Iraqi Journal of Information and Communications Technology(**IJICT**)
**Vol. 8, Issue 2, August 2025**
www.ijict.edu.iq
ISSN:2222-758X
e-ISSN:2789-7362

# DNN-AIDED CODEBOOK DESIGN WITH MPA DECODING USING DEEP NEURAL NETWORK

**Hawraa A. Jasiem** [1], **Hikmat N. Abdullah** [2]
[1,2] Department of Information and Communication Engineering, College of Information Engineering, Al-Nahrain University, Jadriya, Baghdad, Iraq
Hawraaahmed219@gmail.com[1], hikmat.abdullah@nahrainuniv.edu.iq [2]
Corresponding Author: **Hikmat N. Abdullah**

Received:23/08/2024; Revised:22/12/2024; Accepted:05/02/2025
DOI:**10.31987/ijict.8.2.302**

***Abstract*- Sparse Code Multiple Access (SCMA) is an extremely effective non-orthogonal multiple access technology that enables communication between users who have limited orthogonal resources currently. Traditional SCMA methods use manually designed codebooks, potentially leading to subpar performance owing to inadequate optimization for certain encoders. A Deep Neural Network (DNN) is used to produce a deep learning for SCMA codebook using Stochastic Gradient Descent (SGD). This enables the model to determine the optimal weights and biases for generating precise predictions. The proposed approach surpasses existing techniques in a Rayleigh fading channel due to its reduced Bit Error Rate (BER), enhanced Minimum Euclidean Distance (MED), and diminished complexity compared to prior SCMA frameworks.**

***keywords:* Sparse Code Multiple Access (SCMA), Deep Neural Network (DNN), Message Passing Algorithm (MPA), Codebook design, Deep learning.**

## I. INTRODUCTION

Non-Orthogonal Multiple Access (NOMA) approaches facilitate communication in scenarios when the number of users exceeds the available resources. Developing such methodologies is critical for improving communication with maximum effectiveness and minimal delay. Sparse Code Multiple Access (SCMA) is a very promising solution for NOMA that operates in the code domain [1]. The system utilizes a method of allocating resources sparingly in order to achieve a low Bit Error Rate (BER) using decoders that have minimal complexity, such as the Message Passing Algorithm (MPA) [2]. Reference [3] demonstrates that traditional SCMA formulates the problem as a multi-stage optimization, constructing the codebook and decoder separately. This is the main drawback of using heuristics to create users' codebooks, which ignore decoder characteristics. Using information about the decoder's characteristics to construct codebooks may enhance performance [4]. The effectiveness of a SCMA system depends significantly on designing a codebook.
Therefore, the ideal design of a codebook should incorporate the topic has undergone a thorough examination SCMA is a novel non-orthogonal multiple access technique that combines Code Division Multiple Access (CDMA) with Quadrature Amplitude Modulation (QAM). In this technique, QAM symbols are spread across Orthogonal Frequency Division Multiple Access (OFDMA) tones, and CDMA encoder disperses them to generate a pre-designed complex symbol sequence [5,6]. The constellations were considered in [6,7]. However, creating codebooks manually poses challenges. Since the codewords in a codebook are not orthogonal to one another, they interact with one another and consist of intricate multidimensional structures principles. In the last few years, deep learning techniques have shown remarkable advancements in the domains of computer vision, voice recognition, and natural language processing. Profound impacts of profound learning have garnered

significant attention. Within the context of conventional research in the realm of wireless communication, deep learning has made significant advancements [8].
The rest of this paper is organized as follows. Section II presents the related works. Section III introduced the system model used in this work. The proposed method is presented in Section IV. Section V presents the simulation results. Finally, the conclusion is presented in Section VI.

## II. RELATED WORKS

Deep learning technologies have recently led to substantial improvements in several jobs' performance. Significantly, several studies have verified the feasibility of deep learning.
In [9] addressed the process of formulating standard SCMA as a multi-stage optimization problem, where the codebook and decoder are constructed individually. The authors also addressed the primary limitation of using heuristics in the creation of users' codebooks, which is essential to take into account the features of the decoder. Constructing codebooks based on the decoder's features may improve performance. Machine learning can effectively overcome the limitations of classical SCMA. The use of data and the intrinsic attributes of the decoder improve performance by enabling efficient collection levels. The codebook of D-SCMA utilizes Deep Neural Networks (DNN) to represent both the encoder and the decoder. The presence of Additive White Gaussian Noise (AWGN) significantly improves the BER of SCMA on a channel. Nevertheless, its performance on the Rayleigh fading channel is subpar.
In [10], proposed to expand the MPA into a DNN and optimize the weights assigned to the connections between factors to minimize the impact of cycles. Nevertheless, the DNN-MPA model may have slow convergence because of its inheritance of delayed convergence from unrolling MPA.
The goal of the SCMA principles, described in [11], is to reduce waiting times and traffic congestion while providing significant capacity and connectivity in 5G networks. A substantial augmentation of physical connections and an enhanced level of link quality accomplish this. The publication provides instructions for creating SCMA codebooks and achieving a BER below 0.001. Effectively developing and implementing SCMA decoder code may result in a decrease in Field-Programmable Gate Array (FPGA) resource use, as one can expect.
In [12] presented a new method for constructing an SCMA codebook that optimizes the total transmission rate. This method involves transferring a multi-dimensional constellation set's design through a series of intricate 1-dimensional code word optimizations. Th authors optimize the basic M-order Pulse Amplitude Modulation (M-PAM) to boost the sum rate. Determine the rotational angles between the typical M-PAM constellation and the provided 1-dimensional constellation using a computationally efficient method. The numbers show that the proposed codebook works better than the current codebook. It reduces the BER by 1.3 dB over the AWGN channel and 1.1 dB over the Rayleigh channel.
In [13] examined the generation of SCMA codebooks by the use of Genetic Algorithms (GA) to generate SCMA codebooks. Generally, GA do not require a mother constellation since they directly optimize the codebooks throughout the process. The work displayed the codebooks, which are composed of two sets of antipodal code phrases. It is also explores an optimization strategy to boost SCMA signal efficiency in the AWGN channel by limiting the average energy of codewords. The generated codebooks have a Minimum Euclidean Distance (MED) of 0.87. The improvement is 0.8 dB compared to

the most efficient codebooks. The results of the simulations showed that the proposed codebooks work about the same for both maximum likelihood (ML) and maximal a posteriori MPA detection methods. This research aims to resolve the aforementioned concerns by offering a technique for codebook design that is based on a deep neural network. MPA decodes the codebook the only parameter can learn.

In [14], the authors examined the application of machine learning techniques, specifically DNNs, in the design and optimization of communication systems. It shows that data-driven design, which doesn't use explicit channel modeling, often comes up with better answers to problems that are hard to solve computationally. The authors also showed that end-to-end training of a DNN with a large amount of channel data can provide significant system-level enhancements compared to conventional model-based methods. The success of machine learning applications depends on selecting a suitable neural network design that aligns with the issue's inherent structure. The work suggested that big benefits can be gained by not using explicit channel modeling, designing the entire system from start to finish, and solving optimization problems using data.

In [15] focused on two main ways to use deep learning in wireless communications: deep learning-based architectural design, which changes the way traditional model-based block design works, and deep learning-based algorithm design, which uses standard methods for 5G and beyond. The authors also identified unresolved issues and potential research avenues, emphasizing the interaction between deep learning and wireless communications. The goal was to inspire innovative ideas and contribute to intelligent wireless communications.

## III. SYSTEM MODEL

### A. *SCMA System Model*

This work analyzes $J$ distinct data streams that are combined into $K$ separate resources, such as subcarriers in OFDMA or Multi Input Multi Output (MIMO) spatial layers where a single signal stream is sent. The process of multiplexing divides and distributes information across numerous resources. The multiplexing process distributing signal streams allows for sparse resource allocation. Such that only a portion of the signals merge into one asset. This article assumes that $K$ is less than $J$, where $K$ and $J$ are variables. The number of data streams outweighs the number of orthogonal entities. It is inevitable that resources will be allocated, including non-orthogonal resources; next, it can depict the received signal as follows [9]:

$$y = \sum_{j=1}^{J} \operatorname{diag}(h_j)\, x_j + n = \sum_{j=1}^{J} \operatorname{diag}(h_j)\, f_j(c_j) + n \tag{1}$$

The vector $(h_{1j}, \ldots, h_{Kj})^T$ represents $h_j$ in this instance, is the channel vector connecting signal stream $j$ and resources. The notation $\operatorname{diag}(h_j)$ indicates a matrix that has been diagonalized. The vector $x_j$ is composed of the elements $(x_{1j}, \ldots, x_{Kj})^T$. In SCMA represents the signal's codeword. Variable $j, c_j$ represents the input data symbol, while variable $n$ represents the additive white noise, the noise is Gaussian, with a mean of zero and a variation of $\sigma^2$. SCMA represents the encoder (codebook) as $f(c)$. The term $f_{kj}(c_j)$ represent complex constellation extracting resource $k$ from signal $j$ of input data symbol $\mathrm{c_j}$. Fig. 1 shows an example of a factor graph that includes 6-signal streams (users) and several 4-resourses.

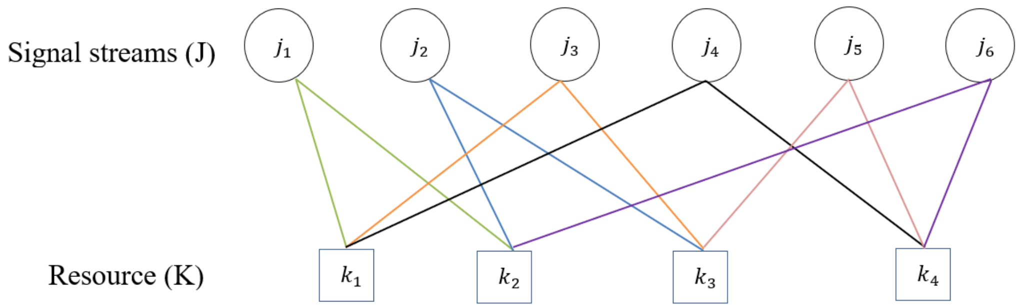

Figure 1: Factor graph of SCMA with $K = 4, and J = 6$.

## *B. MPA Decoding for SCMA*

Given the received signal $y$ in Eq. (1), the resource nodes and user nodes exchange messages in the form of logarithms repeatedly. The probabilities are assigned to the edges in the factor graph. The symbol $\zeta_j$ signifies that the resource node $k$ links the user nodes, while $\xi_k$ represents the user nodes $j$ linked to the resource node. For each given $i$, the text discusses iteration, and the meaning of $l^i_{rk\to uj}(c_j)$ is to move the resource node $k$ to user node $l^i_{uj\to rk}(c_j)$ according to the following equations adopted from [10]:

- The messages from recourse nodes to user nodes, i.e., $j \in \xi_k$

$$l^i_{rk\to uj}(c_j) = \ln\left(\sum_{\substack{\tilde{c}\\ \tilde{c}_j = c_j}} \exp\left(M_k(\tilde{c}) + \sum_{\tilde{j}\in\xi_{k\setminus j}} l^{i-1}_{u\tilde{j}\to rk}(\tilde{c}_{\tilde{j}})\right)\right) \quad (2)$$

$$M_k(\tilde{c}) = -\frac{1}{2\sigma^2}\left\|y_k - \sum_{p\in\xi_k} h_p[k]\,\tilde{c}_p[k]\right\|^2 \quad ()$$

  Where $h_p[i]$ represents the $i$-th element of $h_p$, and $\tilde{c}$ represents every possible combination of user codewords while the $j$-th user is transmitting $c_j$.

- The message from user nodes to resource nodes, i.e., $k \in \zeta_j$, is given by:

$$l^i_{uj\to rk}(c_j) = \sum_{\tilde{k}\in\zeta_j\setminus k} l^{i-1}_{r\tilde{k}\to uj}(c_j) \quad (3)$$

  Once the stopping requirement is satisfied at any iteration $i$, use the following formula to determine the final log probabilities of symbols for user $j$:

$$l^i_{uj}(c_j) = \sum_{\tilde{k}\in\zeta_j} l^{i^*}_{r\tilde{k}\to uj}(c_j) \quad (4)$$

- $A \in (-\infty, 0]^{M\times J}$ express the output log probabilities for all $J$ users and $M$ symbols as follows:

$$\mathrm{MPA}(y, C) \quad (5)$$

## IV. PROPOSED METHOD

This section outlines the process of mapping signal streams to resources via codebook design and decoding the incoming signal using MPA. Next section introduces the training session, which is the proposed approach's architecture of this study (refer to Fig. 2).

### *A. Constructing a Codebook using Deep Neural Network (DNN)*

The DNN-SCMA technique utilizes DNN to determine the mapping between the input data and the constellation plane of the specified resource. The mapping of the M-arry symbol $c_j$ from stream $j$ to the complicated plane representing a single resource $k$ is denoted $f_{kj}(c)$. In order to achieve this objective many fundamental DNN unit are positioned between streams and resources such these DNN units may as seen in Fig. 2. This study suggested the number of hidden layers in basic DNN - the value of $L$ is set to 6, indicating that each hidden layer in the basic model consists of 6 units. Any layer between the input and output layers of a neural network are known as a hidden layer. The processing in the network mostly takes place in these layers, as the network acquires the ability to associate the input data (transmit symbols) with the output (encoded codes or codewords). The DNN consists of 32 hidden nodes, a neuron in a DNN gets input from the

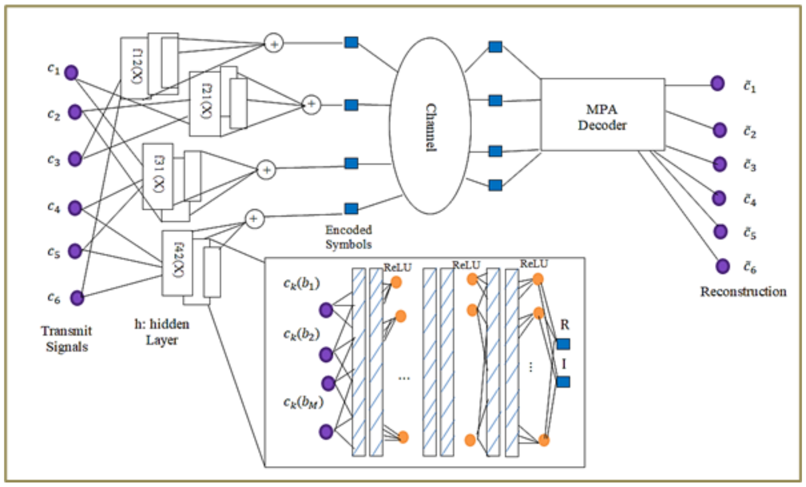

Figure 2: The architecture of deep neural network based -SCMA.

previous layer, goes through a linear transformation using weights and biases, and then goes through an activation function (like ReLU) to add non-linearity. The choice of 32 neurons per layer demonstrates the use of a small network. The size selection is most likely based on a trade-off between the network's ability to represent intricate interactions and the need to maintain reasonable computing costs.
These DNN units and the function $f_{kj}(c)$ may be regarded as a generator of codewords. The codewords were produced, each one corresponding to a certain value. Let $f_{kj}(c;\theta_f)$ be the function that maps the resource k to the stream. When

the parametric representation of a DNN is denoted as $\theta_f$, the variable $\theta_f$ represents the weights and biases of the encoder. Subsequently, the received for the signal at resource $k$ of the receiver may be represented as:

$$y_k = \sum_{j=1}^{J} h_{kj}\, f_{kj}(c_j; \theta_f) + n_{kj} \tag{6}$$

As seen in Fig. 2, the DNN encoder module $f_{kj}(c_j; \theta_f)$ has M input nodes for each M-array symbol $c_j$.

## *B. The Process of Training*

In this work, the suggested DNN architecture should be trained to ensure that the encoders, $f_{kj}(c_j; \theta_f)$, accurately reproduce the original data. Therefore, this work expresses the comprehensive loss function of the DNN as follows:

$$L(c, \tilde{c}; \theta_f) = \left\| c - \mathrm{MPA}\big(hf(c; \theta_f) + n\big) \right\|^2 \tag{7}$$

Here, $c$ represents the original input data symbol. The equation of $\tilde{c} = \mathrm{MPA}\big(hf(c; \theta_f) + n\big)$, represents the data the MPA decoder reconstructs. The symbols $\theta_f$, denote the weights and biases of the DNN encoder. The channel vector, denoted as $h$, represents the channel's characteristics. The noise simulates the conditions of the channel environment. Following MPA decoding, the goal is to minimize the squared error between the original symbols $c$ and the reconstructed symbols $\tilde{c}$. Referring to the loss function in Eq. (4), update the weights and biases of the DNN encoder using a Stochastic Gradient Descent (SGD) technique as follows:

$$\theta_f \;:=\; \theta_f - \alpha \nabla_{\theta_f} L(c, \tilde{c}; \theta_f) \tag{8}$$

The learning rate $\alpha$, denoted as, affects the magnitude of the update step. Additionally, there are other pre-made updating algorithms that use the SGD approach, such as adaptive moment estimation. The DNN parameters. It has value Note that you can construct the encoder independently. In a flexible manner for an arbitrary number of signal streams or unlike earlier SCMA designs that included specific resources, it must be manually redesigned, taking into account the system's specifications.

Fig. 2 provides a visual depiction the input data symbols $c_1, c_2, \ldots, c_6$ represent the original data streams that need transmission without any changes. In Fig. 2, the functions $f_{12}, f_{21}, f_{31}$, and $f_{42}$ correspond to the mapping of codewords for individual users and resources in the SCMA system. These functions determine the process of encoding each user's data for transmission across the allocated resources.

DNNs transform input data symbols into complex constellation points, or codewords, which are pre-established mappings connecting every input sign to a corresponding output codeword. In traditional SCMA, humans manually design codebooks, but DNNs receive instructions to construct these independently. A conventional DNN comprises multiple layers, including an input layer that acquires signals, hidden layers that analyze the input signals, and an output layer that generates encoded symbols corresponding to the elements of the codebooks. These layers hold all the complicated connections between the symbols that are sent and the best codewords for them, making sure that the data is sent quickly and correctly [9].

This work encodes the symbols for transmission across the channel. It is necessary to teach the DNN using supervised learning methods, aiming to minimize the BER at the output after decoding. What the DNN does during the training phase is change the weights and biases of its neurons to reduce the loss function, which is usually Mean Squared Error (MSE) or cross-entropy. The DNN encoder produces a series of encoded symbols that represent the input data in an ideal manner for efficient transmission via the communication channel. The physical channel then receives the signals.

The MPA Decoder employs an iterative approach to decode the incoming signals and retrieve the broadcast symbols. Particularly in a DNN or any other method, the decoding process often handles the real and imaginary components of the received signals either independently or in conjunction to restore the original supplied data. The decoder may use both real components $R(y)$ and the imaginary part $I(y)$ to comprehensively analyze the received signal and precisely estimate the sent data. The $c_k(b)$ variable denotes the value of the reconstructed data bit or symbol $(b)$ for resource $(k)$. It is a crucial component of the decoding process, as it demonstrates the system's capacity to retrieve the original sent data from the received, distorted signal.

See the iterative nature of MPA by using arrows to illustrate feedback loops, which indicate that the decoder improves its estimates of the transmitted symbols over several iterations. In contrast to the DNN decoder, the MPA decoder employs fewer layers of neural networks. Instead, it uses probabilistic algorithms to determine the most likely symbols in a sentence. The MPA decoder generates the final reconstructed data symbols $\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_6$. The symbols should closely mirror the real sent signals, ideally with minimal deviations.

## V. SIMULATION RESULTS

In this study, the proposed method was implemented using the Python programming language. The simulation employs the codebook provided in [12]. This work also employed the SCMA system with an overloading factor of 1.5. The configuration of the system was set and the values assigned are as follows: $J = 6, K = 4, N = 2, M = 4$, and $d_f = 3$. The TensorFlow [16] framework was used in this work. TensorFlow is a highly popular and powerful open-source framework widely used in deep learning applications. It is particularly well-suited for building and training DNNs for a variety of tasks, including computer vision, natural language processing, and codebook generation in communication systems like SCMA, developed by Google. Due to its encapsulation and user-friendly nature.

The execution of the proposed neural network's based on the following configuration. The total number of iterations configured the MPA decoder to use three iterations. During training and testing, the $E_b/N_0$ value set to 6 dB. For the Rayleigh fading channel, $E_b = N_0$ ranges from 0-15 dB for AWGN and from 0-15 dB. Furthermore, the learning rate $\eta$ of the SGD method was set to 0.0001, to ensure sufficient smallness for the loss function to converge. This work used a total of 300,000 training batches, each containing 400 input data sequences. In order to set the starting values for weights and biases, the Xavier method was also used.

### A. BER

Previous studies referenced in [11], [12], and [13]. In Fig. 3 , every resource has a "codebook," which functions as a reference for categorizing or organizing data points. Certain users (e.g., User 3) may be present in both resources, signifying overlap or joint accountability for certain data clusters. Other users, such as User 1 in Resource 1 and User 2 in Resource 2, are restricted to a single resource. These plots could be useful for allocating resources, finding clusters in machine learning, or checking the performance of systems with multiple users where resources are grouped or given to different users. Fig. 4 and Fig. 5 show how BER works on the additive white AWGN channel and the Rayleigh fading channel. Fig. 4

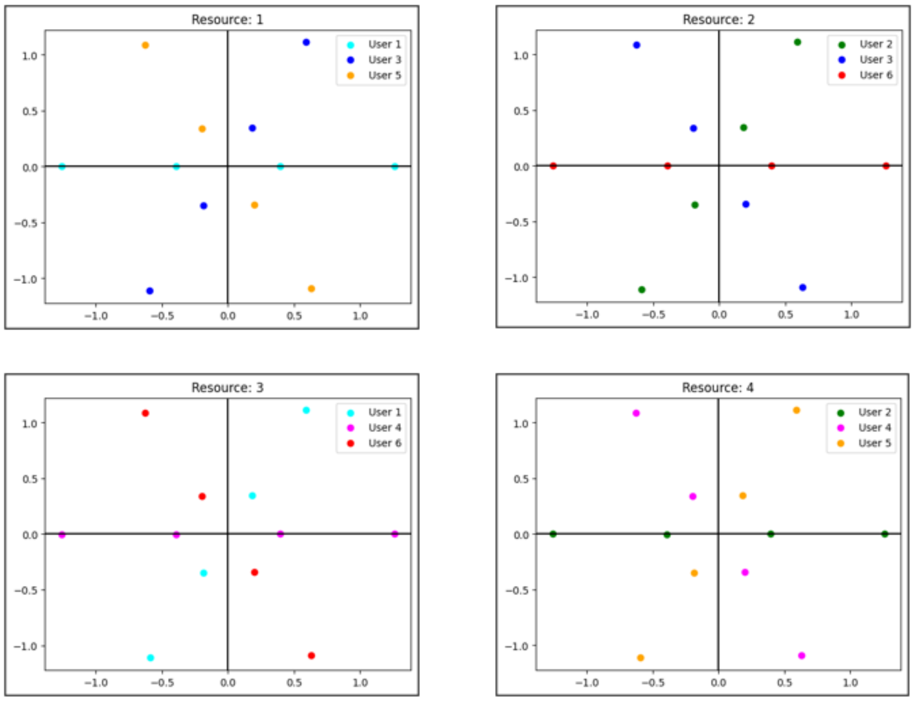

Figure 3: The codebook adaptation based on the training data and the optimization process.

demonstrates that the proposed method achieves superior performance compared to the other approaches while operating on an AWGN channel. To determine the gain, we calculated the difference in Eb/N0 values at each approach corresponds to a specific bit error rate (BER) of $10^{-4}$ between the suggested approach and each of the other methods regarding this situation: The GA technique shows an increase of roughly 0.33 decibels. The capacity-based codebook technique demonstrates an improvement of about 0.22 dB. The Altera codebook approach demonstrates a gain of approximately 0.29 decibels. The proposed approach exhibits an increase of roughly 1 dB. The suggested technique outperforms the GA method by 0.33 dB, Altera codebook method by 0.29 dB, and capacity-based codebook method by 0.22 dB. Therefore, in comparison to the GA approach, the suggested method demonstrates superior performance by significantly reducing the necessary signal power.

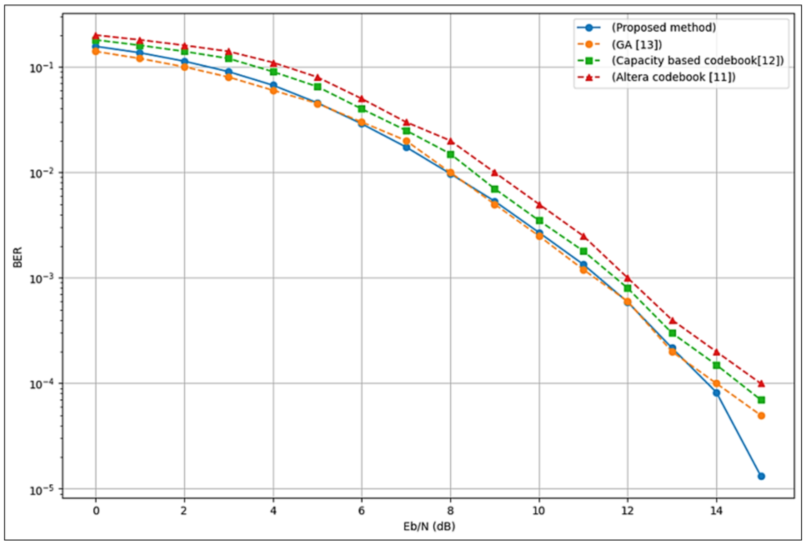

Figure 4: BER Performance comparison on AWGN channel.

Fig. 5 shows a similar pattern on a Rayleigh fading channel. The approach demonstrates superior performance compared to the previously described approaches on the manually designed SCMA codebook. In contrast to Codebook learning techniques, such as Altera codebook, exhibit poor performance when used on the Rayleigh fading channel. To summarize, the suggested technique is superior in this situation because it significantly reduces the amount of signal power required compared to other methods.

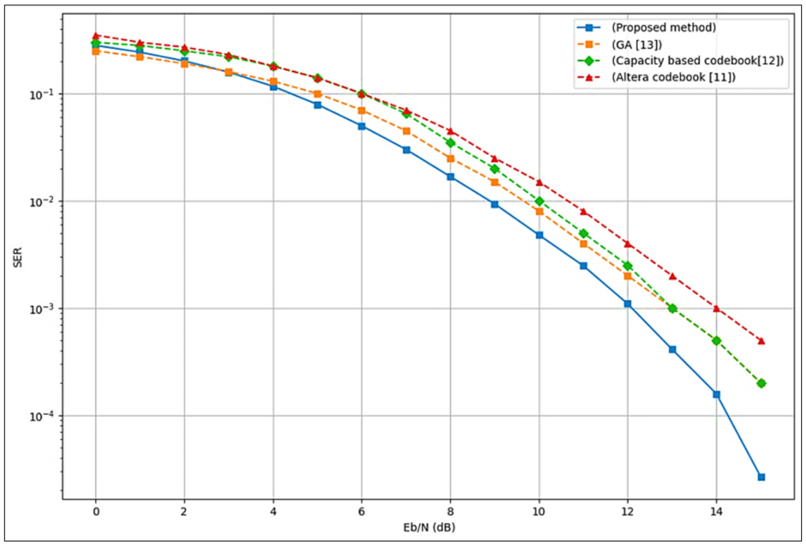

Figure 5: BER Performance comparison on Rayleigh Fading channel.

### *B. Minimum Euclidean Distance (MED)*

Table I presents the MED values of the suggested technique. The value of the proposed method obtained after training the codebook extracted from reference [12] using a DNN. Evaluate the value in relation to other approaches given the specific constraints of $K = 4, N = 2$, and $J = 6$. The tabulated numbers in the table clearly indicate that the suggested method codebook outperforms other methods in terms of MED performance, as it achieves the highest MED value.

TABLE I
A comparative analysis of SCMA codebook approaches.

| **Method** | **MED** |
|---|---|
| Altera codebook [11] | 0.46 |
| Capacity based codebook [12] | 0.67 |
| Genetic algorithm [13] | 0.87 |
| Proposed Method | 2.00 |

### *C. Complexity*

The examine computational difficulty of various ways addressed in this subsection. A neural network's time complexity refers to the computational effort required to process input as it moves through the network, including operations such as multiplication and addition. The time complexity of a forward pass in a neural network refers to the computing efficiency needed to analyze input data and transmit it through the network to produce an output. Table II provide a comparison of the computational complexity of the proposed technique with the methods described in references [11], [12], and [13]. The quantity of layers and neurons in this DNN dictates its computational complexity. The complexity for each forward run through the network is about $O(L \times n \times m) = O(6 \times 32 \times 6)$, $O = (1152)$ operations where $L$ is the number of layers, $n$ is the number of neurons per layer, and $m$ is the number of input characteristics. With $L = 6, n = 32$ neurons per layer, and $m = 6$ real-time encoding can handle this task, especially with the use of specialized technology like GPUs. According to [11], the equation $O(N_{iter}F \times M^{d_f} + V \times M^{d_v})$ represents the complexity of the Altera codebook this

TABLE II
The computational complexity of the four techniques

| **Method** | **Time Complexity** | **Number of Operations** |
|---|---|---|
| Altera codebook [11] | $O(N_{\text{iter}}F \times M^{d_f} + V \times M^{d_v})$ | $O(1408)$ |
| Capacity based codebook [12] | $O(M^{d_f} \times J \times N_{\text{iter}})$ | $O(1536)$ |
| Genetic algorithm [13] | $O(M^{2J}PG)$ | $O$(30,777,216) |
| Proposed Method | $O(L \times n \times m)$ | $O(1152)$ |

equation takes into account the important elements that affect the complexity of MPA decoders. The variable nodes $(V)$ indicate the number of users. $F$: The number of function nodes is equivalent to four resources. The modulation order $(M)$ signifies the quantity of codewords assigned to each user. Given these conditions, the allocated numerical values

are $V = 6, F = 4$, and $M = 4$. The variable $d_f$ represents the distribution of resources among various users, precisely equivalent to 3. Assign the value of 2 to the variable $d_v$. The expected number of iterations for the MPA decoder is denoted by $N_{iter} = 4, O(4 \times 4 \times 4^3 + 6 \times 4^2) = O(1408)$ operations.
In [12] the complexity of the MPA decoder is represented as $O(M^{d_f} \times J \times N_{iter}) = O(4^3 \times 6 \times 4) = O(1536)$ operations. The GA is denoted as [13]. The given complexity is represented as $O(M^{2J}PG)$ , where $M$ is the modulation order equal to 4, $J$ is the number of users, $p = 60$ is the number of individuals in the population, and $G = 30$ is the number of generations needed by the method, $O(4^{2*6} \times 60 \times 30) = O(30, 777, 216)$ operations.

## D. Accuracy

Fig. 6 illustrates the "accuracy over iterations" inside a deep learning framework, often occurring during the model's training phase. As the number of iterations increases the precision improves progressively. The accuracy begins at roughly 0.88 and gradually increases to about 0.96 after 200 repetitions. Minor variations in accuracy occur between iterations, which is customary during training since the model persistently modifies weights to reduce loss. Nonetheless, the overarching tendency is upward, indicating that the model is acquiring knowledge and improving its performance In the later section of the graph (after around 150 repetitions), the accuracy stabilizes at approximately 0.96. This suggests that the model is attaining its peak performance, and more training may not much enhance accuracy.

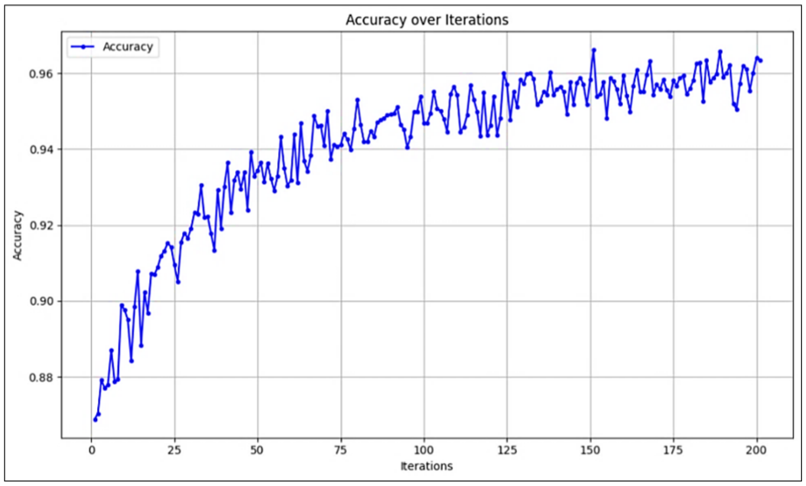

Figure 6: Accuracy over 200 epoch (iterations).

Fig 7. Initially (iteration 0), the loss is high, exceeding 0.30, indicating that the model's predictions were substantially far from the actual outcomes. Noted a substantial reduction in loss within the first 50 rounds. This indicates that the model quickly adapted its weights, improving its predictions in the initial training phases. Despite the overall decline in loss, variations occur. These are typical as the model experiments with various modifications to enhance its performance. The little fluctuations illustrate the stochastic characteristics of gradient descent used in the training process. After around 150 cycles, the loss begins to settle and diminishes more gradually, with values oscillating at 0.10. This indicates that the

model is nearing its ideal condition and is experiencing a deceleration in learning, possibly arriving at a stage when further improvements are negligible.

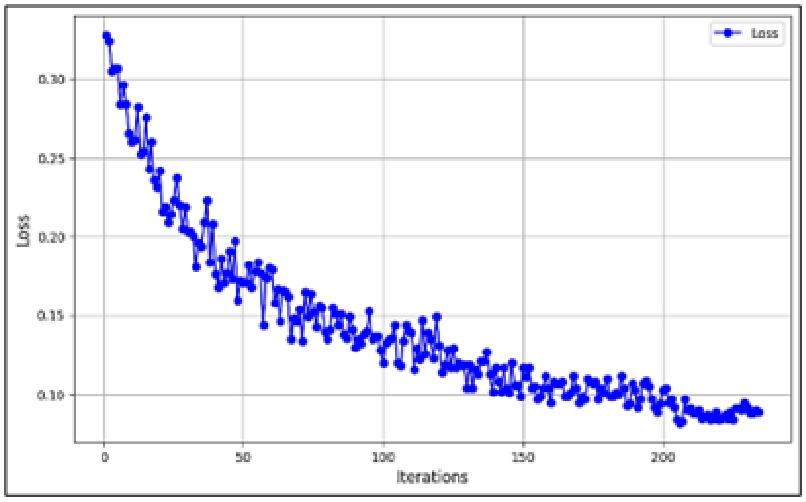

Figure 7: Loss over 200 epoch.

## VI. CONCLUSION

A proposed approach known as DNN-SCMA has been introduced. Stochastic Gradient Descent can autonomously construct a capable codebook using deep learning techniques. This results in an efficient codebook designed for sparse multidimensional overlay inputs. Simulations indicate that the proposed system outperforms traditional methods, systems should be evaluated based on both BER and computational complexity. This approach overcomes the limitations of previous methodologies, demonstrating exceptional performance in both Additive AWGN and Rayleigh fading channels. Additionally, it exceeds the maximum probability decoding performance of a conventional SCMA codebook. The method enhances conventional SCMA while maintaining its runtime complexity. Codebook development is conducted independently, without human intervention.

**FUNDING**

None.

**ACKNOWLEDGEMENT**

The author would like to thank the reviewers for their valuable contribution in the publication of this paper.

**CONFLICTS OF INTEREST**

The author declares no conflict of interest.

## REFERENCES

[1] H. Nikopour and H. Baligh, “Sparse code multiple access,” in 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). IEEE, 2013, pp. 332–336.

[2] R. Hoshyar, F. P. Wathan, and R. Tafazolli, “Novel Low-Density Signature for Synchronous CDMA Systems Over AWGN Channel,” IEEE Trans. Signal Process., vol. 56, no. 4, pp. 1616–1626, Apr. 2008, doi: 10.1109/TSP.2007.909320.

[3] M. Taherzadeh, H. Nikopour, A. Bayesteh, and H. Baligh, “SCMA codebook design,” in 2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall). IEEE, 2014, pp. 1–5.
[4] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 498–519, Feb. 2001.
[5] S. S. Mohamed and H. N. Abdullah, “Interleaving based SCMA Codebook Design Using Arnold’s Cat Chaotic Map,” Iraqi Journal of ICT, vol. 6, no. 2, pp. 42–57, Jan. 2024, doi: 10.31987/ijict.6.2.225.
[6] H. Nikopour and M. Baligh, “Systems and methods for sparse code multiple access,” Jan. 19, 2016, US Patent 9,240,853.
[7] F. Wei and W. Chen, “A low complexity SCMA decoder based on list sphere decoding,” in Proc. IEEE GLOBECOM, Washington, DC, USA, Dec. 2016, pp. 1–6.
[8] C. Lu, W. Xu, H. Shen, H. Zhang, and X. You, “An Enhanced SCMA Detector Enabled by Deep Neural Network,” Aug. 24, 2018, arXiv: arXiv: 1808.08015.
[9] M. Kim, N.-I. Kim, W. Lee, and D.-H. Cho, “Deep learning-aided SCMA,” IEEE Communications Letters, vol. 22, no. 4, pp. 720–723, 2018.
[10] E. Ranjan, A. Vikram, A. Rajesh, and P. K. Bora, "Auto-SCMA: Learning Codebook for Sparse Code Multiple Access using Machine Learning," in Proc. 2021 National Conference on Communications (NCC), San Francisco, CA, USA, Mar. 2021, pp. 1–5, doi: 10.1109/NCC52529.2021.9530173
[11] Altera Innovate Asia website, Presentation “1st 5G Algorithm InnovationCompetition-ENV1.0-SCMA,”
[12] S. Zhang et al., “A capacity-based codebook design method for parse codes multiple access systems,” in 2016 8th International Conference on Wireless Communications  Signal Processing (WCSP), Yangzhou, China: IEEE, Oct. 2016, pp. 1–5. doi: 10.1109/WCSP.2016.7752620.
[13] V.P. Klimentyev and A. B. Sergienko, “SCMA codebooks optimization based on genetic algorithm,” in Proc. 23rd Eur. Wireless Conf. Eur. Wireless, May 2017, pp. 1–6. doi: 10.3969/j.issn.0372-2112.2014.09.002
[14] W. Yu, F. Sohrabi, and T. Jiang, "Role of Deep Learning in Wireless Communications," IEEE BITS the Information Theory Magazine, Nov. 2022. vol. 2, no. 2, pp. 56-72,
[15] L. Dai, R. Jiao, F. Adachi, H. V. Poor, and L. Hanzo, "Deep Learning for Wireless Communications: An Emerging Interdisciplinary Paradigm," IEEE Journal on Selected Areas in Communications, Oct. 2020 vol. 38, no. 10, pp. 2207-2225.
[16] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” arXiv preprint arXiv: 1603.04467, 2016.