

# MODIFIED O-RAN 5G EDGE REFERENCE ARCHITECTURE USING RNN

M.V.S Phani Narasimham<sup>1</sup> and Y.V.S Sai Pragathi<sup>2</sup>

<sup>1</sup>Senior Architect, Wipro Technologies, Hyderabad, India

<sup>2</sup>Department of Computer Science & Engineering, Stanley College of Engineering & Technology for Women (Autonomous), Hyderabad, India

## **ABSTRACT**

*This paper explores the implementation of 6G/5G standards by network providers using cloud-native technologies such as Kubernetes. The primary focus is on proposing algorithms to improve the quality of user parameters for advanced networks like car as cloud and automated guided vehicle. The study involves a survey of AI algorithm modifications suggested by researchers to enhance the 5G and 6G core. Additionally, the paper introduces a modified edge architecture that seamlessly integrates the RNN technologies into O-RAN, aiming to provide end users with optimal performance experiences. The authors propose a selection of cutting-edge technologies to facilitate easy implementation of these modifications by developers.*

## **KEYWORDS**

*5G O-RAN, 5G-Core, AI Modelling, RNN, Tensor Flow, MEC Host, Edge Applications.*

## **1. INTRODUCTION**

This paper integrates artificial intelligence (AI) into 5G, by estimating the end to end delay of video streaming and reducing the risks associated with latency and network failures. The paper illustrates the fundamental components of 5G networks deployed at the edge. Cloud-native application architecture is employed for 5G components, indicating a flexible and scalable approach to building and deploying applications.

AI is leveraged by both 5G network vendors and users to achieve optimal user experience at the edge. The aim is to mitigate latency and minimize the risk of network failures and improve the user quality of experience. Key parameters for AI in the 5G context include end to end delay, network slicing, and air-quality parameters. These parameters play a crucial role in optimizing network performance and user experiences.

The paper introduces AI Modelers, which are responsible for processing and analysing data. This work explores real time AI models employing Recurrent Neural Networks (RNNs) and Long Short-Term memory models. The incorporation of AI, especially with advanced models like RNN and LSTM optimizes 5G networks at the edge.

This paper encompasses seven section, with its outline and structure described in the following paragraph.

Section1: Introduction: Emphasizes the high demand for the application of AI in 5G networks by networking companies. Highlights the modifications made to the 5G reference architecture to incorporate AI models, reduce latency, and meet new 6G standards.

Section 2: Related Work: This section conducts a comprehensive review of AI related publications in 5G networks, there by establishing a foundation for AI-centric enhancements presented in this work.

Section 3: 5G Cloud Native Architecture: Explores the cloud-native architecture adopted for 5G networks, indicating a modern and scalable approach to network design.

Section 4: Applying AI Models (ANN, LSTM, RNN) to 5G: Describes the integration of AI models, including Artificial Neural Networks (ANN), Long Short-Term Memory (LSTM), and Deep-Q, into the 5G network.

Section 5: Modified Reference Architecture: Proposes modifications to the 5G reference architecture, showcasing how AI integration, latency reduction, and air quality improvement are implemented.

Section 6: Simulation Results: Discusses the results of simulations conducted to validate the effectiveness of the proposed modifications.

Section 7: Conclusion: Summarizes the key findings and contributions of the paper. Concludes the paper by addressing challenges and proposing an AI framework that integrates Graphics Processing Unit (GPU) and Field-Programmable Gate Array (FPGA)-based edge pods for achieving real-time performance for 6G/5G O-RANs.

## **2. RELATED WORK**

Hai T. Nguyen et.al in their paper, “ Scaling UPF Instances in 5G/6G Core with Deep Reinforcement Learning”, have applied AI techniques - deep reinforcement learning, actor-critic based policy optimization algorithm to optimize the creation and deletion of UPF pods based on the pdu session traffic. The simulation results shows that proposed AI algorithm gives better performance compared to the stands k8s horizontal scaling algorithm [1]. In their research paper titled "Research and Applications of AI in 5G network Operation and maintenance," Mingxin Li et.al have put forth an AI-driven framework aimed at tackling challenges inherent to 5G networks. This proposed framework seeks to address issues related to network failures, security vulnerabilities, and optimal resource allocation within the context of 5G telecommunications systems [2].

Manuel E. M. Cayamcela et.al in their paper, “Artificial Intelligence in 5G Technology: A Survey”, has suggested using supervised learning, unsupervised learning and reinforcement learning for improving the network performance. Edge caching techniques are suggested allowing base station to predict what content user uses in the near future.[3]. Amit S et.al in their paper, “AI-Driven Provisioning in the 5G core”, has proposed usage of AI in network slicing of different traffic categories like enhanced mobile broad band (ex: virtual reality), massive machine type communications ( ex: IOT ), ultrareliable low-latency communications ( ex: Autonomous Driving) [4]. Muhammed U et.al in their paper, “Examining Machine Learning for 5G and Beyond Through an Adversarial Lens” have discussed solutions for security attaches when AI/ML is applied on 5G network [5]. Bharath B et.al in their paper, “A RAN Intelligent Controller Platform for AI-Enabled cellular Networks” has proposed RAN intelligent controller

(RIC) that decouples the control and data planes of RAN. AI is integrated to RIC and concluded that OS context switching and virtualization overheads reason for latency in the RIC [6].

Xian S et.al in their paper, “Intelligent and Scalable Air Quality Monitoring with 5G Edge”, has suggested AI-powered air quality monitoring. This research paper presents an innovative architecture that harnesses the capabilities of 5G edge computing infrastructure, artificial intelligence methodologies, and large-scale deployment of devices across urban environments to facilitate real-time monitoring and analysis of air quality conditions [7]. Yaohua Sun et.al in their paper, “Applications of Machine Learning in Wireless Networks: Key Techniques and Open Issues”, have discussed open data sets and platform for developers, AI based network slicing, infrastructure update to support AI algorithms [8].

Xiaohu Y et.al in their paper, “AI for 5G: Research Directions and Paradigms” have proposed usage of AI for baseband signal processing, AI for downlink resource allocation in 5G NR, usage of AI in automatic root cause analysis for 5G network [9]. Shiyu Z et.al in their paper, “5G Core Network Framework Based on Artificial Intelligence”, have proposed automatic resource prediction of computing, storage and network using the traffic data. AI model will use the flow forecasting, festivals and other factors are included to get better traffic prediction [10].

L. P. Kaelbling in their paper, “Reinforcement Learning: A Survey” has analysed different reinforcement learning and proposed improvements like shaping, local reinforcement, and problem decomposition [11]. In their research paper titled "Deep Reinforcement Learning: A Brief Survey," K. Arulkumaran et al. explored value-based and policy-based agents within the context of deep reinforcement learning. Their work explored various techniques, including Deep Q-networks, trust region policy optimization, and hybrid actor-critic neural network architectures, which are utilized for training policy models in reinforcement learning applications [12]. In their paper titled "An Introduction to Deep Reinforcement Learning," Vincent F.L. explored the fundamental concepts of reinforcement learning (RL), including value-based methods, policy gradient techniques, model-based approaches, and the application of deep learning algorithms to partially observable Markov decision process (POMDP) environments [13].

ANFIS modelling is used to solve the non-linear problems. An adaptive neuro-fuzzy inference system (ANFIS) is a kind of artificial neural network that is based on Takagi–Sugeno fuzzy inference system. It integrates both neural networks and fuzzy logic principles and captures the benefits of both in a single framework [14]. In their research paper titled "A Neural Networks Based Approach for the Real-Time Scheduling of Reconfigurable Embedded Systems with Minimization of Power Consumption," Ghofrane Rehaïem, Hamza Gharsellaoui, and Samir Ben Ahmed have employed an artificial neural network (ANN) technique based on backpropagation to model real-time task scheduling for embedded systems. Their approach aims to minimize power consumption by optimizing the scheduling process, thereby achieving a cost-effective solution [15].

An artificial neural network (ANN) architecture, tuned with an optimized number of input, output, and hidden neurons, demonstrates the capability to effectively identify and accurately classify intricate patterns or regions within the data. S.Agatanovic-kustrin, R. Beresford, “Basic concepts of artificial neural network ( ANN) modelling and its application in pharmaceutical research” has applied ANN to find optimal dosage of the drug and analysing chromatographic data[16]. In their research paper titled "Development of Realistic Models of Oil Well by Modelling Porosity Using Modified ANFIS Technique," M.V.S Phani Narasimham et.al introduced an artificial neural network (ANN)-based neutron porosity model. This model aimed to develop reliable static reservoir representations crucial for

simulating and exploring oil well operations [17]. Their paper leverages a modified adaptive neuro-fuzzy inference system (ANFIS) technique to enhance the modelling of porosity, a critical parameter in oil reservoir characterization.

Richard S. Sutton et al in their paper, “Policy gradient methods for reinforcement learning with function approximation”, have proposed actor critic method for function approximation [18]. Scott F et. al in their paper, “Addressing Function Approximation error in Actor-Critic Methods” have used Double-Q learning to reduce function approximation errors and to find optimal policy agents [19].

Debaditya S et.al in their paper, “Deep Q-learning for 5G network slicing with diverse resource stipulations and dynamic data traffic” proposes Deep-Q technique for network slicing. Algorithm uses sigmoid transform Quality of experience, price satisfaction and spectral efficiency as the reward function for UPF selection [20].

Giovanni N et al in their paper, “Simu5G: A System-level Simulator for 5G Networks”, have validated 5G MEC scenario have shown the latency improved significantly in a migration scenario compared to the non-migration scenario. With  $\mu=3$  the latency decreases due to lesser TTI at the MAC layer [22].

### 3. 5G CLOUD NATIVE ARCHITECTURE

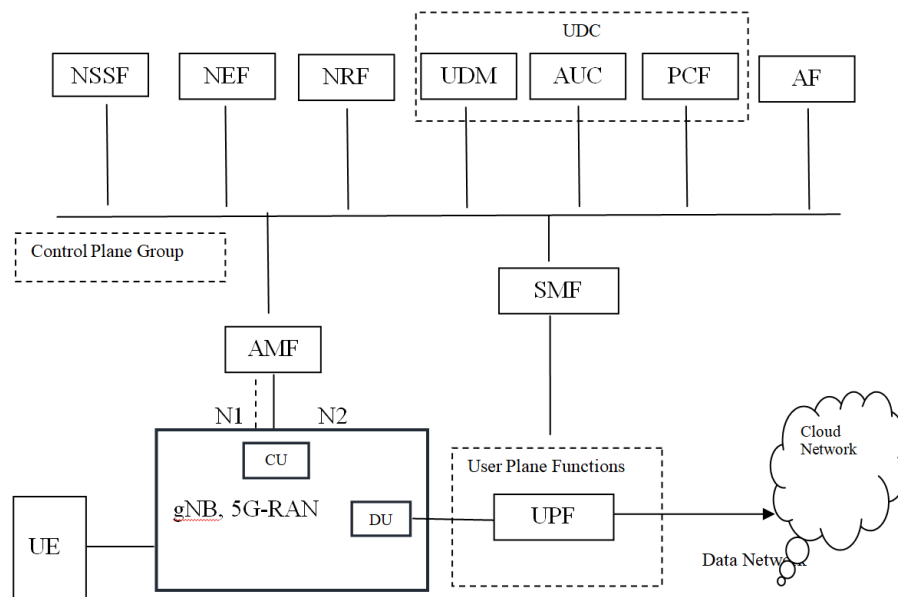


Figure -1 5G Integrated Basic Blocks

**AMF:** AMF provides access and mobility management functions. It implements the features of registration management, connection management, reachability management, mobility management, access authentication, access authorization and UE mobility event notification.

**SMF:** SMF stands for session management function. IT implements the 5G features of session management, UE IP address allocation, Configuring traffic steering at UPF, policy enforcements.

**UPF:** UPF stands for user plane functions. UPF implements the 5G features packet routing & forwarding, PDU session interaction, Uplink traffic verification and downlink packet buffering.

**PCF:** PCF stands for policy control function. It supports applying policy decisions, access of subscription information and govern the network behaviour.

**UDM:** Unified data management. It performs operations like user identification, subscription management, user authentication and access authorization.

**NRF:** Network repository function. This component is responsible for maintaining an up-to-date inventory of available network function instances and their corresponding profiles. Additionally, it facilitates service registration and discovery processes, enabling different network functions to locate and communicate with one another through the utilization of RESTful Application Programming Interfaces (APIs).

**AF:** Application function. This component exposes the network exposure function, allowing for the retrieval of resources and enabling seamless interaction with the policy control function (PCF) for the purpose of implementing and managing policy control mechanisms within the network.

**gNB:** The gNB, or gNodeB, is a critical component within the 5G RAN, is a 5G base station and performs Radio Signal Processing. The gNB processes radio signals, including modulation, demodulation, and encoding, for wireless communication with user devices. It supports low latency communication for autonomous vehicles, virtual reality. It also supports massive connectivity suitable for internet of things.

**CU:** The CU is a central component of the RAN. It helps in optimizing network resources, ensuring efficient utilization of the available spectrum, and providing seamless handovers as devices move through different coverage areas. The CU is also involved in functions related to mobility management, security, and connection setup. The CU is responsible for functions such as radio resource management, scheduling, and coordination between different DUs.

**DU:** The distributed unit, strategically positioned at the edge of the network in close proximity to radio antennas or base station sites, assumes the responsibility of radio signal processing functions. These functions encompass tasks such as digital signal processing, radio modulation and demodulation, beamforming techniques, and channel coding algorithms. The distributed units bear the onus of facilitating real-time processing of radio signals, thereby playing an indispensable role in minimizing latency and improving the overall efficiency of the radio communication. In a 5G RAN, DUs work closely with the CU, which is typically located at a centralized data centre or network core.

**NSSF:** Network slice selection function. It implements the 5G features of selecting set of Network slice. It determines the AMF to authorize the user in the network.

### **5G- RAN:**

Architecture has two major blocks Radio Block, Management Block.

- Radio block consists of Near-RT RIC, O-CU-CP, O-CU-UP, O-DU, and O-RU components
- Near-RT RIC enables near-real-time control and optimization via the E2 interface
- O-CU handles RRC, SDAP, and PDCP protocols (control plane in O-CU-CP, user plane in O-CU-UP)

- O-DU hosts RLC/MAC/Higher PHY layers
- O-RU hosts Lower PHY layer and RF processing

Management Side

- Non-RT RIC enables non-real-time control and optimization.

xAPPs improves Near-RT RIC to extend RAN functionality.

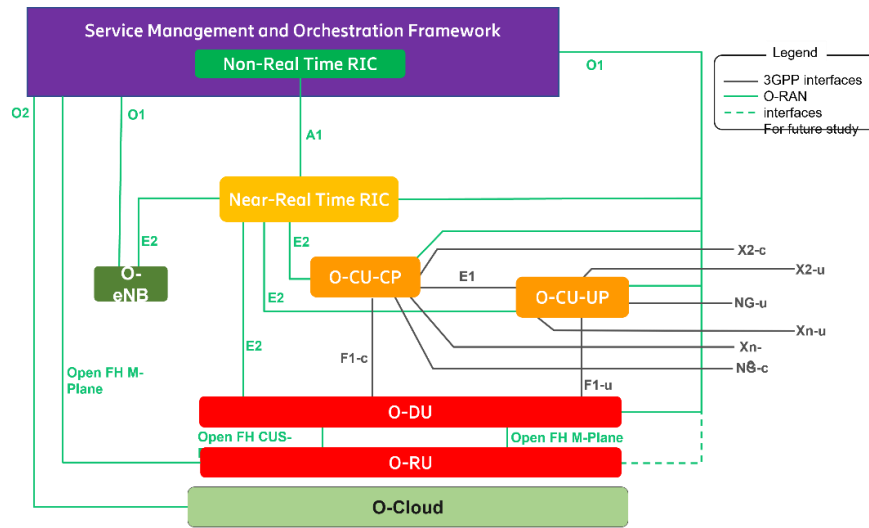


Figure 2: 5G O-RAN Components

#### 4. AI MODELS – RNN

Unlike conventional artificial neural networks, recurrent neural networks (RNNs) incorporate feedback loops that enable it to recursively propagate information from one stage of the network to the next. This unique feature endows RNNs with the ability to maintain an internal state or memory of previous inputs within a given sequence, setting them apart from their feedforward counterparts. This memory of previous inputs make RNNs suited for modelling time series data and natural language processing.

RNNs have vanishing gradient problem, where gradients during training become very small making it to difficult to model long range dependencies. Long Short-Term Memory (LSTM) is a type of recurrent network ( RNN ) which is designed to overcome vanishing gradient problem. LSTM are highly deployed to model time series data like stock prices, weather patterns and IOT sensor data. Isha V et.al in their paper, “Stock Market prediction using LSTM” have used LSTM to predict the stock prices for the next 30 days [23].

LSTM uses Ct component which serves capturing long term memory information over long sequences. LSTM uses ht component short-term memory that contains the recent information required of the model.

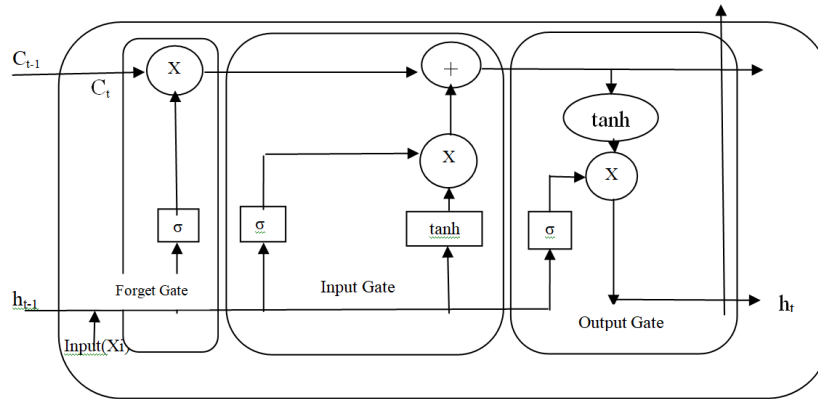


Figure 3: LSTM Basic Element

**Input Gate:** Permits relevant information from the current cell state for the current input. Determines necessary information for the current input using the sigmoid activation function. Stores relevant information in the current cell state. Utilizes the tanh activation mechanism to compute vector representations of input-gate values, which are added to the cell state.

**Forget Gate:** Decides what information from the previous state should be discarded or forgotten. Involves pointwise multiplication with its weight matrix and sigmoid activation. Generates probability scores to determine the relevance of information. This gate determines what is useful and what is not needed.

**Output Gate:** Determines what to output from the current cell state. Utilizes a weight matrix for input tokens and the output from the previous hidden state. Involves pointwise multiplication with sigmoid activation to obtain probability scores. Multiplies the sigmoid scores with the updated cell state. Applies a final tanh multiplication to ensure the output sequence values range from [-1,1].

## 5. INTEGRATED 5G REFERENCE ARCHITECTURE

Existing 5G O-RAN implementation is modified to use gRPC based microservices are containerized as docker containers. To evaluate the proposed reference architecture, the testing process employs Kubernetes, a container orchestration platform, augmented with custom scheduling capabilities specifically tailored for utilizing spot instances, which are temporary and cost-effective computing resources offered by cloud service providers.

### 5.1. RNN Based Real Time Cloud Architecture

ANN Supervised learning model with backpropagation, hybrid algorithms are explored to develop optimal ANN based cloud reference model. ANN perceptron network with input layer, hidden layer and output layer classifies tasks.

An incoming batch of simulator tasks representing a real-time scenario undergoes classification based on the task features extracted from the training batches. The training batches assess various characteristics of virtual machine (VM) containers, including CPU speed, GPU capabilities, memory intensity, and generic resource requirements. The artificial neural network (ANN) classifier leverages this information to identify the most suitable containers for executing the tasks within the given batch, optimizing resource allocation and performance.

Class 1 containers are characterized by their CPU-intensive nature, leveraging multiple threads to maximize computational power.

Class 2 containers are GPU-intensive, harnessing the parallel processing capabilities of graphics processing units (GPUs) through the utilization of CUDA and OpenCL frameworks.

Class 3 containers, which are designated as memory-intensive, placing substantial demands on system memory resources.

Class 4 category that encompasses generic containers, which do not have specific resource-intensive characteristics and can accommodate a wide range of workloads.

## 5.2. RNN Modelling Algorithm

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

# Load data from simu5G
data = pd.read_csv('output.csv', index_col=0)

# Preprocess data
#scaler = MinMaxScaler()
#data_scaled = scaler.fit_transform(data)
data_scaled = data.astype(float)

# Split data into input (X) and output (y) variables
X = data_scaled.iloc[:, :-1] # All features except first and last columns
y = data_scaled.iloc[:, -1].values.reshape(-1, 1) # Last column (end-to-end delay)

# Split data into train and test sets
train_size = int(len(data_scaled) * 0.8)
X_train, X_test = X[:train_size], X[train_size:]
y_train, y_test = y[:train_size], y[train_size:]

# Calculate the number of time steps based on the number of features
n_features = X_train.shape[1]
time_steps = 2 # Adjust this value based on your data

while n_features % time_steps != 0:
    time_steps += 1

n_steps = n_features // time_steps

if n_features % time_steps != 0:
    print(n_features, time_steps)
    raise ValueError("Number of features is not divisible by the number of time steps")
```



```

print(n_features, time_steps, n_steps)
# Reshape data for RNN input
print(X_train)
X_train = np.reshape(X_train.values, (X_train.shape[0], time_steps, n_steps))
X_test = np.reshape(X_test.values, (X_test.shape[0], time_steps, n_steps))

# Define the LSTM model
model = Sequential()
model.add(LSTM(64, input_shape=(time_steps, X_train.shape[2])))
model.add(Dense(1)) # Output layer with 1 unit for end-to-end delay

# Compile the model
model.compile(optimizer='adam', loss='mse')

# Train the model
model.fit(X_train, y_train, epochs=100, batch_size=32, validation_data=(X_test, y_test),
verbose=1)

# Evaluate the model
score = model.evaluate(X_test, y_test, verbose=0)
print('Test loss:', score)

# Make predictions
y_pred = model.predict(X_test)

# Inverse transform predictions and actual values for interpretation
# y_pred = scaler.inverse_transform(y_pred)
# y_test = scaler.inverse_transform(y_test)

# Print sample predictions
print('Sample predictions:')
for i in range(10,100):
    print(f'Actual: {y_test[i]}, Predicted: {y_pred[i]}')

# Save the actual and predicted values to a CSV file
results = pd.DataFrame({'Actual': y_test.flatten(), 'Predicted': y_pred.flatten()})
results.to_csv('predictions.csv', index=False)

print("Data saved to 'predictions.csv'")

# Plot actual vs. predicted values
plt.figure(figsize=(10, 6))
plt.plot(y_test[:100], label='Actual', marker='o')
plt.plot(y_pred[:100], label='Predicted', marker='x')
plt.xlabel('Index')
plt.ylabel('Value')
plt.title('Actual vs. Predicted Values')
plt.legend()
plt.savefig('act_vs_pred.png')
print("Figure saved ")

```

## 6. RESULTS

Graph showing the predicted and actual end to end delay results of edge app video streaming. Execution snapshot of the algorithm.

```

Epoch 94/100
148/148 [=====] - 0s 3ms/step - loss: 0.0332 - val_loss: 0.0306
Epoch 95/100
148/148 [=====] - 0s 3ms/step - loss: 0.0332 - val_loss: 0.0306
Epoch 96/100
148/148 [=====] - 0s 3ms/step - loss: 0.0331 - val_loss: 0.0312
Epoch 97/100
148/148 [=====] - 0s 3ms/step - loss: 0.0330 - val_loss: 0.0306
Epoch 98/100
148/148 [=====] - 0s 3ms/step - loss: 0.0331 - val_loss: 0.0308
Epoch 99/100
148/148 [=====] - 0s 3ms/step - loss: 0.0330 - val_loss: 0.0321
Epoch 100/100
148/148 [=====] - 0s 3ms/step - loss: 0.0331 - val_loss: 0.0314
Test loss: 0.03141561150550842
Sample predictions:
Actual: [0.27724919], Predicted: [0.42337212]
Actual: [0.27763329], Predicted: [0.42259115]
Actual: [0.27801739], Predicted: [0.42181498]
Actual: [0.28646752], Predicted: [0.4226152]
Actual: [0.28800391], Predicted: [0.42340183]
Actual: [0.29722223], Predicted: [0.4234273]
Actual: [0.29727226], Predicted: [0.423439]
Actual: [0.29731247], Predicted: [0.4234443]
    
```

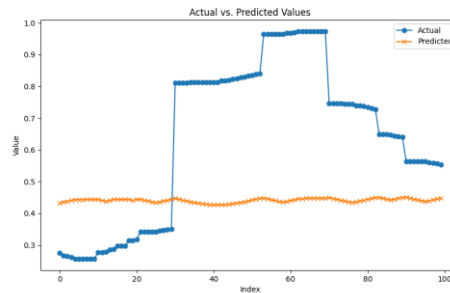


Figure 4: Actual vs Predicted values of End to End delay

Validation of simulations: Simu5G is used for the validation of simulation results using video streaming MEC application. Graph indicates results converging of the end to end delays the simulation progresses.

Additional gnb parameters apart from queue length, outgoing data rate will be considered to improve the results in future work.

## 7. CONCLUSION

As a culmination of this research endeavour, the paper has presented an innovative methodology for seamlessly integrating artificial intelligence (AI) techniques into the 5G Open Radio Access Network (O-RAN) architecture. The objective of this approach is to optimize the configuration parameters of the next-generation NodeB (gNB) component, thereby enhancing its performance and capabilities in supporting cutting-edge applications such as vehicle-based cloud computing (car-as-a-cloud) and automated guided vehicle (AGV) systems.

Our proposed modifications hold the promise of enhancing the performance and user experience of both 5G and future 6G networks. Throughout our research and implementation, we have resolved various obstacles, effectively clearing the path for forthcoming breakthroughs in artificial intelligence-driven telecommunications solutions. Additionally, we have proposed an AI framework incorporating GPU and FPGA-based edge pods to achieve real-time performance, thus providing a roadmap for the continued evolution of 5G networks.

Future area work would involve including 5G-core parameters into the algorithm and improving the performance of the 5G streams. Future research efforts will focus on exploiting the capabilities of deep learning accelerators, which are specialized hardware platforms designed to efficiently process and accelerate computationally intensive deep learning workloads. The integration of these accelerators aims to significantly improve the timing and performance of artificial intelligence algorithms, thereby enhancing the overall system's responsiveness and real-time capabilities.

## ACKNOWLEDGEMENTS

Thankful to Wipro Managers Nagarjuna Revenasiddappa, Ganeshan K, Rupesh K in supporting this research work. Grateful to Stanley College of Engineering & Technology for Women, Principal Dr. Satya Prasad Lanka, and the college management for their encouragement to explore new research areas.

## REFERENCES

- [1] Hai T. Nguyen, T.Van Do and C. Rotter, "Scaling UPF Instances in 5G/6G Core with Deep Reinforcement Learning", DOI 10.1109/ACCESS.2021.3135315, IEEE Access.
- [2] Mingxin Li, Mingde Huo, Xinzhou Cheng, Lexi Xu, "Research and Applications of AI in 5G Network Operation and Maintenance", 2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking, DOI 10.1109/ISPA-BDCloud-SocialCom-SustanCom51426.2020.00212.
- [3] Manuael E. M. Cayamcela, "Artificial Intelligence in 5G Technology: A Survey", 978-1-5386-5041-7, 2018,
- [4] Amit Sheroran, Sonia Fahmy, Lianjie Cao, Puneet Sharma, "AI-Driven Provisioning in the 5G Core", IEEE Internet Computing, March/April 2021, DOI 10.1109/MIC.2021.3056230.
- [5] Muhammed Usama, Inaam Ilahi, Junaid Quadir, Rupendra Nath Mitra and Mahesh K. Marina, "Examining Machine Learning for 5G and Beyond Through an Adversial Lens", IEEE Internet Computing, DOI 10.1109/MIC.2021.3049190.
- [6] Xiang Su, Xiaoli Li, Jacky Cao, Petri Pellikka, Yongchun Liu, Pan Hui, "Intelligent and Scalable Air Quality Monitoring with 5G Edge", IEEE internet Computing, DOI 10.1109/MIC.2021.3059189.
- [7] Bharath B, E. Scott Daniels, Matti Hiltunen, Ritwik Jana, Kaustubh Joshi, Tuyen X. Tran, Changwei Wang, Rajarajan Sivaraj, Mavenir, "RIC: A RAN Intelligent Controller Platform for AI-Enabled Cellular Networks", IEEE Internet Computing, DOI: 10.1109/MIC.2021.3062487.
- [8] Yaohua San, Mugen Peng, Yangcheng Zhou, Yuzhe Huang and Shiwen Mai, "Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues", IEEE Communications Survey & Tutorials, DOI 10.109/COMST.2019.2924243.
- [9] Shiya Zhou, Xiqing Liu, Dengsheng Fu, Zingzhou Cheng, Bin Fu, Zhenqiao Zhao, "5G Core Network Framework Based on Artificial Intelligence", IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking, 2020, DOI 10.1109/ISPA-BDCloud-SocialCom-SustanCom51426.2020.00219.

- [10] Xiaohu You, Chuan Zhang, Xiaosi Tan, Shi Jin and Hequan WU, "AI for 5G: Research Directions and Paradigms", Science China, Informatica Sciences, 23 July 2018.
- [11] Vincent F L, Peter H, Riashat I, Marc G B and Joelle P, "An introduction to Deep Reinforcement Learning", *Foundation and Trends in Machine Learning*: Vol. 11, No. 3-4, pp 219-354. DOI: 10.1561/22000000071.
- [12] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage and Anil Anthony Bharath, "Deep Reinforcement Learning, A brief survey", *IEEE Signa Processing Magazine*, 13 Nov 2017, DOI 10.1109/MSP.2017.2743240.
- [13] Vincent F L, Peter H, Riashat I, Marc G B and Joelle P, "An introduction to Deep Reinforcement Learning", *Foundation and Trends in Machine Learning*: Vol. 11, No. 3-4, pp 219-354. DOI: 10.1561/22000000071.
- [14] Jyh-Shing Roger Jang, "ANFIS:Adaptive-Network-Based Fuzzy Inference System", *IEEE Transactions on Systems Man and Cybernetics*, June 1993.
- [15] Ghofrane Rehaïem, Hamza Gharsellaoui, Samir Ben Ahmed, "A Neural Networks Based Approach for the Real-Time Scheduling of Reconfigurable Embedded Systems with Minimization of Power Consumption.", *ICIS Conference*, At Okayama, Japan, June 2016, 10.1109/ICIS.2016.7550777.
- [16] S Agantonovic-kustrin, R Beresford, "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research", *Journal of Pharmaceutical and Biomedical Analysis*", 22(2000) 717-727.
- [17] M V S Phani Narasimham, Dr Y V S Sai Pragathi, "Development of realistic models of oil well by modeling porosity using modified ANFIS technique", *International Journal on Computer Science and Engineering*, Vol.11, No.07, July 2019.
- [18] Richard S Sutton, David McAllester, Satinder Singh, Yishay Mansour, " Policy gradient methods for reinforcement learning with function approximation", *Proceedings of the 12th international conference on Neural Information Processing*, November 1999, PP 1057-1063.
- [19] Scott F, Herke V Hoof, David M, "Addressing Function Approximation Error in Actor-Critic Methods", *Proceesings of the 35th internaton conference on Machine Learning*, Stockholm, Sweden, PMLR 80, 2018.
- [20] Debaditya Shome, Ankit Kudeshia, "Deep Q-learning for 5G network slicing with diverse resource stipulations and dynamic data traffic", *IEEE Internation conference on Artificial intelligence in Information and Communication*, DOI 10.1109/ICAIIIC5149.2021.9415190.
- [21] Akhter Mohiuddin Rather, "LSTM-based Deep Learning Model for Stock Prediction and Predective Optimizaiton Model", Elsevier, *EURO Journal on decision processes*, vol 9, 2021, 100001.
- [22] Giovanni Nardini, Giovanni Stea, Antonio Verdis and Dario Sabella, "Simu5G: A system-lvel Simulator for 5G Networks", DOI: 10.5220/0009826400680080, In *Proceedings of the 10th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2020)*, pages 68-80.
- [23] Isha Venikar, Jaai Joshi, Harsh Jalnekar, Shital Raut, "Stock Market Prediction Using LSTM", *IJRASET47967*, 10.22214/ijraset.2022.47967, 2022-12-08.

## AUTHORS

**M.V.S Phani Narasimham, M.S(LL.Sc), B.E**, Senior Architect at Wipro Technologies, specializing in authoring solutions for Linux and embedded platform issues related to 5G RAN servers and developing platform architecture for 5G. My work includes integrating networking applications on Intel SmartEdge using 5G use cases, designing features for 3D ship simulation with AWS and Azure clouds, and creating proposals for reservoir simulation, gaming, and animation. Additionally, I have developed realistic models of oil wells by modelling porosity using a modified ANFIS technique (DOI: 10.35940/ijitee.H6428.069820, 2020) and authored "Realtime Cost and Performance Improved Reservoir Simulator Service Using ANN and Cloud Containers" (DOI: 10.35940/ijitee.H6428.069820). I have also contributed to a secure routing protocol for VANETs using ECC, presented at an IEEE conference (DOI: 10.1109/ICCSEA49143.2020.9132896,2020).



**Dr. Y.V.S Sai Pragathi**, Head of department of Computer Science & Engineering, Stanley College of Engineering & Technology for Women (Autonomous), Hyderabad, [ypragathi@stanley.edu.in](mailto:ypragathi@stanley.edu.in)

