kim, H. Chae, D. Yoon and J. W. Choi, "Deep neural network-based automatic modulation classification technique," *2016 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2016*, pp. 579-582.

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

[13] S. Ramjee, S. Ju, D. Yang, X. Liu, A. E. Gamal, and Y. C. Eldar, "*Fast Deep Learning for Automatic Modulation Classification*," arXiv:1901.05850 [cs, eess, stat], Jan. 2019.

[14] X. Liu, D. Yang and A. E. Gamal, "Deep neural network architectures for modulation classification," *2017 51st Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 2017*, pp. 915-919.

[15] BAGGA, Jaspal; TRIPATHI, Neeta. Automatic modulation classification using statistical features in fading environment. *International Journal of Advanced Research in electrical, electronics and instrumentation engineering*, 2013, 2.8: 3701-3709.

[16] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders and S. Pollin, "*Deep Learning Models for Wireless Signal Classification With Distributed Low-Cost Spectrum Sensors*," in IEEE Transactions on Cognitive Communications and Networking, vol. **4**, no. 3, pp. 433-445, Sept. 2018.

[17] D. Hong, Z. Zhang and X. Xu, "Automatic modulation classification using recurrent neural networks," *2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 2017*, pp. 695-700.

[18] WU, Hao, et al. VHF radio signal modulation classification based on convolution neural networks. In: *Matec Web of Conferences*. EDP Sciences, 2018. p. 03032.

[19] Y. Wu, X. Li and J. Fang, "A Deep Learning Approach for Modulation Recognition via Exploiting Temporal Correlations," *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Kalamata, Greece, 2018*, pp. 1-5.

[20] ZHANG, Duona, et al. Automatic modulation classification based on deep learning for unmanned aerial vehicles. *Sensors*, 2018, 18.3: 924.

[21] K. Ma, Y. Zhou and J. Chen, "CNN-Based Automatic Modulation Recognition of Wireless Signal,"*2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE),Dalian, China, 2020*, pp. 654-659.

[22] ZHOU, Siyang, et al. A robust modulation classification method using convolutional neural networks. *EURASIP Journal on Advances in Signal Processing*, 2019, 2019: 1-15.

[23] LIU, Fugang; ZHANG, Ziwei; ZHOU, Ruolin. Automatic modulation recognition based on CNN and GRU. *Tsinghua Science and Technology*, 2021, 27.2: 422-431.

[24] MA, Pengfei, et al. A Robust Constellation Diagram Representation for Communication Signal and Automatic Modulation Classification. *Electronics*, 2023, 12.4: 920.

[25] SHARMA, Dr K.; MISHRA, A.; SAXENA, Rajiv. Analog & digital modulation techniques: an overview. *International Journal of Computing Science and Communication Technologies*, 2010, 3.1: 2007.

[26] CHANNI, Harpreet Kaur. A comparative study of various digital modulation techniques. *International Journal in IT and Engineering*, 2016, 4.03.

[27] OETTING, John. A comparison of modulation techniques for digital radio. *IEEE Transactions on communications*, 1979, 27.12: 1752-1762.

[28] FRENZEL, Lou. Understanding modern digital modulation techniques. *Electron. Des. Technol. Commun*, 2012.

[29] SALZ, J. Communications efficiency of certain digital modulation systems. *IEEE Transactions on Communication Technology*, 1970, 18.2: 97-102.

[30] D'ANDREA, Aldo N.; MENGALI, Umberto; MORELLI, Michele. Symbol timing estimation with CPM modulation. *IEEE Transactions on communications*, 1996, 44.10: 1362-1372.

[31] S. Haykin, M. Moher, "*Introduction to analog and digital communications*," Wiley, 2007.

[32] AULIN, T.; RYDBECK, N.; SUNDBERG, C. E. Continuous phase modulation—Part II: partial response signalling', ibid., 1981. *COM-29*, 210-225.

[33] F. Wang, O. A. Dobre, C. Chan, J. Zhang, "Fold-based Kolmogorov–Smirnov Modulation Classifier," *IEEE Signal Process*, Lett. 2016, 23, 1003–1007.

[34] D. Zhu, V.J. Mathews, D.H. Detienne, "A Likelihood-Based Algorithm for Blind Identification of QAM and PSK Signals," IEEE Trans. Wirel. Commun., 2018, 17, 3417–3430

[35] KHARBECH, Sofiane, et al. On classifiers for blind feature-based automatic modulation classification over multiple-input–multiple-output channels. *IET Communications*, 2016, 10.7: 790-795.

[36] ORLIC, Vladimir D.; DUKIC, Miroslav L. Automatic modulation classification algorithm using higher-order cumulants under real-world channel conditions. *IEEE Communications Letters*, 2009, 13.12: 917-919.

[37] WU, Hsiao-Chun; SAQUIB, Mohammad; YUN, Zhifeng. Novel automatic modulation classification using cumulant features for communications via multipath channels. *IEEE Transactions on Wireless Communications*, 2008, 7.8: 3098-3105.

[38] SHAH, Ali H.; MIRY, Abbas Hussien; SALMAN, Tariq M. Automatic modulation classification based deep learning with mixed feature. *International Journal of Electrical & Computer Engineering (2088-8708)*, 2023, 13.2.

[39] GUO, Qiang; YU, Xin; RUAN, Guoqing. LPI radar waveform recognition based on deep convolutional neural network transfer learning. *Symmetry*, 2019, 11.4: 540.

[40] ALI, Ahmed K.; ERÇELEBI, Ergun. Algorithm for automatic recognition of PSK and QAM with unique classifier based on features and threshold levels. *ISA transactions*, 2020, 102: 173-192.

[41] S.B. Sadkhan, “A proposed digital modulated signal identification based on pattern recognition,” *In Proceedings of the 2010 7th International Multi- Conference on Systems, Signals and Devices, Amman, Jordan*, 27–30 June 2010; pp. 1–6.

[42] DULEK, Berkan. Online hybrid likelihood based modulation classification using multiple sensors. *IEEE Transactions on Wireless Communications*, 2017, 16.8: 4984-5000.

[43] XIAO, Wenshi; LUO, Zhongqiang; HU, Qian. A review of research on signal modulation recognition based on deep learning. *Electronics*, 2022, 11.17: 2764.

[44] J. McCarthy, M. L. Minsky, N. Rochester and C. E.Shannon, “*A proposal for the dartmouth summer research project on artificial intelligence*, August 31, 1955.,” AI magazine, **27**(4), 12-12. 2006.

[45] J. H. Fetzer, "*What is Artificial Intelligence*?," Artificial Intelligence: Its Scope and Limits,” Springer, Dordrecht, 1990. 3-27

[46] MITCHELL, Tom M.; MITCHELL, Tom M. *Machine learning*. New York: McGraw-hill, 1997.

[47] KOTSIANTIS, Sotiris B., et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 2007, 160.1: 3-24.

[48] D. Bhamare, T. Salman, M. Samaka, A. Erbad, R. Jain, “Feasibility of supervised machine learning for cloud security,” *In: 2016 International Conference on Information Science and Security (ICISS)*, pp. 1–5, 2016.

[49] M. Xie, J. Hu, J. Slay, “Evaluating host-based anomaly detection systems: application of the one-class SVM algorithm to ADFA-LD*,” 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD),* pp. 978–982, 2014.

[50] KATO, Keisuke; KLYUEV, Vitaly. An intelligent ddos attack detection system using packet analysis and support vector machine. *IJICR*, 2014, 14.5: 3.

[51] A.R. Yusof, N.I. Udzir, A. Selamat, “An evaluation on KNN-SVM algorithm for detection and prediction of DDoS attack,” *In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 95–102, 2016.

[52] M. Feily, A. Shahrestani, S. Ramadass, “A survey of botnet and botnet detection,” *2009 Third International Conference on Emerging Security Information, Systems and Technologies*, pp. 268–273, 2009.

[53] M. Ben Salem, S. Hershkop, S.J. Stolfo, “*A survey of insider attack detection research,” Insider Attack and Cyber Security*, pp. 69–90. Springer, 2008.

[54] SINAGA, Kristina P.; YANG, Miin-Shen. Unsupervised K-means clustering algorithm. *IEEE access*, 2020, 8: 80716-80727.

[55] F. Badran, M. Yacoub, and S. Thiria, "*Self-organizing maps and unsupervised classification*," Neural networks. Springer, Berlin, Heidelberg, 2005.

[56] R. S. Sutton and A. G. Barto, “*Reinforcement learning: An introduction*,” MIT press, 2018.

[57] KAELBLING, Leslie Pack; LITTMAN, Michael L.; MOORE, Andrew W. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 1996, 4: 237-285.

[58] Y. LeCun, B. Yoshua, and G. Hinton, "*Deep learning*," Nature 521.7553, 2015, 436-444.
[59] SVOZIL, Daniel; KVASNICKA, Vladimir; POSPICHAL, Jiri. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 1997, 39.1: 43-62.
[60] ILONEN, Jarmo; KAMARAINEN, Joni-Kristian; LAMPINEN, Jouni. Differential evolution training algorithm for feed-forward neural networks. *Neural Processing Letters*, 2003, 17: 93-105.
[61] K. O'Shea, and R. Nash, "*An introduction to convolutional neural networks*." arXiv preprint arXiv:1511.08458, 2015.
[62] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, B., T. Chen, T, "*Recurrent neural networks*," Recent advances in convolutional neural networks. Pattern recognition, **77**, 354-377, 2018.
[63] L.R. Medsker, and L. C. Jain, "Recurrent neural networks," *Design and Applications* 5, 2001, 64-67.
[64] T. Mikolov, "Learning longer memory in recurrent neural networks," arXiv preprint arXiv:1412.7753, 2014.
[65] Q. Yao, et al, "A dual-stage attention-based recurrent neural network for time series prediction," arXiv preprint arXiv:1704.02971, 2017.
[66] ZHANG, Jia-Shu; XIAO, Xian-Ci. Predicting chaotic time series using recurrent neural network. *Chinese Physics Letters*, 2000, 17.2: 88.
[67] A. Graves, "*Long short-term memory," Supervised sequence labelling with recurrent neural networks*, 2012, 37-45.
[68] HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. *Neural computation*, 1997, 9.8: 1735-1780.
[69] T. J. O'Shea, C. Johnathan, and T. C. Clancy, "Convolutional radio modulation recognition networks," *International conference on engineering applications of neural networks*, Springer, Cham, 2016.

To Cite This Article: O. Kaya, T. Güçlüoğlu, H. İlhan, *Depth Analysis in Deep Learning-Based Automatic Modulation Classification*, Journal of Aeronautics and Space Technologies **17**(2), 18-45 (2024).

VITAE

Osman Kaya graduated from Sakarya University, Department of Electrical and Electronics Engineering in 2016. He graduated from Yıldız Technical University Electronic Communication Engineering Communication Master Program in 2023. He is currently continuing his PhD education in the Communication program in the Department of Electronic Communication Engineering at the same university.

Tansal Güçlüoğlu received the B.S. degree from Middle East Technical University, Ankara, the M.S. degree from Syracuse University and the Ph.D. degree from Arizona State University. He worked as an R&D Engineer for four years in Ankara and New York. He was a Research Associate and a Teaching Associate at Arizona State University. He has been working as a faculty member since 2006. His research interests include communication theory, wireless communication networks, signal processing, channel coding/decoding and artificial intelligence algorithms.

Hacı İlhan (M'07, SM'19) received the B.Sc. degree in electronics and communication engineering from Yildiz Technical University, Istanbul, Türkiye. The M.Sc. and the Ph.D. degree all in electronics and communication engineering from Istanbul Technical University, Istanbul, Türkiye. From 2001 to 2011 he was a research and teaching assistant in communication department at Istanbul Technical University. Since 2011, he has been with the Department of Electronics and Communication, Yildiz Technical University, Istanbul, Türkiye, where he is a Professor Doctor. His general research interests lie in communications theory and signal processing for communications with particular emphasis on wireless applications. Specific research areas include cooperative communication, MIMO communication techniques, space-time coding, multicarrier communication, network coding, spatial modulation, index modulation, full-duplex communication, UAV communication, NOMA, energy harvesting, SWIPT, vehicular communication, MAC protocols for vehicular ad hoc networks. Dr. Ilhan is a senior IEEE member.