Original scientific paper
https://doi.org/10.33180/InfMIDEM2025.201

Informacije MIDEM
Journal of Microelectronics, Electronic Components and Materials
Vol. 55, No. 2(2025), 77 – 86

Coati Optimized Hybrid Neural Network for Efficient Network Slicing in 5 Generation Network

Ayya Dhurai Suceelal Sindhu[1], Chellappan Agees Kumar[2]

[1]Department of Electronics and Communication Engineering, Rajas Institute of Technology, Nagercoil, Tamil Nadu, India.
[2]Department of Electrical and Electronics Engineering, Arunachala College of Engineering for Women, Vellichanthai, India.

Abstract: Network slicing (NS) divides the physical network into many logical networks in order to support the variety of new applications with higher performance and flexibility needs. As a result of these applications, a massive amount of data has been generated with a huge number of mobile phones. Due to this, NS performance has been greatly impacted and extreme challenges have been created. To efficiently handle the challenges, this paper proposes a novel Optimal Network slice Classification Using Deep learning (ONE-CLOUD) technique, which integrates the Coati Optimization Algorithm (COA), GhostNet, and Gated Dilated Convolutional Neural Network (CNN). COA optimizes features such as user device type, packet loss ratio, and delay rate, employing GhostNet model, and Gated Dilated CNN for network slice classification. The proposed method classifies network slices into enhanced Mobile BroadBand (eMBB), Ultra-Reliable and Low-Latency Communications (URLLC), and massive Machine-Type Communications (mMTC). The effectiveness of the suggested approach has been evaluated using the 5G-SliciNdd dataset, utilizing evaluation criteria like accuracy, precision, recall, sensitivity, specificity, throughput, and reduced latency. The overall accuracy of the proposed method is 5.78%, 2.78% and 4.70% higher than the existing DQN-E2E, DRL, and AAA techniques respectively.

Keywords: Network Slicing; Deep learning; GhostNet; Gated Dilated CNN; Coati Optimization.

Coatijevo optimizirano hibridno nevronsko omrežje za učinkovito rezanje omrežja v omrežju petih generacij

Izvleček: Razrez omrežja (NS) razdeli fizično omrežje na več logičnih omrežij, da bi podprl različne nove aplikacije z večjo zmogljivostjo in prilagodljivostjo. Zaradi teh aplikacij se je z velikim številom mobilnih telefonov ustvarila ogromna količina podatkov. To je močno vplivalo na zmogljivost omrežja NS in povzročilo izjemne izzive. Za učinkovito obvladovanje teh izzivov članek predlaga novo tehniko optimalne klasifikacije omrežnih rezin z uporabo globokega učenja (ONE-CLOUD), ki združuje algoritem COA (Coati Optimization Algorithm), GhostNet in gated dilated konvolucijsko nevronsko mrežo (CNN). COA optimizira lastnosti, kot so vrsta uporabniške naprave, stopnja izgube paketov in stopnja zamude, pri čemer uporablja model GhostNet in Gated Dilated CNN za klasifikacijo omrežnih rezin. Predlagana metoda razvršča omrežne rezine v izboljšano mobilno širokopasovno omrežje (eMBB), izjemno zanesljive komunikacije z nizko zakasnitvijo (URLLC) in množične komunikacije strojnega tipa (mMTC). Učinkovitost predlaganega pristopa je bila ocenjena z uporabo podatkovne zbirke 5G-SliciNdd, pri čemer so bila uporabljena merila za ocenjevanje, kot so natančnost, točnost, priklic, občutljivost, specifičnost, prepustnost in zmanjšana zakasnitev. Skupna natančnost predlagane metode je za 5,78 %, 2,78 % in 4,70 % višja od obstoječih tehnik DQN-E2E, DRL in AAA.

Ključne besede: Rezanje omrežja; globoko učenje; GhostNet; Gated Dilated CNN; Coati optimizacija.

** Corresponding Author's e-mail: ageeskumar12@outlook.com*

How to cite:
A. D. S. Sindhu et al., "Coati Optimized Hybrid Neural Network for Efficient Network Slicing in 5 Generation Network", Inf. Midem-J. Microelectron. Electron. Compon. Mater., Vol. 55, No. 2(2025), pp. 77–86

1 Introduction

Network slicing is an innovative architype for building system services that 5G networks have promoted with the growth of Software-Defined Networking (SDN) and Network Function Virtualization (NFV) [1]. 5G systems face immense demand due to mobile tech growth and app diversity. They must bolster Quality of Service (QoS) for multiple sectors like virtual reality, augmented reality, and remote healthcare, necessitating unprecedented advancement [2,3].

In 5G systems, NS defines autonomous, cohesive networks composed of a blend of dedicated and communal resource instances, including system equipment, radio spectrum, and VNF [4]. 5G system is designed to be a versatile, multi-service infrastructure that accommodates a diverse range of services, including eMBB, URLLC, and mMTC [5]. One of the utilization cases for future 5G, low-inactivity correspondence, is supposed to be upheld by MEC, a basic 5G improvement innovation [6]. It brings far-off systems administration, storage, and public distributed computing capacities nearer to the edge of the organization [7].

A network slice contains different organization components, for example, the terminal, access organization, center organization, and transport organization, which can be used by numerous administrators [8]. Unique in relation to other network slices, a network slice has devoted or potentially shared assets [9]. Portable network slice administrators will deliver diverse network slices bundled into one product for business clients with varying requirements, including a single network slice type catering to different verticals [10,11].

To effectively establish and manage network slices that meet QoS requirements amid changing conditions, handling extensive data swiftly proves challenging for humans [12,13]. The automatic method for managing network slices is critical because manual slice assignment is inefficient when dealing with the vast amount of data and dynamic conditions in 5G networks. Automatic classification enhances resource allocation by quickly adjusting to changing user demands and network conditions, thereby ensuring optimal performance without the delay and potential human error associated with custom manual assignments. It also supports the scalability required to manage the complexity and diversity of modern 5G applications, such as VR, AR, and remote healthcare. To address this issue a novel Optimal Network slice Classification Using Deep learning (ONE-CLOUD) technique, has been suggested. The main contributions are as follows:

- Packet loss ratio, delay rate, speed, device type, slice type, user bandwidth, and other attributes are gathered initially from the various users or devices in the 5G network.
- After collecting these features, Coati Optimization Algorithm (COA) is employed to select features from the collected attributes. Subsequently, the selected features are output in the form of optimal weighted features.
- The NS prediction is achieved by hybridizing GhostNet and Gated Dilated CNN through the AND operation, using the newly extracted weight optimized features. The output categorizes the network slices into three types: eMBB, mMTC, and URLLC.

The remainder of the research is described as follows: Section II examines the study using the literature as a guide. Section III thoroughly explains the suggested system. Section IV shows the result and discussion, whereas Section V shows the conclusion.

2 Literature survey

Several studies have utilized several techniques to NS in recent years. The following section covers a few of the current evaluation approaches along with their disadvantages are as follows:

In 2020, Li, T., et al., [14] suggested an E2E system slicing source distribution system that operates in multi-slice and multi-service scenarios, based on Deep Q-Networks (DQN). This system dynamically allocates resources to optimize by considering both the fundamental system slices and the radio access network slices. To the access side's ideal allocation approach, the typical access rate is enlarged by 9% for slices with delay limitations and by 5% for slices with rate constraints. In 2021, He, Y., et al., [15] recommended a multi-chain 5G NS facility value computation model to ascertain the characteristics of the NS service quality. The Cosi protocol features lower traffic consumption and a steady calculation cost as compared to other protocols. Ultimately, the practicality and effectiveness of the multi-chain 5G NS facility value computation architecture is demonstrated by security analysis and experimental outcomes.

In 2022, Suh, K., et al., [16] suggested a deep reinforcement learning (DRL)-based NS method to determine the source provision strategy that maximizes long-term amount in B5G systems while meeting QoS standards. The suggested method is shown to be efficient in addressing the coexistence of use cases in B5G environments and optimizing long-term throughput by numerical findings. In 2023, Dangi, R. and Lalwani, P., [17] suggested a successful hybrid learning algorithm-based network-slicing technique to enhance QoS and maximize NS, the results produced by the suggested model are contrasted with those of current deep learning, machine learning, and optimization methods. It proves that the suggested model performed better than the others and identified the right network slices to provide top-notch services.

In 2023, Hu, Y., et al., [18] recommended neural network-based carrying technique. This paper provides a power 5G slicing service carrying mechanism based on neural networks. Through simulation verification, proved that the properties of electric power services are retrieved, classified, matched, and compliant with the 5G power NS. In 2023, Botez, R., et al., [19] suggested a modified A* algorithm Targeting services with low or extremely low latency requirements, it offers a better way to NS in 5G backhaul networks. According to experimental data, the suggested technique improves processing time by an order of magnitude. These outcomes show how well our method works in 5G backhaul networks to achieve URLLC. In 2024, Gomes, R., et al., [20] suggested the Artificial Algae Algorithm (AAA) as a 5G-specific NS solution for the VNE problem. The runtime presentation of AAA is independent of the number of simulated nodes this results in execution times that are up to ten times faster than DE and PSO when taking into account 30 nodes. The suggested method using AAA demonstrated an improvement of more than 60% in an implementation time that was ten epochs faster.

The aforementioned techniques have a number of issues with NS, including low accuracy, high Latency. To overcome these challenges a novel ONE-CLOUD technique has been proposed and discussed in next section.

3 Optimal network slice classification using deep learning

In this section, a novel Optimal Network slice Classification Using Deep learning (ONE-CLOUD) technique has been proposed to optimize resource utilization, and enhance the flexibility and efficiency of 5G and beyond networks. Initially, various attributes like bandwidth, device type, speed, slice type, packet loss ratio, and delay rate are gathered from different devices or users in the 5G network. These features undergo COA for selection, resulting in optimal weighted features. GhostNet is hy-

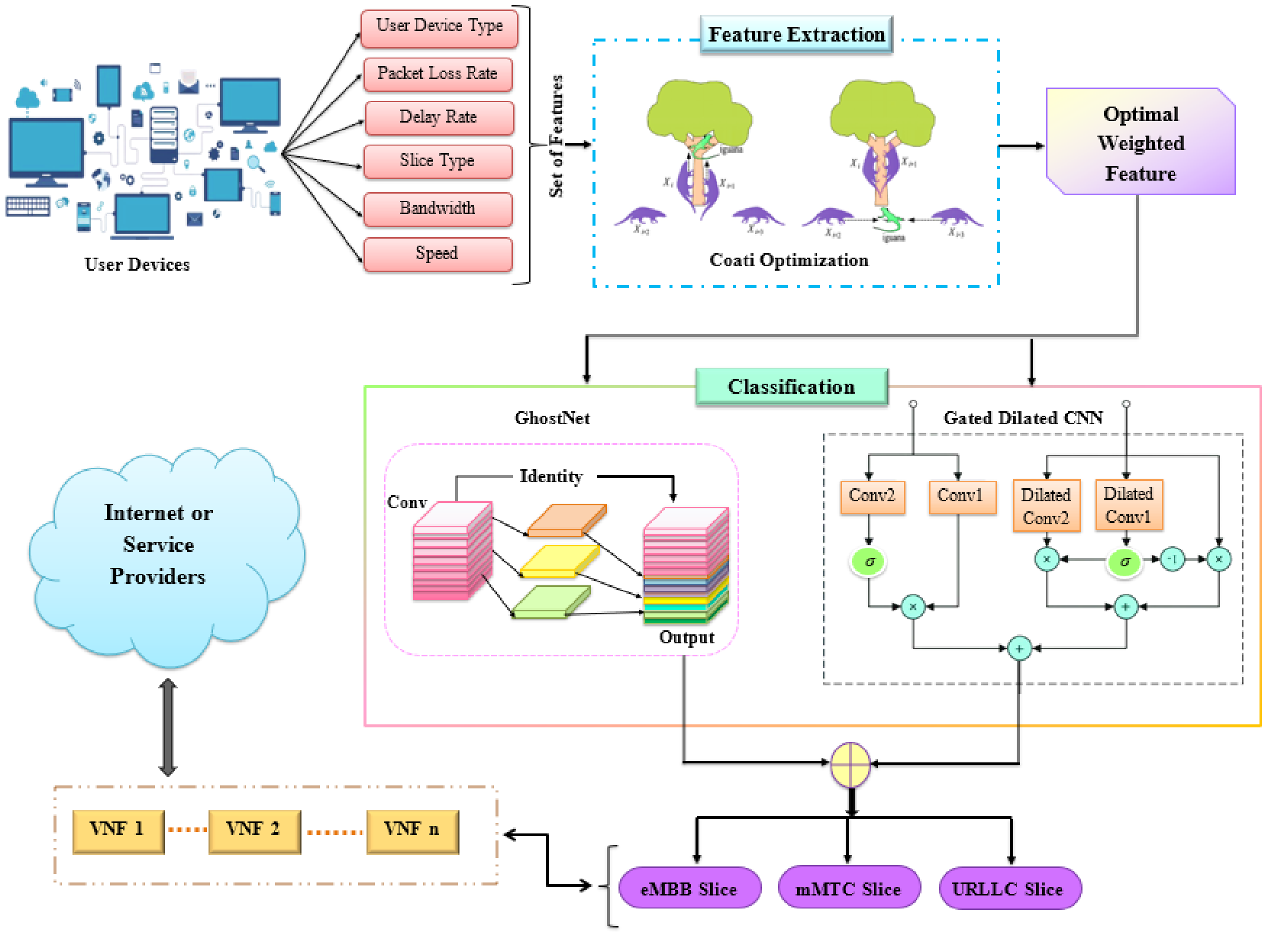


Figure 1: Overall workflow of Proposed ONE-CLOUD Method

bridized with Gated Dilated CNN using the AND operation to predict NS, leveraging the newly optimized features. The output classifies network slices into three types: eMBB, mMTC, and URLLC. The overall proposed ONE-CLOUD's workflow is depicted in Figure 1.

3.1 Feature extraction

In NS, feature extraction is the process of locating and obtaining pertinent data or attributes from the collection of features linked to each slice.

3.1.1 Feature Extraction

The COA [21] is a recently developed bioinspired optimization technique influenced by the natural behavior of coatis, presents a novel approach for feature extraction in NS. COA is based on the essential idea of imitating two important coatis' behaviors: (i) chasing and fighting iguanas and (ii) running away from predators. The COA is considered the most suitable technique for network slicing feature extraction since its bioinspired mechanisms mimic coatis' hunting and evasion behaviours. These behaviours enable more effective exploration and exploitation of the solution space. The COA is especially well-suited for challenges like network slicing, which demands for dynamic adaptability to changing conditions and limits, as it has proven to perform well in balancing global search capability and local search precision. When compared to other optimization algorithms like particle Swarm Optimization (PSO) and Cuckoo Search Optimization (CSO), COA's dual strategy allows for a more thorough exploration of potential solutions and reduces the probability of getting trapped in local optima. This COA behaviour facilitates more efficient resource distribution in complicated 5G scenarios. Performance measures like latency and throughput are improved by COA's ability to manage high-dimensional features such as device type, packet loss, and delay rates in network slicing of 5G. Because the mentioned algorithms might not provide the same balance between exploration and exploitation needed for network slice optimization, this helps in our decision to choose the COA algorithm. The coatis's original location in the hunt space is determined at random using Eqn. (1) at the initial stage of the COA implementation.

$$Z_j : Z_{ij} = LW_i + k \cdot \left(UP_i - LW_i\right), j = 1, 2, \ldots, m, \quad i = 1, 2, \ldots, n \tag{1}$$

where *m* is the amount of coatis, *n* is the amount of choice variables, *k* is a chance actual amount in the interlude 0,1, Z_j is the location of the jth coati in hunt space, and LW_i and UP_i are the inferior and superior bounds of the jth choice variable, correspondingly. The subsequent medium Z, known as the population matrix, is used to numerically depict the inhabitants in the COA which is given in Eqn (2)

$$Z = \begin{bmatrix} Z_1 \\ \vdots \\ Z_j \\ \vdots \\ Z_m \end{bmatrix}_{m\times n} = \begin{bmatrix} z_{1,1} & \cdots & z_{1,j} & \cdots & z_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ z_{j,1} & \cdots & z_{j,i} & \cdots & z_{j,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ z_{m,1} & \cdots & z_{m,j} & \cdots & z_{m,n} \end{bmatrix}_{m\times n} \tag{2}$$

Two of coatis's natural activities are modeled in order to update coatis's position (feature solutions) in the COA. Among these behaviors are: i) The method used by coatis to attack iguanas and ii) The coatis' method of avoiding predators. Consequently, there are two steps to the updating of the COA population.

Phase 1: Strategy for hunting and attacking iguanas (exploration phase)

The first phase updates the coati population by simulating their iguana-attacking strategy. Some climb trees to scare iguanas, while others wait below. Half climb trees, and the rest wait for the iguana to drop. The mathematical simulation of the climbing coatis' location is expressed by Eqn. (3).

$$Z_j^{po1} : z_{ji}^{po1} = z_{ji} + k \cdot \left(Ig_i - I \cdot z_{j,i}\right), \quad for\, i = 1, 2, \ldots, \frac{m}{2} and\ j = 1, 2, \ldots, m. \tag{3}$$

Following its release to the ground, the iguana is placed arbitrarily throughout the search area. Eqn (4), (5).is used to approximate the random position that causes coatis on the pounded to transfer in the hunt space.

$$Ig^C : Ig_i^c = LW_i + k \cdot \left(UP_i - LW_i\right), i = 1, 2, \ldots., n \tag{4}$$

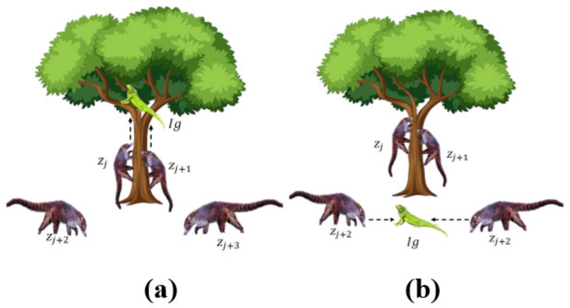
If each coati's new position increases the value of the objective function, it is permitted for the update process; if not, the coati stays in its previous place. Eqn (6) determines the simulated values of j = 1, 2, ... m, to which this update condition is applicable.

$$Z_j^{po1} : z_{j.i}^{po1} = \begin{cases} z_{j.i} + k \cdot \left(Ig_i - I \cdot z_{j,i}\right), E_{Ig^C} < E_j \\ z_{j.i} + k \cdot \left(z_{ji} - Ig_i^c\right), for\ j = \frac{m}{2} + 1, \frac{m}{2} + \ldots, m\, and\ i = 1, 2, \ldots n \end{cases} \tag{5}$$

$$z_j = \begin{cases} Z_j^{po1}, E_j^{po1} < E_j \\ z_j, \ else \end{cases} \quad (6)$$

where k is an arbitrary real number in the range [0, 1], *Jg* stands for the iguana's location in the hunt area, Z_j^{pol} is the original location estimated for the jth coati, $z_{j.i}^{pol}$ is its ith dimension, and E_j^{pol} is its neutral role worth. Jg_i is its ith measurement; Figure 2 shows the coati optimization algorithm's initial phase.

Figure 2: Coati Optimization Algorithm's initial phase: (a) Half of the coatis attacking the tree-dwelling iguana, and (b) the remaining coatis hunting the fallen iguana



Phase 2: The procedure of running away from an assailant (the exploitation stage)

The second phase updates coatis' search space position by modeling their natural behavior when facing predators. When attacked, coatis escape, strategically moving to a safe spot near their current position, showcasing COA's effective local search exploitation ability. To replicate this behavior, a random position is generated near each coati's location using Eqns. (7) and (8).

$$LW_i^{loc} = \frac{LW_i}{d}, UP_j^{loc} = \frac{UP_i}{d}, where\, d = 1, 2, \ldots, D \quad (7)$$

$$Z_j^{po2} : z_{j.i}^{po2} = z_{j.i} + (1-2k)\cdot\left(LW_i^{loc} + k\cdot\left(UP_i^{loc} - LW_i^{loc}\right)\right) \quad (8)$$

The recently computed position is deemed suitable if it enhances the objective function value, a condition simulated by employing Eqn. (9).

$$z_j = \begin{cases} Z_j^{po2}, E_j^{po2} < E_j \\ z_j, \ else \end{cases} \quad (9)$$

Here, Z_j^{po2} is the original location determined for the ith coati using another stage of COA; $z_{j.i}^{pol}$ is its ith dimension; E_j^{po2} is its impartial role value; k is a chance amount within the break [0, 1]; COA to help categorise the weight function, and optimal weight features are the output from the feature extraction process. The description of various features used in NS is given in Table 1.

Table 1: Overall summary of various features used in network slicing

Features	Feature Description
User device type	Properties describe characters and parts of a device
Packet loss rate	Percentage of packet vanish with respect to packet transmitted
Bandwidth	Fastest transfer of information rate possible with an internet connection
Delay rate	The time frame before an event occurs
Speed	Dimensions of location variation

3.2 Network slicing with the commitment of ghostnet and gated dilated CNN

Network slicing employs GhostNet and Gated Dilated CNN for efficient classification, enhancing performance and optimizing resource allocation in diverse network environments. The proposed technique combines the GhostNet and Gated Dilated CNN in network slicing, which addresses the challenges in classification, especially in handling large and distinct data from 5G network slices.

3.2.1 GhostNet Model

The fundamental unit of GhostNet is a stack of Ghost bottlenecks, of which the Ghost modules are the building hunks. The primary layer is a standard convolutional layer with 16 filters, followed by a series of Ghost bottlenecks with increasingly more channels. Using a convolutional layer and global average pooling, the feature maps are ultimately transformed into a feature vector for the final classification. The Optimal weighted features are given as input to the GhostNet. It is suitable for classifying network slices where resource constraints are common and its input is an optimal weighted feature. The convolution operation in a Ghost Module is given in Eqn (10):

$$q_i = \sum_{j \in I_i} v_j y_j \quad (10)$$

Here *p* be the input to the Ghost Module, and q be the output, *i* indexes the output channels, S_j is the set of indices corresponding to the output channels, V_j is the weight associated with input channel y_i and y_i is the input feature map. The Ghost Module introduces a ghost set G_i of randomly selected indices from the set I_i, and the convolution operation is given in Eqn (11)

$$q_i = \sum_{i \in G_i} v_j p_j \tag{11}$$

The ghost set G_i is dynamically sampled during each forward pass, leading to parameter-efficient training. With reference to intrinsic feature maps, $X' \in \mathbb{R}^{b' \times v' \times n}$ can be generated by Eqn (12), $j' \grave{o} \mathbb{R}^{e \times r \times r \times n}$ is the convolutional filters. Nevertheless, as Eqn (13) illustrates, partial convolutional operations are performed, and the remaining feature maps are produced via a linear operation.

$$X' = Y^* j' + b \tag{12}$$

$$X_{j,i} = \theta_{j,i}\left(X_j^{'}\right), \forall_j = 1,\ldots,n, i = 1,\ldots,h, \tag{13}$$

Where, $X_j^{'}$ is the j-th inherent feature map in X', $\theta_{j,i}$ the direct process for generating the i-th ghost feature map $X_{j,i}$. The outputs from all the ghost branches are aggregated to obtain the final output Q which is given in Eqn (14).

$$Q = \sum_{j=1}^{M} Z_j \tag{14}$$

Where, is the total number of ghost breaches. As an improvement, we used the AND operation to optimize the weight function after receiving the output Q from GhostNet. GhostNet model is highly efficient in extracting features from high-dimensional data with fewer parameters, making it suitable for real-time and resource-constrained environments such as 5G networks. GhostNet's ability to generate additional feature maps through simple linear transformations helps in reducing the computational burden while retaining critical information for classifying network slices (eMBB, mMTC, URLLC). Utilizing fewer convolutional operations ensures that the model is lightweight, making it ideal for environments with limited computational power.

3.2.2 Gated dilated CNN model

Gated Dilated Convolutional Neural Networks are a type of DL architecture that combines the concepts of dilated convolutions and gated units to capture long-range dependencies in input data. Dilated convolution that introduces gaps between the weights. Eqn (15) provides the expression for the dilated convolution operation on a 1D sequence.

$$(a \times f)(j) = \sum_{s=1}^{S} a(j + dr \cdot s) \cdot f(s) \tag{15}$$

where indicates the convolution process, is the dilation rate, is the filter size, is the filter or kernal, and is the input sequence. Two gates are used in the gating mechanism: the reset gate (rg) and the update gate (ug). Eqn (16) & (17) is used to calculate the update gate (z) and reset gate (r) using sigmoid activation functions.

$$ug = \mu\left(V_{ug} \times \left[a, h_{s-1}\right]\right) \tag{16}$$

$$rg = \mu\left(V_{rg} \times \left[a, h_{s-1}\right]\right) \tag{17}$$

The Contender concealed state (is then figured using reset gate, which is given in the Eqn (18). Finally, the actual concealed state (is figured Using the update gate to combine the candidate hidden state with the current hidden state, which is given in Eqn (19)

$$\widetilde{h_s} = \tan h\left(V_h \times \left[rg \odot h_{s-1}, a\right]\right) \tag{18}$$

$$h_s = \left(1 - ug\right) \odot h_{s-1} + ug \odot h_s \tag{19}$$

Here, V_{ug}, V_{rg}, and V_h are weight matrices, is the sigmoid activation function, $\odot$ indicates multiplication of elements, and tanh is the hyperbolic tangent activation function. If L describes the output of the last layer before the softmax activation, the final output (O_p) can be computed as in Eqn (20)

$$O_p = softmax\left(L\right) \tag{20}$$

Finally, the output of two models is merged by AND operation and classifies the NS types into 3 classes such as eMBB, mMTC and URLLC. The Gated Dilated CNN is integrated to capture long-range dependencies in the data. 5G networks often generate complex temporal sequences, and traditional CNNs may fail to exploit these patterns fully. The dilated convolutions in the Gated Dilated CNN allow the model to handle long-range dependencies efficiently by expanding the receptive field without increasing the number of parameters. This mechanism is particularly effective in classifying diverse network slices, as it captures both short-term and long-term dependencies in the data, which is crucial for optimizing network performance in real time. By combining GhostNet and Gated Dilated CNN through the AND operation, the proposed ONE-CLOUD technique ensures optimal feature extraction and classification, addressing both the computational efficiency and the complexity of network slicing classification.

4 Results and discussion

The proposed ONE-CLOUD technique's simulation outcomes are obtainable in this section to assess the efficiency of the proposed technique. Performance scrutiny and execution of the suggested 5G NS were conducted in MATLAB.

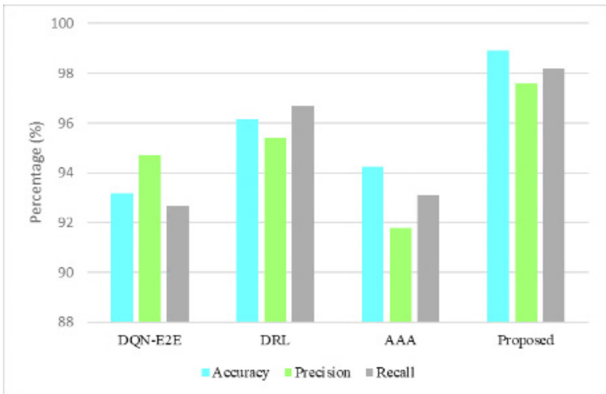


Figure 3: Overall Performance comparison in terms of accuracy, precision, and recall

The effectiveness of the technique was assessed using the 5G-SliciNdd dataset. The dataset used in this work has been split into training, validation, and testing sets, which has been divided into 80%, 10%, and 10% of the entire dataset. The dataset split has been done randomly which ensures that each class is represented proportionally in each subset to prevent class imbalance. The training set has been used to train the network, validation set has been used to fine tune the hyperparameters of the network, and the test set is used to test the network and its performance in NS. With this splitting, we can ensure that the network can reduce overfitting of training data which results in generalization. Additionally, the separate test set will help in evaluating the generalizability and robustness of proposed ONE-CLOUD technique. This ensures that the results state the actual performance of the network on test data.

The proposed ONE-CLOUD model's effectiveness is contrasted with DQN-E2E [14], DRL [16], and AAA [20] in terms F1-score, accuracy, sensitivity, specificity, precision, throughput and latency. In Figure 3, a comprehensive evaluation of overall performance is presented, comparing accuracy, precision, and recall of NS against existing DQN-E2E, DRL, and AAA techniques. The assessment provides insights into how effectively the proposed ONE-CLOUD NS approach performs in comparison to established methods. This comparison aids in gauging the efficacy of NS in comparison to existing techniques.

Figures 4(a) and 4(b) show the training and test data sets, as well as the accuracy and loss curves. The Accuracy Curve in Subfigure (a) shows how the model's correctness upsurges on both the training and authentication sets during the course of training epochs. Both training and validation losses tend to be downward, according to the Loss Curve in Subfigure (b).

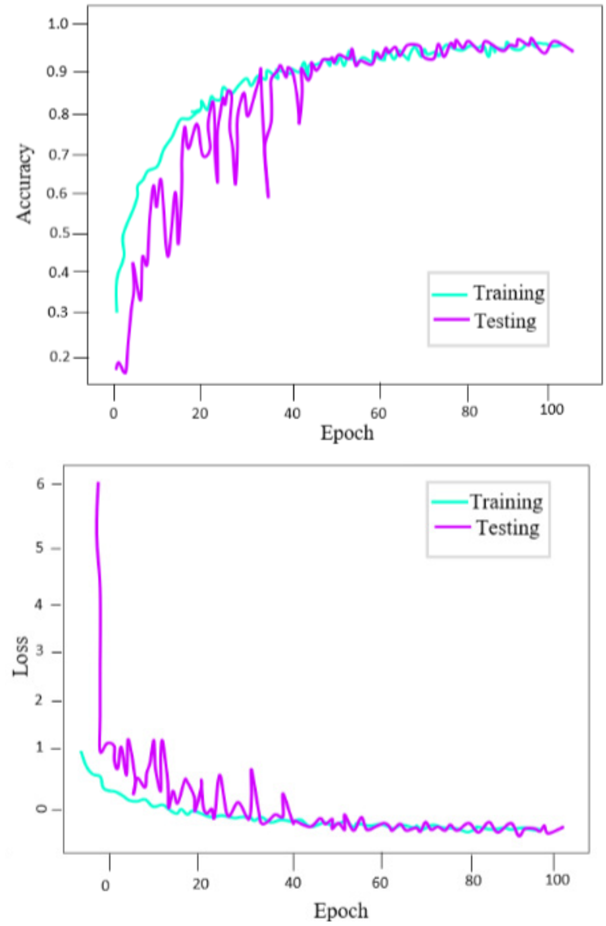


Figure 4: (a) Accuracy Curve; (b) Loss Curve

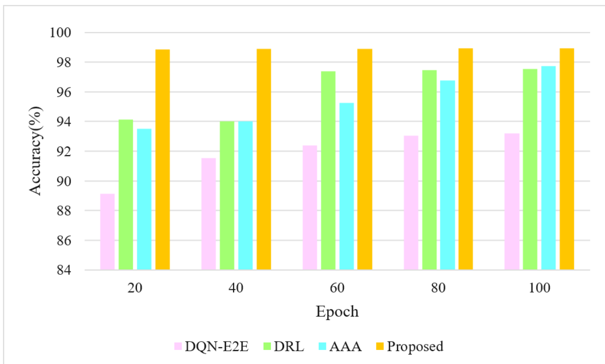


Figure 5: Comparison in terms of accuracy

Figure 5 illustrates a focused comparison in terms of accuracy with 100 epochs between the proposed ONE-CLOUD technique and existing DQN-E2E, DRL, and AAA methods.

The graph offers a visual representation of how well the new approach performs in terms of correctness

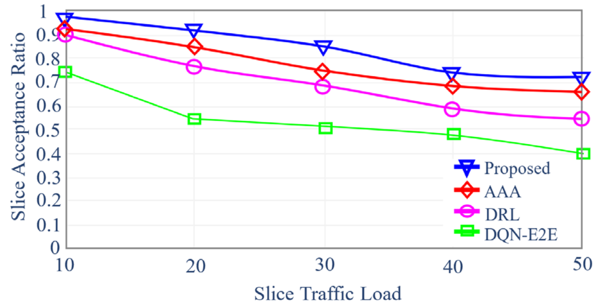


Figure 6: Slice acceptance ratio with variable slice traffic load

compared to established techniques. Comparing the accuracy of the suggested ONE-CLOUD method to the current DQN-E2E, DRL, and AAA procedures, it is 5.78%, 2.78%, and 4.70% higher.

In Figure 6, the slice receiving ratio is presented alongside variable slice traffic loads, linking the presentation of the proposed ONE-CLOUD technique with existing DQN-E2E, DRL, and AAA methods. This figure allows for an assessment of how well the proposed method adapts to varying levels of network demand compared to established techniques.

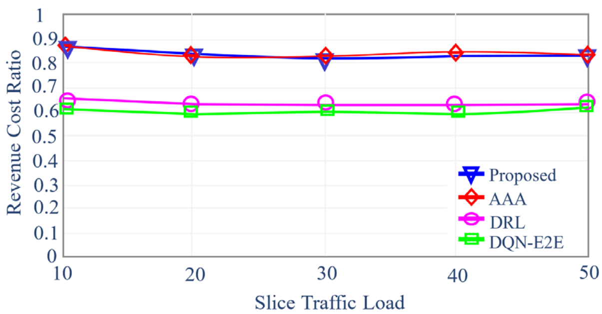


Figure 7: Revenue-to-cost ratio with variable slice traffic load

Figure 7 illustrates the revenue-to-cost ratio in relation to mutable share traffic loads for both the existing DQN-E2E, DRL, and AAA methods and the proposed ONE-CLOUD technique for NS. A higher revenue-to-cost ratio indicates improved cost-effectiveness, highlighting the potential benefits of implementing the proposed NS method in comparison to the conventional system

Figure 8 presents a comparative analysis of sensitivity and specificity between the existing system and the proposed ONE-CLOUD method for NS. The specificity and sensitivity of the proposed ONE-CLOUD method are 3.90%, 9.25%, 12.44% and 5.02%, 4.18%, 6.27% greater than the existing AAA, DRL and DQN-E2E techniques respectively.

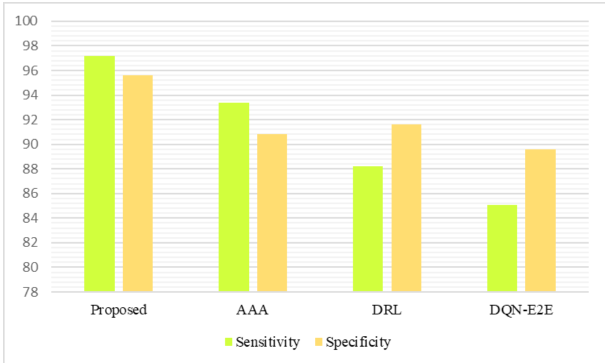


Figure 8: Comparison in terms of sensitivity and specificity

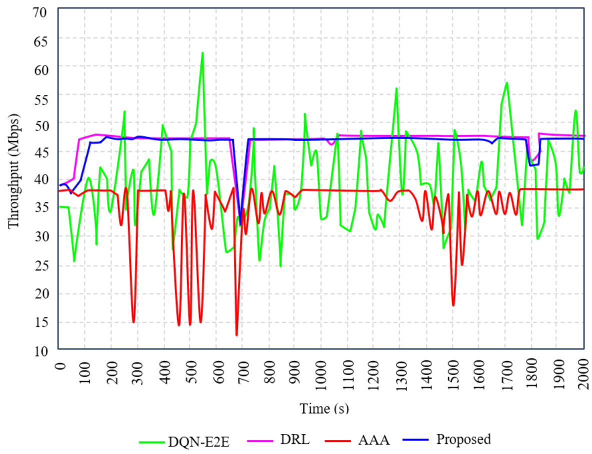


Figure 9: Comparison in terms of throughput

Figure 9 presents a comparison of throughput between the existing AAA, DRL and DQN-E2E techniques and the proposed ONE-CLOUD method for NS. This graph enables an assessment of how the proposed NS method performs in terms of data transmission efficacy associated to the existing system, providing valuable insights into the potential improvements in throughput offered by the proposed ONE-CLOUD approach.

Figure 10 illustrates a latency comparison between the proposed ONE-CLOUD method and existing AAA, DRL and DQN-E2E techniques. The proposed method dem-

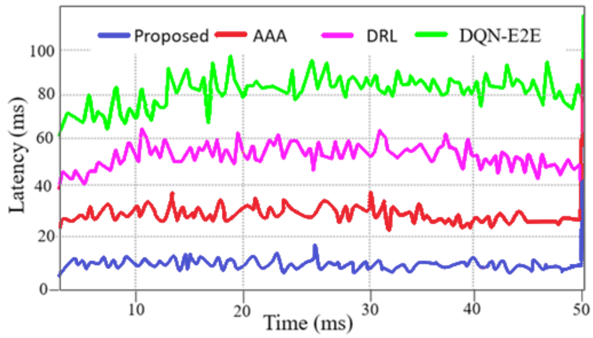


Figure 10: Comparison in terms of latency

onstrates superior latency performance compared to current methods. This comparison offers valued visions into the effectiveness and efficiency of the proposed NS approach, showcasing its potential to minimize communication delays.

5 Conclusion

In this paper, a novel Optimal NEtwork slice CLassificatiOn Using Deep learning (ONE-CLOUD) technique has been proposed to optimize resource utilization, and enhance the flexibility and efficiency of 5G and beyond networks. The COA-based feature extraction optimizes device attributes. These features enhance GhostNet and Gated Dilated CNN models, combined via AND operation, boosting accuracy in classifying eMBB, mMTC, and URLLC network slices. The evaluation of the proposed ONE-CLOUD method, conducted using the 5G-SliciNdd dataset. The proposed ONE-CLOUD method outperforms existing techniques, in terms of precision, accuracy, latency, sensitivity, specificity, throughput, and recall. The overall accuracy of the proposed ONE-CLOUD method is 5.78%, 2.78% and 4.70% higher than the existing DQN-E2E, DRL, and AAA techniques respectively. Future work could explore the scalability and applicability of the proposed technique in large-scale network environments, as well as its adaptability to emerging communication technologies beyond the scope of 5G.

6 Acknowledgments

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

7 Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

8 References

1. A.A. Barakabitze, A. Ahmad, R. Mijumbi, A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," Comput. Networks, vol. 167, pp. 106984, 2020, https://doi.org/10.1016/j.comnet.2019.106984.
2. A. Ahilan, M.A. Rejula, S.N.Kumar, and B.M. Kumar, "Virtual Reality Sensor Based IoT Embedded System for Stress Diagnosis," IEEE Sensors Journal. 2023.
3. S. Zafar, N. Iftekhar, A.Yadav, A. Ahilan, S.N. Kumar, and A. Jeyam, "An IoT method for telemedicine: Lossless medical image compression using local adaptive blocks," IEEE Sensors Journal, vol. 22, no. 15, pp.15345-15352, 2022.
4. Dr. S. Reeba Rex, Dr. T. Pravin Rose, S. Amudaria, "Real Time Remote Monitoring Via Horse Head Optimization Deep Learning NetworK," International Journal of Data Science and Artificial Intelligence, vol. 02, no.02, pp. 42-47, 2024.
5. A. Karimidehkordi, "Multi-Service Radio Resource Management for 5G Networks". 2019.
6. P. Mor, S.B. Bajaj, "Enabling Technologies and Architecture for 5G-Enabled IoT," Blockchain for 5G-Enabled IoT: The new wave for Industrial Automation, pp. 223-259, 2021, https://doi.org/10.1007/978-3-030-67490-8_9.
7. W. Yu, F. Liang, X. He, W.G. Hatcher, C. Lu, J. Lin, X. Yang, "A survey on the edge computing for the Internet of Things," IEEE access, vol. 6, pp. 6900-6919, 2017. https://doi.org/10.1109/ACCESS.2017.2778504.
8. X. Zhou, R. Li, T. Chen, H. Zhang, "Network slicing as a service: enabling enterprises' own software-defined cellular networks," IEEE Commun. Mag., vol. 54, no. 7, pp. 146-153, 2016, https://doi.org/10.1109/MCOM.2016.7509393.
9. R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, H. Zhang, "Intelligent 5G: When cellular networks meet artificial intelligence," IEEE Wireless Commun.., vol. 24, no. 5, pp.175-183, 2017. https://doi.org/10.1109/MWC.2017.1600304WC.
10. P. Govender, "Dynamic quality of service enabled network slicing for fifth generation core network (Doctoral dissertation, University of Johannesburg)". 2023.
11. T. Irshad, "Design and implementation of a testbed for network slicing". 2018.
12. F. Firouzi, B. Farahani, and A. Marinšek, "The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT)," Inf. Syst., vol. 107, pp.101840, 2022, https://doi.org/10.1016/j.is.2021.101840.
13. M. El Rajab, L. Yang, A. "Shami, Zero-touch networks: Towards next-generation network automation," Computer Networks, pp.110294, 2024, https://doi.org/10.1016/j.comnet.2024.110294.
14. T. Li, X. Zhu, and X. Liu, "An end-to-end network slicing algorithm based on deep Q-learning for 5G network," IEEE Access, vol. 8, pp. 122229-122240, 2020, https://doi.org/10.1109/ACCESS.2020.3006502.

15. Y. He, C. Zhang, B. Wu, Y. Yang, K. Xiao, H. Li, "Cross-chain trusted service quality computing scheme for multi-chain model-based 5g network slicing slam," IEEE Internet Things J. 2021. https://doi.org/10.1109/JIOT.2021.3132388
16. K. Suh, S. Kim, Y. Ahn, S. Kim, H. Ju, and B. Shim, "Deep reinforcement learning-based network slicing for beyond 5G," IEEE Access, vol. 10, pp.7384-7395, 2022, https://doi.org/10.1109/ACCESS.2022.3141789.
17. R. Dangi, P. Lalwani, "Harris Hawks optimization-based hybrid deep learning model for efficient network slicing in 5G network," Cluster Comput., pp.1-15, 2023, https://doi.org/10.1007/s10586-022-03960-1.
18. Y. Hu, L. Gong, X. Li, H. Li, R. Zhang, R. Gu, "A Carrying Method for 5G Network Slicing in Smart Grid Communication Services Based on Neural Network," Future Internet, vol. 15, no. 7, pp.247, 2023, https://doi.org/10.3390/fi15070247.
19. R. Botez, A.G. Pasca, A.T. Sferle, I.A. Ivanciu, V. Dobrota, "Efficient Network Slicing with SDN and Heuristic Algorithm for Low Latency Services in 5G/B5G Networks," Sens., vol. 23, no. 13, pp.6053, 2023, https://doi.org/10.3390/s23136053.
20. R. Gomes, D. Vieira, M.B. Pereira, M.F. de Castro, "Artificial algae optimization for Virtual Network Embedding problems in 5G network slicing scenarios," Expert Syst. Appl., vol. 239, pp.122436, 2024, https://doi.org/10.1016/j.eswa.2023.122436.
21. M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, "Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems," Knowledge-Based Systems, vol. 259, pp.110011, 2023. ž https://doi.org/10.1016/j.knosys.2022.110011

Arrived: 13. 07. 2024
Accepted: 24. 11. 2024