International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 6, Issue 1, 209 -217, 2025
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V6I1P123

Original Article

Exponential Pine Cone Optimization Enabled Quantum-Inspired Convolutional Neural Networks for Secure Network Slicing in 5G Network

Saikiran Chevuru
Wireless Engineer II (Incident Manager), Department of Network Operations Center, Dish Network LLC, 5701 S Santa Fe Dr, Littleton, CO 80120.

Received On: 29/12/2024 Revised On: 13/01/2025 Accepted On: 30/01/2025 Published On: 18/02/2025

***Abstract** - Network Slicing is significant in facilitating a multiple of Fifth-generation (5G) applications, use cases, and services. It provides end-to-end isolation among the slices for customizing each slice in terms of service demands. The increasing demands of 5G networks and advanced features to meet complex business needs have rendered conventional approaches insufficient. Therefore, an efficient approach is required to mitigate the issues in the slicing process. As a result, Exponential Pine Cone Optimization enabled Quantum-Inspired Convolutional Neural Network (EPCO_QuCNet) is introduced for secure network slicing in 5G network. The system model of 5G network is carried out. When network slicing receives requests from a User Equipment (UE), various network parameters, like speed, packet loss rate, jitter, and packet delay budget—are gathered from multiple devices. Hence, a secure slicing of network is classified by Quantum-Inspired Convolutional Neural Networks (QuCNet), which is trained by Exponential Pine Cone Optimization (EPCO) algorithm. Furthermore, the services provided in 5G is accessed by the internet service provider using Virtual Network Function (VNF). EPCO_QuCNet has achieved better outcomes for the evaluation metrics, like acceptance rate, execution time and resource efficiency of 0.927, 0.167 sec and 0.911.*

***Keywords** - Network slicing, 5G network, Quantum-Inspired Convolutional Neural Networks, Exponential Weight Moving Average, Pine Cone Optimization Algorithm.*

1. Introduction

5G cellular networks are presently being extended by network operators, with several manufacturers preparing to lower new 5G end-user devices. The developmental aspect of 5G, providing performance improvements over 4G, has already been released to the market. However, the progressive aspect of 5G remains under investigation by the research community and it is still being introduced by standardization organizations. It intends to offer differentiated services, like voice and vehicular communication, video streaming, as well as e-health by sharing the similar 5G infrastructure. This complex target can be attained by utilizing the innovative technology of network slicing [1]. A 5G network enables the creation of multiple isolated logical networks is known as slices, over a shared physical infrastructure, supporting a wide range of use cases with varying service requirements. Network slicing is a significant pillar of 5G networks that enables the operators to provide diverse levels of Quality of Service (QoS) personalized to various user requirements [2]. Network slicing connects full potential of 5G infrastructure to dynamically and efficiently deliver a broad spectrum of heterogeneous services, each with distinct performance as well as reliability requirements. With this emerging technology, security remains as the prime considerations that must be addressed [1].

In 5G networks, enormous amount of data must be evaluated prior to selecting network slices in order to ensure that a network can efficiently attain the QoS requirements. Hence, Machine Learning (ML) and Deep Learning (DL) approaches are employed to evaluate enormous amount of data and make precise identification of network slices in 5G. Furthermore, these approaches are evaluated based on the complexity to provide rapid decisions for network slicing. The integration of various types of DL techniques can maintain the corresponding abilities of all techniques and enhance the generalization. Thus, it leads to superior performance, robustness or flexibility. DL techniques, like Convolutional Neural Networks (CNNs) is superior at extracting and compressing spatial features, while Recurrent Neural Networks (RNNs) are better suited to model sequential or temporal dependencies. Thus, integrating CNN with RNN model may lead to superior accuracy while it is employed for network slicing. While integrating diverse DL methods can lead to complex models, optimization modules help to reduce this complexity by selecting optimal hyperparameters that ensures superior model performance [3]. In this research, a secure module is designed to improve the privacy, and network's efficiency by integrating optimization with DL approach.

The significance of this research is to devise a module for secure network slicing in 5G network named EPCO_QuCNet. The 5G network is initially simulated. Upon receiving requests from a UE, the network slicing process collects parameters from multiple devices. Hence, a secure slicing of network is classified by EPCO_QuCNet. Here, EPCO is devised by incorporating Exponential Weight Moving Average (EWMA) and Pine Cone Optimization Algorithm (PCOA). Furthermore, the services provided in 5G is accessed by internet service provider using VNF.

- **Proposed EPCO_QuCNet for secure network slicing:** In this novelty, an efficient framework for secure network slicing is introduced using hybrid optimization-enabled DL approach. Here, the networks are effectively sliced by employing QuCNet that is tuned by EPCO. Here, EPCO is devised by the integration of EWMA and PCOA.

2. Motivation

The purpose of 5G networks is to support various vertical industries, each with distinct performance requirements. Here, it is regarded as a key enabler for enhancing cellular networks, providing the flexibility needed to achieve this goal. Nevertheless, some issues occurred during the network slicing process are not mitigated. Hence, the researchers are motivated to develop an efficient module for secure network slicing in 5G network.

2.1. Literature survey

R. Dangi and P. Lalwani., [4] developed Harris Hawk Optimization-Convolution Neural Network+Long Short-Term Memory (HHO-CNN+LSTM) for effectual network slicing in 5G. This approach dynamically allocated the network resources, ensuring better user satisfaction. However, the poor data in this approach degraded the performance of the module. K. Suh, *et al*. [5] designed Deep Reinforcement Learning-based network slicing (DRL-NS) for beyond 5G network slicing. This technique efficiently learned optimal resource allocation policies in dynamic environments. Nevertheless, it was unstable and converged slowly, particularly in non-stationary network environments. Z. Z. Saleh, *et al*. [6] presented Double Deep Q-Network with Prioritized Experience Replay with Pointer Network-based Long Short-Term Memory (DDQN-PER with PtrNet-LSTM) for network slicing in 5G. This module had the potential to integrate multiple algorithms for maintaining complex, high-dimensional network states and actions. Nonetheless, it often required significant training data and computational resources, making the module more difficult. S. B. Saad, *et al*. [7] introduced Blockchain-based trust architecture for 5G network slicing. Even though this module reduced risk of cross-slice interference or attacks by facilitating secure and isolated slices tailored to specific service requirements, it obtained high complexity and high cost.

2.2. Challenges

The limitations of previous techniques are demonstrated below:

- The system in [6] improved its policies over time, adapting to emerging network patterns and threats. However, it was complex to maintain real-time learning and decision-making since the network size and complexity was increased.
- Although the module in [7] reduced the vulnerabilities by incorporating security measures in the entire network stack, security mechanisms acquired latency and reduced overall network performance.
- Network slicing in 5G allows operators to create virtual networks on distributed infrastructure. It supports flexible, rapid deployment of services with varying requirements. While offering many benefits, this emerging technology also presents challenges that continue to attract attention from both industry and academia. Thus, hybrid optimization enabled DL framework is devised to overcome these issues.

3. System Model

Figure 1 illustrates a secure systematic model of 5G network. A 5G network is derived by means of maximal amount of data rates [8]. When the link efficiency attained the Shannon limit, researchers are focusing on increasing spectral efficiency by deploying network nodes at higher densities. It comprises macro, pico, and femto cells, wherein the Femto-cells helps to alleviate congestion and reduce the load on macro-cell base stations. A 5G networks supports user-centric connectivity system by direct Device-to-Device (D2D) communication that increases the data rates and decreasing end-to-end delays.

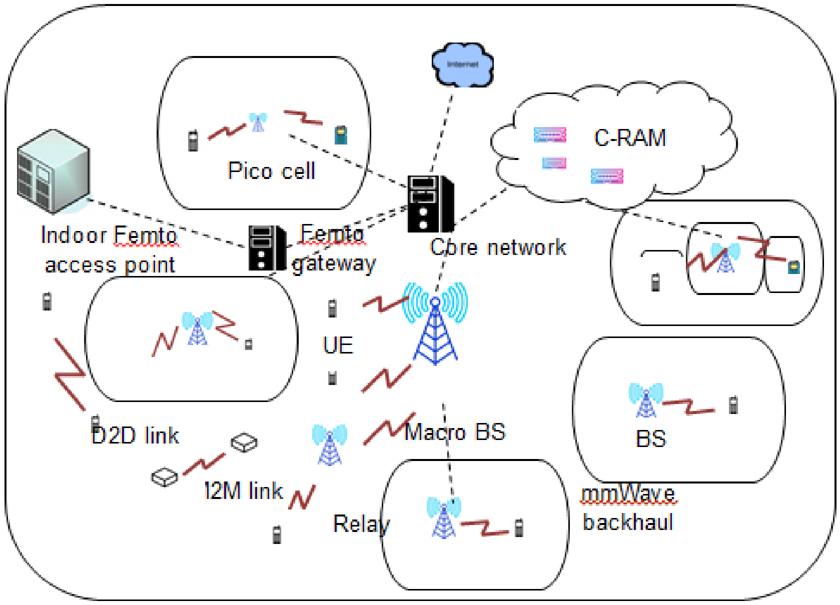


Fig 1: System Model

4. Proposed Exponential Pine Cone Optimization enabled Quantum-Inspired Convolutional Neural Network for secure network slicing in 5G network

The aim of this research is to introduce EPCO_QuCNet for secure network slicing in 5G network. Initially, the systematic module of 5G network is performed. When a UE sends a request, network slicing collects key network parameters, such as speed, packet loss, jitter, and delay from several devices. Therefore, a secure network slice is carried out using QuCNet [9], where the proposed QuCNet will be trained by employing EPCO algorithm. Here, EPCO algorithm is the hybridization of EWMA [10] and PCOA [11]. Furthermore, the services provided 5G is accessed by internet service provider using VNF. Figure 2 displays general illustration of secure network slicing in 5G network using EPCO_QuCNet.

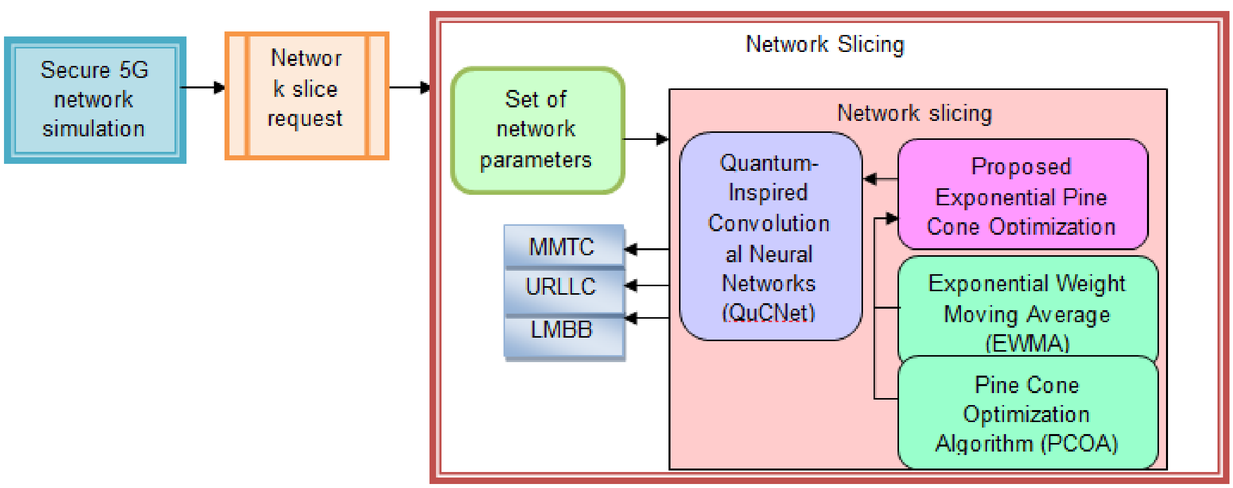


Fig 2: Illustration of Secure Network Slicing in 5g Network Using Epco_Qucnet

4.1. Network Slice Request (NSR)

NSR [12] comprises three kinds of slices for three use case families, such as enhanced mobile broadband (eMBB) $\left(\Re_a\right)$, massive machine-type communications (mMTC) $\left(\Re_b\right)$ and ultra-reliable and low-latency communication (uRLLC) $\left(\Re_c\right)$ that is indicated as $\Re_{\Im}$, which is expressed as $\Re_{\Im} = \Re_a \bigcup \Re_b \bigcup \Re_c$. All requests are indicated as

$Z_{\Re}=\left(Y_{\Re},X_{\Re},W_{\Re},V_{\Re},U_{\Re}\right)$. Here, $Y_{\Re}$ expresses network slice's node, $X_{\Re}$ denotes links, $W_{\Re}$ elucidates capacity, $V_{\Re}$ bandwidth and $U_{\Re}$ represents duration of NSR left in the infrastructure network. Hence, $Z_{\Re_a}=\left(Y_{\Re_a},X_{\Re_a},W_{\Re_a},V_{\Re_a},U_{\Re_a}\right)$ is for the request $\Re_a$ and similarly $Z_{\Re_b}$ and $Z_{\Re_c}$ is for requests $\Re_b$ and $\Re_c$.

4.2. Network Parameters

The network parameters employed in this analysis are described as follows:

4.2.1. Speed

It is represented as scalar quantity and it measures the dimension of location variation [13], which is specified as NP_1

4.2.2. Packet loss rate

It occurs when multiple data packets transmitted across computer network collide, preventing them from reaching their destination. It is determined as a percentage of packets disappeared based on the transmitted packets [13]. It is indicated as NP_2.

4.2.3. Jitter

It refers to the deviation from true periodicity of probably periodic signal, typically measured based on reference clock signal [13]. It is enumerated as NP_3.

4.2.4. Packet delay budget

It specifies the upper bound on the delay that a packet can tolerate [13], which is indicated as NP_4.

Therefore, the network parameters NP are generally enumerated as,

$$\mathrm{NP}=\left\{\mathrm{NP}_1,\mathrm{NP}_2,\mathrm{NP}_3,\mathrm{NP}_4\right\} \tag{1}$$

4.3. Network Slicing

It enables several virtual networks to run on distributed physical infrastructure in 5G networks, where each of them is optimized for certain applications or services. This process is carried out using EPCO_QuCNet with an input of network parameters NP.

4.3.1. Structure of QuCNet

In QuCNet [9], Quanvolutional neural network is employed, where its filters employ random quantum circuits for extracting features from input data by changing the localized spatial subsegments of data, which can be provided as random configurations. This transforming process are explained below:

4.3.1.1. Encoding:

For all filters, it allocates a particular encoding function termed E is considered. It is then applied to the 2D sub segment $\mathrm{M}_{\hbar}$ achieved from input data by the filter. Thus, an initial quantum state $\mathrm{K}_{\hbar}$ is computed by,

$$\mathrm{K}_{\hbar}=\mathrm{E}\left(\mathrm{M}_{\hbar}\right) \tag{2}$$

4.3.1.2. Quantum Random Circuits:

The encoded quantum state $\mathrm{K}_{\hbar}$ is directly passed to this circuit to combine the principles of quantum computing. It is expressed as H that is employed to perform quantum computations and generated through random circuit X. Thus, computational function is resulted as quantum output state $\mathrm{B}_{\hbar}$, which is enumerated as,

$$\mathrm{B}_{\hbar}=\mathrm{X}\left(\mathrm{K}_{\hbar}\right)=\mathrm{X}\left(\mathrm{E}\left(\mathrm{M}_{\hbar}\right)\right) \tag{3}$$

4.3.1.3. Decoding:

In this process, quantum measurement is significant, where quantum state collapses and analyzed by relevant probability distribution during the measurement. Here, decoding function Z is utilized, which allows it to achieve terminal decoded state $\Upsilon_{\hbar}$. This enumerates scalar value and it is computed as,

$$\Upsilon_{\hbar}=\mathrm{Z}\left(\mathrm{B}_{\hbar}\right)=\mathrm{Z}\left(\mathrm{X}\left(\mathrm{E}\left(\mathrm{M}_{\hbar}\right)\right)\right) \tag{4}$$

Therefore, Quantum Filter Transformation (QFT) is modeled as,

$$\Upsilon_{\hbar}=QFT\left(\mathrm{M}_{\hbar},\mathrm{E},\mathrm{X},\mathrm{Z}\right) \tag{5}$$

This integration of QFT enables QuCNet with augmenting its potential to attain accurate slicing process.

Module*:* It comprises two different systems where these systems have same structure of QuCNeet with various complexity. Initially, QFT is applied over network parameters, which is modeled as,

$$D_{ef}^{(g,g)}=QFT\left(\mathrm{M}_{(g,g)}\mathrm{E},\mathrm{X}_f,\mathrm{Z}\right) \tag{6}$$

Here, f implies filter dimension. Quantum-based segments are aligned as f respective channels that is expressed as,

$$channel_f=\left[D_{ef}^{(1,1)},D_{ef}^{(1,2)},...,D_{ef}^{(k,l)}\right] \tag{7}$$

Here, the variables k,l ranging from 1 to height/ f and width/ f. The channeled state signifies the resultant computed state of each sub-segments in its subsequent channels that is described as,

$$D_{channelled} = [C_1, C_2, C_3, C_4] \quad (8)$$

Here, C_1 to C_4 implies four channels. It has input layer, QFT generation, flattening and dense layers. Z-channeled state in Eq. (8) performs flattening before it is forwarded to dense layer.

$$D_{flat} = flat\left(D_{CNN}^2\right) \quad (9)$$

$$D_{den}^1 = h^1\left(G^1 \cdot D_{flat} + i^1\right) \quad (10)$$

$$D_{den}^2 = h^1\left(G^2 \cdot D_{den}^1 + i^2\right) \quad (11)$$

Here, G and i expresses weight and bias of network, h^1 elucidates ReLU function. Moreover, an output layer O with i^2 as softmax function is described as,

$$\mathrm{O} = h^2\left(G^0 \cdot D_{den}^2 + i^0\right) \quad (12)$$

Furthermore, $D_{channelled}$ is acted as input for convolutional as well as pooling layers. They are expressed as,

$$D_{CNN}^1 = \max\_\ poll\left(conv2D\left(D_{channelled}\right)\right) \quad (13)$$

$$D_{CNN}^2 = \max\_\ poll\left(conv2D\left(D_{CNN}^1\right)\right) \quad (14)$$

$$conv2D\left(\mathrm{A}_{(g,o)}\right) = h\left(\sum_{s=1}^{S}\sum_{t=1}^{T} G_{s,t} \cdot \mathrm{A}_{g+s-1,o+t-1} + i\right) \quad (15)$$

Here, s,t depicts dimensions of convolutional filters, h enumerates activation function, G elucidates weight and A enumerates input data.

$$\max\_\ poll\left(\mathrm{A}_{g,o}\right) = \max_{s=1}^{\rho} \max_{t=1}^{\rho} \mathrm{A}_{(o-1)\rho+s,(o-1)\rho+t}$$

Here, ρ elucidates pooling window size. The outcome from D_{CNN} is flattened and process by two dense layers for providing resultant output. Hence, the sliced networks are enumerated as Q.

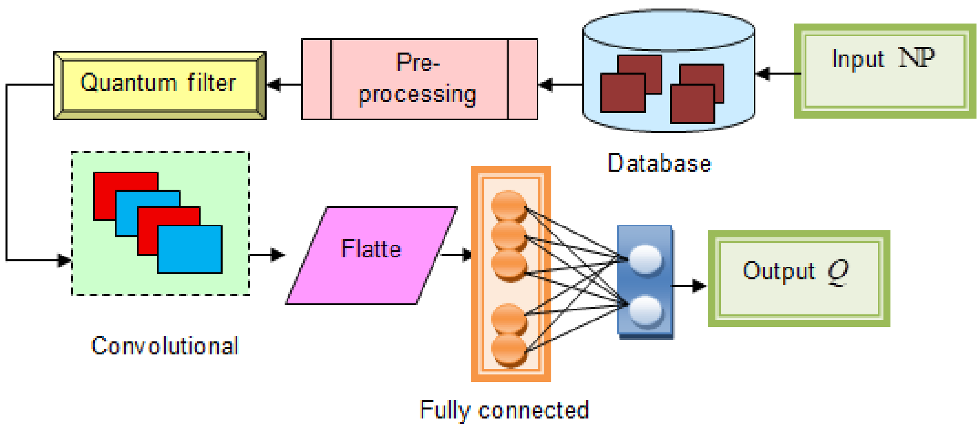


Fig 3: Illustrates Qucnet Structure

4.3.2. Tuning of QuCNet using EPCO

The tuning process is performed to optimize the hyperparameters of QuCNet classifier, which improves convergence speed, accuracy. This process is done using EPCO, which is designed by the integration of EWMA and PCOA. PCOA [11] is inspired by the diverse reproductive strategies of pine trees, including pollination and the dispersal of pine cones through gravity and animal behaviours. EWMA [10] is employed for smoothening time series data by applying exponentially decreasing weights to past observations. Integrating these two approaches, the developed algorithm EPCO can obtain better scalability and generalizability. The following stages are the derivative steps of EPCO algorithm, which is detailed as,

4.3.2.1. Initialization:

It contains two different populations namely, pine trees and their cones. The population of pine cone is represented as,

$$PC_{\ell,\lambda} = \Psi_{L_{b(\lambda)}} + \overrightarrow{rand}_{1\times\dim} \times \left(rand \times \Psi_{U_{b(\lambda)}} - rand \times \Psi_{L_{b(\lambda)}}\right) \quad (17)$$

Here, Ψ_{U_b} and Ψ_{L_b} elucidates lower and upper boundaries of all segments, PC indicates pine cone's position, $\overrightarrow{rand}_{1\times\dim}$ implies random vector with normal distribution among $[0,1]$, $rand$ specifies random value with normal distribution among $[0,1]$, λ elucidates pine tree index and ℓ expresses cone index.

4.3.2.2. Compute fitness:

It measures the effectiveness of a candidate solution in achieving the desired outcomes of the optimization problem, which is formulated based on min–max method as,

$$fit = \frac{\overline{fit}_{\ell} - \min\left(\overline{fit}_{\ell}\right)}{\max\left(\overline{fit}_{\ell}\right) - \min\left(\overline{fit}_{\ell}\right)} \tag{18}$$

Here, $\overline{fit}_{\ell}$ elucidates average fitness value for ℓ^{th} parameter.

4.3.2.2. Pine cone dispersal by gravity:
To simulate the consequence of gravitational forces on cone dispersion in the region of trees, it is important to produce numerous solutions around all trees. It is expressed as,

$$PC_{\ell,\lambda}^{new} = \begin{cases} PT_{\lambda} + \psi_1 \times \aleph_1 \times \left(\aleph_2 \times \left(U_{bu}^{\lambda} - L_{bu}^{\lambda} - PT_{\lambda}\right)\right), & if\ control_{parameter} = 0 \\ PC_{\ell,\lambda} + \psi_1 \times \aleph_1 \times \left(\aleph_2 \times \left(U_{bu}^{\lambda} - L_{bu}^{\lambda} - Opop_{all,v_1}\right) - O_{best_{m,\lambda}}\right), & otherwise \end{cases} \tag{19}$$

Here, $PC_{\ell,\lambda}^{new}$ elucidates pine cone's new position, $PC_{\ell,\lambda}$ indicates pine cone's updated solution, PT_{λ} expresses tree's position, U_{bu}^{λ} and L_{bu}^{λ} enumerates super-cube's upper and lower boundaries, $Opop_{all,v_1}$ implies randomly chosen solution, $O_{best_{m,\lambda}}$ represents λ^{th} top solution, ψ_1 signifies adaptive weight, v_1 specifies random value, $\aleph_1, \aleph_2$ symbolizes random values between 0 and 1. λ and ℓ depicts pine and cone index.

4.3.2.3. Pollination:
The agents are generated from male cones and drift to female cones, where it may occur either between cones on different trees or between cones on same tree. Here, the wind pollination is simulated by employing,

$$PC_{\ell,\lambda}^{new} = PC_{\ell,\lambda} + 0.5 \times \varphi_{v_1} \times \left(O_{best_{\ell}} - PC_{v_1,\lambda}\right) + 0.5 \times \varphi_{v_3} \times \left(O_{best_{\ell}} - PC_{v_3,\lambda}\right) \tag{20}$$

Here, φ enumerates chance of successful pollination of pine cones. Assume, $PC_{\ell,\lambda}^{new} = J_{\ell,\lambda}(d+1)$, $PC_{\ell,\lambda} = J_{\ell,\lambda}(d)$, $PC_{v_1,\lambda} = J_{v_1,\lambda}(d)$ and $PC_{v_3,\lambda} = J_{v_3,\lambda}(d)$. Then, substitute these values in Eq. (20), then it is computed as,

$$J_{\ell,\lambda}(d+1) = J_{\ell,\lambda}(d) + 0.5 \times \varphi_{v_1} \times \left(O_{best_{\ell}} - J_{v_1,\lambda}(d)\right) + 0.5 \times \varphi_{v_3} \times \left(O_{best_{\ell}} - J_{v_3,\lambda}(d)\right) \tag{21}$$

The update solution from EWMA should be integrated with the above expression to achieve better stability. Thus, it is computed as,

$$J_{\ell,\lambda}{}^{E}(d) = \chi \cdot J_{\ell,\lambda}(d) + (1-\chi) J_{\ell,\lambda}{}^{E}(d-1) \tag{22}$$

$$J_{\ell,\lambda}(d) = \frac{1}{\chi}\left[J_{\ell,\lambda}{}^{E}(d) - (1-\chi) J_{\ell,\lambda}{}^{E}(d-1)\right] \tag{23}$$

Apply Eq. (23) in Eq. (21), then it expresses,

$$\begin{aligned} J_{\ell,\lambda}(d+1) = &\left[\frac{1}{\chi}\left[J_{\ell,\lambda}{}^{E}(d) - (1-\chi) J_{\ell,\lambda}{}^{E}(d-1)\right]\right] + 0.5 \times \varphi_{v_1} \times \left(O_{best_{\ell}} - J_{v_1,\lambda}(d)\right) \\ &+ 0.5 \times \varphi_{v_3} \times \left(O_{best_{\ell}} - J_{v_3,\lambda}(d)\right) \end{aligned} \tag{24}$$

Here, χ implies constant and $J_{\ell,\lambda}(d+1)$ elucidates updated position.

4.3.2.4. Pine Cone Dispersal by Animals:
Animals eat and disperse pine cones; this behavior is simulated using four operators and optimized by quadratic programming for better solutions. The initial point $L_{initial}$ is computed as,

$$L_{initial} = L_{best} + rand \times \left(\overline{PC} - L_{best}\right)$$

Here, $\overline{PC}$ elucidates mean of the positive of each pine cones. The second operator is determined as,

$$L_{animal} = \frac{L_{best} + L_{animal}}{2} + Levy \times \left(Levy \times \left(L_b + U_b - \frac{L_{best} + L_{animal}}{2}\right) - \frac{L_{best} + L_{animal}}{2}\right) \tag{26}$$

Here, L_{animal} implies position of cones carried by animals, $Levy$ expresses Levy distribution. The third process is determined as,

$$L_{animal} = PC + (1 - z_{\tau}) \times \overline{PT} + z_{\tau} \times Levy \times \left(Levy \times \left(L_b + U_b - \overline{PT}\right) - \overline{Tree_m}\right) \tag{27}$$

Here, $\overline{PT}$ expresses mean position of pine trees and z_{τ} signifies adaptive weight. The fourth operators is determined as,

$$L_{animal} = PC + z_{\tau} \times Levy \times \left(Levy \times \left(L_b + U_b - PT\right) - PT\right) \tag{28}$$

4.3.2.5. Re-evaluate fitness:
This fitness value should be performed to eliminate error present in the solution.

4.3.2.6. Termination:
After several iterations, above-mentioned process will be terminated.

5. Results and Discussions

The EPCO_QuCNet evaluation is performed in the beneath sections, comparing with the prior techniques.

5.1. Experimental setup

EPCO_QuCNet is implemented using PYTHON.

5.2. Performance Measures

The metrics employed for EPCO_QuCNet are described below:

5.2.1. Acceptance Rate

It refers to the ratio of network slice requests that are efficiently admitted and allocated resources by 5G network.

5.2.2. Resource Efficiency

It measures how effectively a 5G network utilizes its available physical and virtual resources across different network slices.

5.2.3. Execution time

It is the duration needed to complete the process of network slice selection within 5G network.

5.3. Comparative Methods

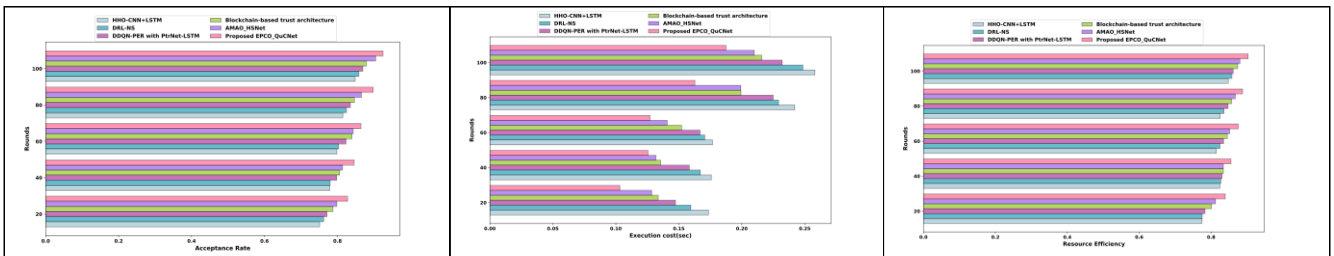
HHO-CNN+LSTM [4], DRL-NS [5], DDQN-PER with PtrNet-LSTM [6], Blockchain-based trust architecture [7] and AMAO_HSNet are the prior techniques of EPCO_QuCNet.

5.4. Comparative Analysis

EPCO_QuCNet evaluation is carried out by means of slice requests 10 and 20 based on rounds.

5.4.1. Assessment of EPCO_QuCNet for slice requests=10

EPCO_QuCNet analysis by altering rounds is portrayed in figure 4. Here, number of rounds is considered as 100. Assessment of EPCO_QuCNet based on acceptance rate is enumerated in figure 4 a). Acceptance rate achieved by EPCO_QuCNet is 0.925, while other techniques obtained acceptance rate of 0.849, 0.859, 0.870, 0.879 and 0.905. Figure 4 b) shows EPCO_QuCNet assessment regarding execution time. Conventional modules attained execution time of 0.903 sec where other techniques acquired execution time of 0.258 sec, 0.249 sec, 0.232 sec, 0.216 sec and 0.210 sec. EPCO_QuCNet Evaluation in regards of resource efficiency is displayed in figure 4 c). Traditional approaches gained resource efficiency of 0.848, 0.858, 0.861, 0.874 and 0.880 whereas EPCO_QuCNet attained resource efficiency of 0.187.

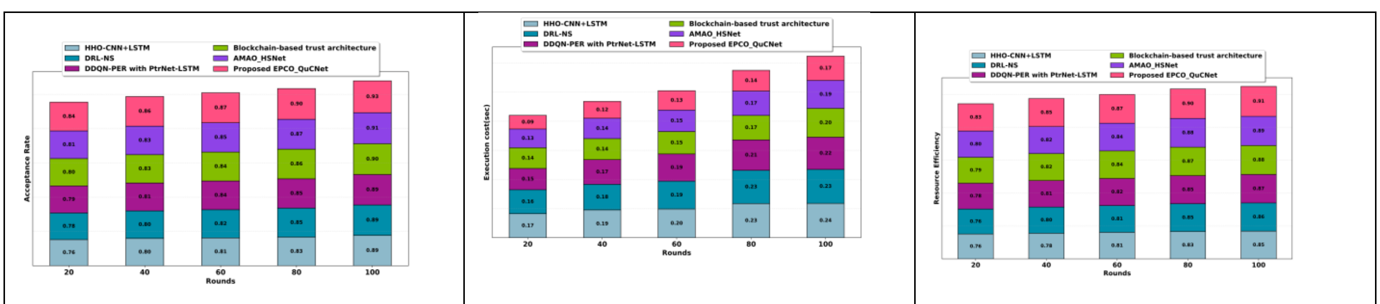


(a) (b) (c)

Fig 4. EPCO_QuCNet evaluation based on slice requests=10, a) Acceptance rate, b) Execution time, c) Resource efficiency

5.4.2. Assessment of EPCO_QuCNet for slice requests=20

Figure 5 demonstrates EPCO_QuCNet assessment by altering rounds. Here, number of rounds is taken as 100. EPCO_QuCNet analysis with respect to acceptance rate is demonstrated in figure 5 a). Prior techniques achieved acceptance rate of 0.886, 0.887, 0.888, 0.898 and 0.907 whereas EPCO_QuCNet attained acceptance rate of 0.927. Assessment of EPCO_QuCNet in accordance of execution time is elucidated in figure 5 b). Execution time obtained by EPCO_QuCNet is 0.167 sec where former methods obtained execution time of 0.236 sec, 0.233 sec, 0.222 sec, 0.197 sec and 0.192 sec. Figure 5 c) enumerates EPCO_QuCNet analysis regarding resource efficiency. Preceding approaches attained resource efficiency of 0.911 where existing approaches acquired resource efficiency of 0.846, 0.856, 0.867, 0.879 and 0.889.



(a) (b) (c)

Fig 5. EPCO_Qucnet Estimation Based on Slice Requests=20, A) Acceptance Rate, B) Execution Time, C) Resource Efficiency

5.5. Comparative Discussions

Table 1 illustrates comparative discussions of EPCO_QuCNet. From this discussion, it shows that the EPCO_QuCNet acquired acceptance rate of 0.927, less execution time of 0.167 and high resource efficiency of 0.911. The high acceptance rate shows that EPCO_QuCNet has the ability to accommodate incoming service demands without rejection due to resource limitations. Moreover, minimal execution time represents that how quickly the EPCO_QuCNet can respond to slice requests and adapt to changing service demands. In addition to that, higher resource efficiency indicates optimal usage, minimizing waste while meeting service requirements. Therefore, this overall assessment indicates that EPCO_QuCNet has higher efficiency, reliability and adaptability.

Table 1: Comparative discussion

Setups	Metrics/ Methods	HHO-CNN+LSTM	DRL-NS	DDQN-PER with PtrNet-LSTM	Blockchain-based trust architecture	AMAO_HSNet	Proposed EPCO_QuCNet
Number of slice requests=10	Acceptance rate	0.849	0.859	0.870	0.879	0.905	0.925
	Execution time (sec)	0.258	0.249	0.232	0.216	0.210	0.187
	Resource efficiency	0.848	0.858	0.861	0.874	0.880	0.903
Number of slice requests=20	Acceptance rate	0.886	0.887	0.888	0.898	0.907	0.927
	Execution time (sec)	0.236	0.233	0.222	0.197	0.192	0.167
	Resource efficiency	0.846	0.856	0.867	0.879	0.889	0.911

6. Conclusion

Network slicing enables virtual networks on distributed physical system to obtain diverse requirements of various applications and industries. It allows tailored services for use cases, like autonomous vehicles, and smart cities. Nonetheless, managing and optimizing these slices remains a significant complexity due to dynamic network conditions, resource constraints, and the need for strict QoS, posing a difficult issue for network operators. As a result, EPCO_QuCNet is introduced for secure network slicing in 5G network to conquer these issues. Initially, the systematic module of secure 5G network is carried out. When a network slicing receives requests from UE, certain network parameters are collected from multiple devices. Thus, a secure network slicing process is efficiently performed by employing EPCO_QuCNet. Furthermore, the services provided in 5G are accessed by internet service provider by employing VNF. EPCO_QuCNet has acquired acceptance rate, execution time and resource efficiency of 0.927, 0.167 sec and 0.911. In future, the developed will focus on AI-based techniques for dynamic slice management, strengthening slice isolation and security, and tackling scalability issues in large-scale 5G network environments.

Reference

[1] R. F. Olimid and G. Nencioni, "5G network slicing: A security overview," Ieee Access, vol. 8, pp. 99999-100009, 2020.

[2] K. Abbas, M. Afaq, T. A. Khan, A. Mehmood and . W.-C. Song, "BNSlicing: Intent-based network slicing framework for 5G networks using deep learning," In proceedings of 2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 19-24, September 2020.

[3] A. R. Nasser and O. Y. Alani, "Investigation of Multiple Hybrid Deep Learning Models for Accurate and Optimized Network Slicing," Computers, vol. 14, no. 5, p. 174, 2025.

[4] R. Dangi and P. Lalwani, "Harris Hawks optimization-based hybrid deep learning model for efficient network slicing in 5G network," Cluster Computing, vol. 27, no. 1, pp. 395-409, 2024.

[5] 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.

[6] Z. Z. Saleh, M. F. Abbod and R. Nilavalan, "Intelligent Resource Allocation via Hybrid Reinforcement Learning in 5G Network Slicing," IEEE Access, vol. 13, pp. 47440-47458, 2025.

[7] S. B. Saad, A. Ksentini and B. Brik, "An end-to-end trusted architecture for network slicing in 5G and beyond networks," Security and Privacy, vol. 5, no. 1, p. e186, 2022.

[8] D. K. Sanyal, U. N. Kar and M. Roy, "Mobile communications and computing: A broad review with a

focus on smart healthcare," Smart Healthcare Analytics in IoT Enabled Environment, pp. 9-33, 2020.

[9] G. Swathi, A. Altalbe and . R. P. Kumar, "QuCNet: quantum-inspired convolutional neural networks for optimized thyroid nodule classification," IEEE Access, 2024.

[10] W. Febrina and W. Fitriana, "Exponential weight moving average (EWMA) control chart for quality control of crude palm oil product," International Journal of Management and Business Applied, vol. 1, no. 1, pp. 19-27, 2022.

[11] M. V. Anaraki and S. Farzin, "The Pine Cone Optimization Algorithm (PCOA)," Biomimetics, vol. 9, no. 2, p. 91, 2024.

[12] W. Guan, X. Wen, L. Wang, Z. Lu and Y. Shen, "A service-oriented deployment policy of end-to-end network slicing based on complex network theory," IEEE access, vol. 6, pp. 19691-19701, 2018.

[13] M. H. Abidi, H. Alkhalefah, K. Moiduddin, M. Alazab, M. K. Mohammed, W. Ameen and T. R. Gadekall, "Optimal 5G network slicing using machine learning and deep learning concepts," Computer Standards & Interfaces, vol. 76, p. 103518, 2021.