A Deep and Transfer Learning Approach for Handover Management in O-RAN

Ioannis Panitsas, Akrit Mudvari, Ali Maatouk, Leandros Tassiulas
Department of Electrical and Computer Engineering, Yale University, USA

***Abstract*—Despite advancements in the Open-Radio Access Network (O-RAN) architecture and its applications, handover management remains a use case without a comprehensive methodology for orchestrating seamless cell transitions within the O-RAN context. Existing predictive handover algorithms have not been designed or tested with scalability in mind, which is crucial for large-scale cellular networks where a Radio Intelligent Controller (RIC) can manage multiple distributed cells. In addition, their static architecture limits adaptability to dynamic conditions, making them irresponsive and impractical for real-world deployments. To address these challenges, we propose an algorithm that complies with O-RAN standards and features a flexible architecture, implemented as an xApp in the Near Real-Time RIC, to facilitate high-accuracy predictive handover decisions. We demonstrate that our algorithm outperforms state-of-the-art predictive handover algorithms, achieving over 95% classification accuracy even in large-scale networks and reducing the number of handovers compared to traditional handover algorithms. Finally, it effectively adapts to dynamic environments with a varying number of base stations, requiring at least 30% less retraining time while maintaining high classification rates.**

***Index Terms*—5G, Radio Access Network, Machine Learning**

I. Introduction

Next-Generation (NextG) cellular networks are evolving into more disaggregated, virtualized, programmable, and open systems to meet increasing traffic demands and to support the provision of new services [1]. For instance, the Open Radio Access Network (O-RAN) architecture, proposed and standardized by the O-RAN Alliance, is a new RAN architecture that embraces disaggregation and openness to reduce operational and capital costs, promotes multi-vendor interoperability, and enables the integration of Artificial Intelligence and Machine Learning (AI/ML) capabilities in the RAN for network optimization [2], [3].

The O-RAN Alliance is continually proposing and refining use cases to promote this open architecture [5]. One use case that has been identified is the *handover management*, which remains a critical area of academic focus. Handover management is not a new network related problem; it has been studied since the days of 2G networks and involves transferring an active connection from one Base Station (BS) to another to maintain seamless service continuity. Despite extensive research, handover optimization in NextG networks remains an open problem [1], [2], [5] due to new and unexplored challenges [3], such as: 1) network densification and access heterogeneity; 2) utilization of new frequency bands (e.g., mmWave and THz); 3) emergence of ultra-reliable, low-latency services with minimal interruption time; 4) dynamic, high-speed user mobility; and 5) integration of non-terrestrial and terrestrial networks —all of which further complicate handover decision-making.

In recent years, significant research efforts have focused on data-driven approaches, specifically applying ML techniques in handover optimization due to their powerful ability to accurately model the handover decision process and adapt to complex, dynamic, and temporal network conditions [4]. Predictive handover management has gained a lot of attention [6-10], due to its potential to enhance user experience by minimizing service interruptions during mobility. For instance, in [6] and [7], the authors proposed Deep Neural Network (DNN) architectures to predict radio link failures and early handovers. In [8], the authors introduced new Long Short-Term Memory (LSTM) based architectures to predict future channel quality reference signals. Similarly, in [9], a Recurrent Neural Network (RNN) architecture is presented to learn the optimal timing and destination BS for initiating handovers. Finally, in [11], the authors formulated a joint connection management and load balancing problem and solved it using Deep Reinforcement Learning (DRL) techniques.

Despite the contributions of these works, there is still progress to be made. Although the O-RAN architecture has matured since its introduction in 2018, a comprehensive methodology for handover management within the O-RAN context still remains absent. While the work in [11] addresses several issues in that direction, it lacks details on data collection, model training, and deployment within the O-RAN context. Second, while many predictive handover algorithms exist in the literature [6-10], none of them have been evaluated in scenarios where scalability is a real concern, neglecting the fact that, in production cellular networks, a RAN Intelligent Controller (RIC) can control multiple distributed cells. Finally, a significant gap remains in testing and evaluating these approaches in scenarios where dynamic events can occur, potentially impacting both the RAN topology and the algorithm architectures in real time. For instance, in NextG networks, Unmanned Aerial Vehicles (UAV) mounted BSs are expected to be deployed during short-term traffic surges, such as social events, to boost network capacity [12]. As a result, these algorithms must be inherently designed with flexible architectures capable of adapting to dynamic network environments without complete retraining, as frequent retraining with each topology change is computationally inefficient and impractical for production cellular networks.

A. Methodology and Contributions

To address the aforementioned challenges, we propose a predictive handover algorithm with an adaptable architecture consisting of three components: 1) Encoder – a flexible DNN with a dynamic input size that scales linearly with the number of active BSs connected to the RIC; 2) Stacked LSTM Component – a multi-layer LSTM model designed to extract and learn complex temporal patterns; and 3) Decoder – a configurable DNN with a dynamic output size that generates a probability distribution across cells, enabling the selection of the optimal BS for handover in the near future. We demonstrate how our architecture can predict the optimal cell based on temporal UE channel quality measurements (note that our architecture also enables the introduction of dynamic features if needed). We formulate it as a multi-class classification problem, where the objective is to minimize the cross-entropy loss between the predicted class (i.e., the predicted BS to connect to) and the actual class (ground truth BS). Finally, we show that our algorithm achieves high classification performance (more than 95% accuracy) even in large-scale networks and can be utilized in dynamic networks where the number of BSs rapidly changes with minimal retraining time while preserving high accuracy rates. Overall, the contribution of this paper can be summarized as follows:

1) *Methodology:* We present a comprehensive methodology outlining the implementation steps of a predictive handover algorithm within the O-RAN context.

2) *Prediction Algorithm:* We propose a DL-based algorithm compliant with O-RAN specifications for predictive handover management. We find that our approach achieves better prediction performance than four popular state-of-the-art handover prediction algorithms for large-scale RAN.

3) *Architecture:* We design a novel and adaptable architecture that can adjust to a dynamic number of BSs, reducing retraining time by applying transfer learning techniques to reuse learned weights in the updated model while maintaining high classification performance.

II. Background

A. Brief Introduction to O-RAN

O-RAN embraces the RAN disaggregation by splitting the monolithic BS into three functional units: the Central Unit (CU), Distributed Unit (DU), and Radio Unit (RU) [2]. The CU is responsible for functions such as radio resource control, connection management, and encryption while the DU manages real-time functions, including resource scheduling and error correction. The RU handles the transmission and reception of radio signals. The CU and DU, referred commonly as E2 nodes, are interconnected with software-based RICs via open interfaces. This enables telemetry data streaming from the E2 nodes and the implementation of closed loop control actions and policies using ML algorithms to optimize the RAN. There are two types of RICs that manage and control the RAN: the Near-Real-Time RIC, which operates on time scales between 10 milliseconds and 1 second and connects to the E2 nodes via the E2 interface, and the Non-Real-Time RIC, which operates on time scales greater than 1 second and can receive data from any functional unit of the BS through the O1 interface. ML models that are deployed in the Near-RT-RIC are called xApps, while those in the Non-RT RIC are called rApps. Each xApp can subscribe to specific RAN functions that are published from the E2 nodes to receive data that can be useful for predictions or to send control actions to the RAN. For instance, an xApp can utilize an E2 Service Model (E2SM) to receive Key Performance Metrics from the E2 nodes (E2SM-KPM) or to send a RAN Control action (E2SM-RC). With that way, a ML model can actually control the RAN and make adaptive decisions based on real-time network conditions.

B. Conventional Handover Procedure in 3GPP

According to the 3GPP technical specification [15], the handover procedure involves several key steps. Each BS first calculates a *Handover Margin (HOM)* value, which is then transmitted to all User Equipments (UEs). When a UE detects that the Reference Signal Received Power (RSRP) from a neighboring BS exceeds that of its serving BS by at least the specified *HOM* for a duration known as the *Time To Trigger (TTT)*, it generates and sends a measurement report to the serving BS. Upon receiving this report, the current serving BS determines the optimal target BS to which the UE should be handed over, thereby completing the handover process. Despite its advantages, this procedure is highly sensitive to the selected handover triggering parameters: TTT and HOM. Low values for these parameters can lead to an increase in unnecessary handovers, while higher values may result in radio link failures.

III. Model and Network Description

A. Model Design

To avoid unnecessary handovers that are highly correlated with the selection of the triggering parameters, we propose a UE-centric predictive handover algorithm that utilizes temporal RSRP measurements, to predict the future cell association of a UE. Our algorithm can be easily implemented as an xApp in the Near-RT RIC, utilizing two E2SMs: E2SM-KPM for collecting RSRP measurements and E2SM-RC to send control actions to the E2 nodes whenever handover requests need to be initiated. Building on this approach, we formalize the process as follows: given a temporal sequence of channel quality measurements (RSRP) $X_{1,N}(t-K-1),\ldots,X_{1,N}(t)$ collected over the past $K>0$ time slots from N cells, we propose a model that can predict the UE cell association vector $C_{1,N}(t+1),\ldots,C_{1,N}(t+W)$ in the subsequent time window W. The dataset of input/output features (RSRP sequences/cell associations) can be easily built by the Near-RT-RIC, which processes the time series of RSRP measurements using a sliding window approach. We emphasize that this objective represents a multi-class classification task that can be effectively addressed by our model.

In addition, we design our model with a flexible architecture to address dynamic and heterogeneous RAN environments,

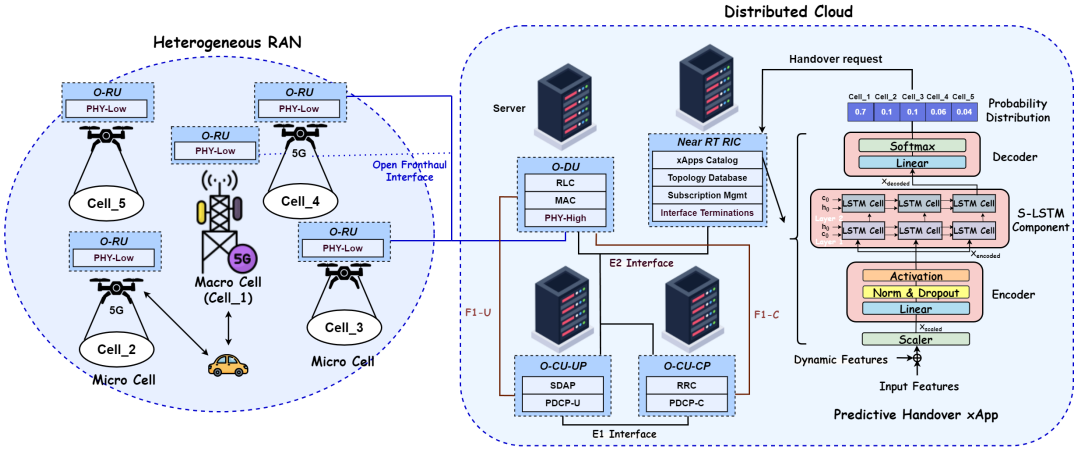


Fig. 1: Our proposed predictive handover algorithm is implemented as an xApp in the near-RT RIC.

where the number of cells vary over time. In this work, we focus on dynamic events that increase the number of BSs in the RAN resulting from the deployment of new UAV BSs within the cellular network. Evidently, a similar procedure can be followed when BSs are removed from the network. We assume that the addition of BSs can occur in random time intervals after our model has been trained. Furthermore, we assume that the placement of the new BSs is carefully managed to prevent overlap with existing cells. These dynamic events introduce variability in the input/output size of our model, which can potentially impact its performance during inference. To address this challenge, our architecture utilizes transfer learning techniques, leveraging previously learned weights to minimize retraining time while ensuring satisfactory performance.

B. Model Architecture

Based on the previous discussion, our architecture comprises three components: the Encoder, the Stacked Long Short-Term Memory (S-LSTM) layers, and the Decoder, as illustrated in the right part of Fig. 1. The Encoder is designed to dynamically adjust its input size during dynamic events and consists of three fully connected Multi-Layer Perceptrons (MLPs), each with 128 neurons, followed by a ReLU activation function. It encodes the input temporal sequence into a higher-dimensional space, enabling flexible management of variations in input size. Additionally, the Encoder produces a fixed-size output that the S-LSTM layers use as input, which is shared across all the LSTM cells. We introduce the Encoder before the S-LSTM layers to minimize the retraining time of our model. When dynamic events occur, retraining the Encoder is more time-efficient than retraining the complex stacked LSTM structure. By focusing on retraining the Encoder, we anticipate that the weights of the S-LSTM layers will stay relatively stable, thereby minimizing the overall retraining needed for these layers. Regarding the S-LSTM component, it consists of two stacked layers, each featuring a hidden size of 128 neurons, aimed at learning complex temporal patterns and extracting valuable features. The outputs from the S-LSTM layers serve as inputs for the Decoder, enabling it to produce the model's final predictions. Finally, the Decoder is a fully connected MLP with a number of neurons equal to the number of active cells in the network. It uses the last hidden state from the final S-LSTM layer, followed by a softmax layer that generates a probability distribution over all the cells in the network. The Decoder also has a dynamic structure (varying number of neurons) to adapt to changes in the number of cells during dynamic events. Leveraging this probability distribution, our model can determine future cell associations for the UE by selecting the cell with the highest probability. Note that a network operator can further refine this distribution to enhance decision-making by establishing a threshold before sending control actions to the serving BS for initiating a handover request to the predicted BS.

C. Network Model

We examine a heterogeneous RAN composed of a number of disaggregated BSs, denoted as N, which provide network access to end users. We consider two types of cells: a macro cell that operates in the n5 Frequency Division Duplexing (FDD) 5G band with a central frequency f_1 of 850 MHz, and a micro cell that operates in the n79 Time Division Duplexing (TDD) 5G band with a central frequency f_2 of 4.5 GHz. In this access network, we assume that a macro cell is deployed to provide wide coverage over an urban area, while the remaining $N-1$ cells are micro-cells that are deployed within this geographical area to enhance network capacity in regions with high user density, as illustrated in the left part of Fig. 1. In this setup, we assume that the RU of the macro cell is deployed on a tall building at a height of 25 m, characterized by an Urban Macro (UMa) Non-Line-of-Sight (NLOS) path loss model $(13.54 + 39.08 \cdot \log_{10}(d) + 20 \cdot \log_{10}(f_1))$ [14], where d is the 3D distance between the RU and a UE. The macro BS operates with a transmission power of 40 dBm and uses an omnidirectional antenna with a gain of 3 dBi. On the other site, the RU of each micro cell is deployed on a UAV at a fixed location height of 80 m, and it is characterized by an Urban Micro (UMi) LOS path loss model $(32.4 + 21 \cdot \log_{10}(d) + 20 \cdot \log_{10}(f_2))$ [14]. To simulate

more realistic path losses due to signal degradation from environmental factors, we included shadow fading with a standard deviation of 7 dB for the macro cell and 4 dB for the micro cells. Each micro BS operates with a transmission power of 10 dBm, and uses an omnidirectional antenna with a gain of 1 dBi. Finally, each UE has a RU at a height of 1.5 m and can access the network if the received power from a cell exceeds -110 dBm. Based on these network parameters, the macro cell can serve users within a maximum radius of 5 km, while each micro cell serves up to 0.5 km. Consequently, within one macro cell, up to 75 non-overlapping micro cells can be placed to create various simulated heterogeneous RAN deployments. In this work, interference between micro cells is not considered due to their non-overlapping placement and the LOS connectivity between the RUs and end users.

D. Data Collection

The next step is to generate high-quality data for training our model. We consider a scenario, commonly known in the telecom world as *drive test*, where instead of a real technical crew taking measurements of reference signals across all the geographical area, we use a simulated UE that traverses all the cells and reports channel quality metrics to fingerprint the radio environment. In our simulations, we adopt a random mobility model, with the UE traveling at a speed of 60 km/h across the entire geographic area for an extended duration (24 hours). The UE measures the received power (RSRP) for all cells, with a sampling period of 50 ms. In addition, the serving E2 node reports to the Near-RT RIC the current serving cell. In our simulations, we set the HOM to 0 dB and the TTT to 500 ms. Therefore, every 50 ms, the UE reports to the serving E2 node a feature vector $X_{1,N}(t)$ containing all the measured signals from the cells, while the serving E2 node reports a one-hot encoded vector $C_{1,N}(t)$ indicating which cell the UE is currently connected to. These measurements are aggregated to the Near-RT RIC and afterwards to our xApp via the E2 interface. With this procedure, we generate a large dataset (10 GB in size) with temporal RSRP measurements and cell associations (classes) that can be used to train our model. Note that by implementing this procedure, we can seamlessly integrate new features into our xApp as needed. Each measurement step can be parameterized with different numbers of cells, channel quality metrics, handover triggering parameters, and sampling rates, allowing us to explore various network configurations.

E. Model Training

To train our model, we split our collected dataset into sliding windows containing K temporal RSRP measurements from all deployed N cells, resulting in a feature vector size of $X_{K,N}$. This data is then flattened into a one-dimensional vector $X_{1,K\times N}$ prior to inputting it into the Encoder. To generate the label for each sliding window, we identify the most frequently associated cell for the UE during the subsequent W measurements and replicate this label across the entire window. This strategy enables our model to effectively filter unnecessary handovers, thus reducing the overall number of handovers to BSs. After constructing the sliding windows, we split the dataset into training and testing sets in a 70/30 ratio, allocating 70% of the data for training and 30% for testing. To optimize our model, we conducted hyperparameter tuning to identify the best parameters and architecture. Following this, we trained the model using the cross-entropy loss function with a batch size of 256, a learning rate of 0.0001, and employed a standard scaler for feature normalization. Finally to avoid overfitting during training, we employed batch normalization and dropout to our model. More details regarding our model training procedure can be found in Algorithm 1.

Algorithm 1 Model Training

Input: Randomized parameters θ of our model, learning rate η, batch size B, number of epochs E, number of batches N_{batches}, ground truth labels $Y_{W,N}$
Output: learned parameters θ of the model

```
for e = 1 to E do
    for b = 1 to N_batches do
        X_scaled ← Scaler(X_{K,N})
        X_encoded ← Encoder(X_scaled)
        X_S-LSTM ← S-LSTM(X_encoded)
        C_{W,N} ← Decoder(X_S-LSTM)
        L(θ) ← −(1/B) Σ_{i=1}^{B} Σ_{j=1}^{N} Y_{i,j} log(C_{i,j}(θ))
        via Back Propagation Through Time.
        Update θ ← θ − η∇_θ L.
    end for
end for
```

IV. Evaluation Results

A. Performance Benchmarking

We evaluated the performance of our model in this multi-class classification problem by comparing it with four state-of-the-art predictive handover algorithms proposed in the literature. For a fair comparison, we utilized a uniform architecture across all algorithms, consisting of three layers with 128 neurons each, followed by a ReLU activation function. The learning rate was set to 0.0001, and the batch size was 256. More specifically, we compare our model with:

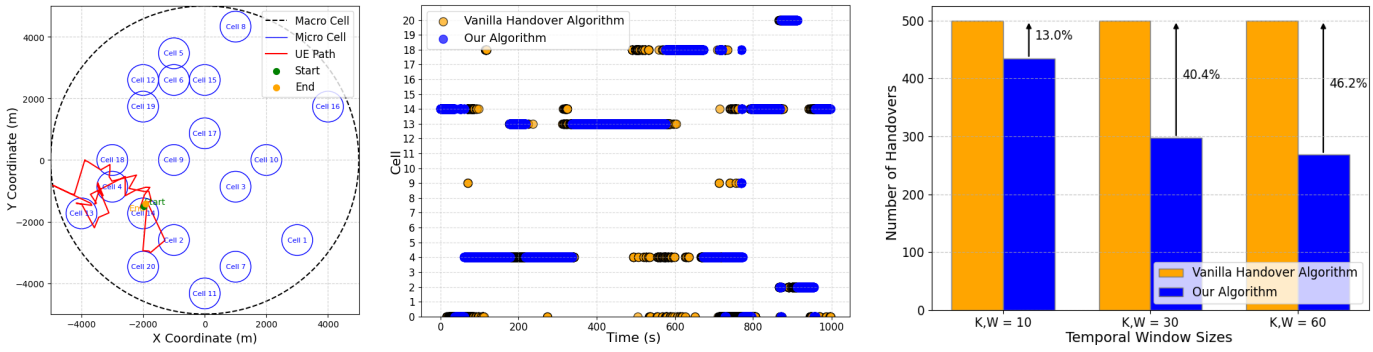
1) LSTM-based Predictive Handover Algorithm: This model utilizes stacked LSTM layers to predict cell association in the next time window [8].

2) Gated Recurrent Unit (GRU)-based Predictive Handover Algorithm: This algorithm leverages GRU as the learning component for estimating the cell association sequence [10].

3) RNN-based Predictive Handover Algorithm: Predictions are generated using a deep RNN architecture, as proposed in [9].

4) MLP-based Predictive Handover Algorithm: In these works [6], [7], a fully connected MLP is utilized for handover predictions, using features from K previous measurements.

Table I summarizes the experiments performed with different configurations, varying the number of cells N, temporal windows K, and W. For each scenario, random UE paths



(a) UE mobility scenario. (b) Cell associations over time. (c) Reduction in number of handovers.

Fig. 2: Performance evaluation showcasing (a) the UE mobility scenario, (b) the dynamic cell associations over time, and (c) the corresponding reduction in the number of handovers.

were selected for a duration of 1000 seconds, characterized by random mobility patterns, an average velocity of 60 km/h, and a sampling rate of 50 ms. Given the imbalanced nature of our dataset, we prioritize precision as a key performance metric alongside accuracy, recall, and F1 score to provide a comprehensive assessment of the predictive algorithms. As shown in Table 1, our algorithm consistently outperforms all other predictive algorithms across all metrics for every configuration tested. Notably, our model achieves higher precision, recall, and F1 scores, indicating its ability to accurately predict cell associations while minimizing false positives and negatives. The results also illustrate that our algorithm is highly scalable across different numbers of deployed cells. Even as the number of cells increases, our model maintains nearly perfect performance, showcasing its efficiency, and scalability.

N	K,W	Method	Metrics			
			Accuracy	Precision	Recall	F1 Score
21	10	**Our Model**	**0.94**	**0.93**	**0.92**	**0.93**
		LSTM	0.87	0.82	0.78	0.79
		GRU	0.85	0.83	0.77	0.78
		RNN	0.84	0.79	0.76	0.76
		MLP	0.85	0.81	0.72	0.75
	30	**Our Model**	**0.96**	**0.94**	**0.95**	**0.94**
		LSTM	0.90	0.89	0.84	0.85
		GRU	0.91	0.91	0.83	0.85
		RNN	0.88	0.83	0.78	0.79
		MLP	0.83	0.75	0.72	0.73
	60	**Our Model**	**0.98**	**0.97**	**0.97**	**0.96**
		LSTM	0.93	0.89	0.90	0.90
		GRU	0.94	0.93	0.92	0.91
		RNN	0.83	0.69	0.67	0.67
		MLP	0.81	0.74	0.68	0.69
36	10	**Our Model**	**0.94**	**0.92**	**0.93**	**0.93**
		LSTM	0.86	0.83	0.78	0.79
		GRU	0.85	0.85	0.77	0.78
		RNN	0.85	0.79	0.74	0.76
		MLP	0.83	0.72	0.71	0.70
	30	**Our Model**	**0.95**	**0.94**	**0.95**	**0.95**
		LSTM	0.90	0.89	0.83	0.85
		GRU	0.89	0.88	0.80	0.81
		RNN	0.87	0.83	0.79	0.80
		MLP	0.81	0.75	0.69	0.69
	60	**Our Model**	**0.97**	**0.96**	**0.95**	**0.94**
		LSTM	0.92	0.89	0.86	0.87
		GRU	0.91	0.91	0.86	0.88
		RNN	0.80	0.76	0.72	0.73
		MLP	0.79	0.73	0.70	0.70
51	10	**Our Model**	**0.93**	**0.92**	**0.89**	**0.90**
		LSTM	0.85	0.78	0.75	0.75
		GRU	0.84	0.78	0.73	0.75
		RNN	0.83	0.75	0.73	0.74
		MLP	0.79	0.72	0.67	0.68
	30	**Our Model**	**0.96**	**0.94**	**0.94**	**0.93**
		LSTM	0.87	0.85	0.80	0.81
		GRU	0.88	0.80	0.78	0.79
		RNN	0.85	0.78	0.75	0.76
		MLP	0.76	0.72	0.70	0.70
	60	**Our Model**	**0.97**	**0.97**	**0.96**	**0.96**
		LSTM	0.89	0.88	0.85	0.86
		GRU	0.90	0.88	0.85	0.86
		RNN	0.74	0.58	0.52	0.53
		MLP	0.72	0.58	0.52	0.54

TABLE I: Performance Evaluation.

B. Reducing Unnecessary Handovers

Leveraging the improved performance of our algorithm against the state-of-the-art methods, we assess its effectiveness in reducing unnecessary handovers, compared to the conventional 3GPP handover algorithm (referred to as the vanilla handover algorithm in Fig.2) outlined in Section II.B. We examine a scenario with 21 cells deployed in the RAN, where the deployment of cells and the UE's mobility path are shown in Fig. 2a. For this setup, the algorithm's performance is tested across different values of K and W (10, 30, and 60). As illustrated in Fig. 2b, an example with $K, W = 10$ demonstrates that our algorithm effectively reduces unnecessary handovers during the UE's mobility path. In addition, we show in Fig. 3a that the number of handovers is minimized across various temporal window sizes. Notably, our algorithm significantly decreases the number of handovers while maintaining a consistent average RSRP value for the UE throughout its mobility.

C. Learning in Dynamic Environments

Finally, we examine the retraining time gains of our model in response to dynamic events that may alter the total number of deployed cells in the network while our algorithm is deployed in the Near-RT RIC for inference. To facilitate fast retraining of our model, we utilize transfer learning by applying the previous weights to the new model, allowing for quicker adaptation to changing network conditions while preserving previously learned knowledge, as described in Algorithm 2. We demonstrate this with an example illustrated

in Fig. 3, where we initially assume that our model has been deployed for inference in the Near-RT RIC for a RAN with $N = 41$ cells. At a random moment, new $M = 4$ UAV BSs are introduced at random locations within the macro cell. As shown in Fig. 3, our adaptable architecture reduces the retraining time by approximately 50% while maintaining high accuracy (green line) compared to the alternative approach of training from scratch (red line, $N = 45$). Finally, Table II shows the retraining time improvements achieved by our architecture across various initial network configurations and dynamic additions of UAV BSs in the RAN. As it can be seen, in all cases, our adaptable architecture reduces the retraining time by at least 30%.

Algorithm 2 Fast Retraining Through Transfer Learning

Input: Pre-trained model parameters $\theta_{\text{pre-trained}}$, initial number of cells N, number of added cells M after the dynamic event, first Encoder's hidden layer h_{l1}, last S-LSTM Component's hidden state h_{lstm}.

Output: Updated parameters θ of the model.

Load the pre-trained model from the xApp catalog.
Set model parameters $\theta \leftarrow \theta_{\text{pre-trained}}$.
Deploy the model in the Near-RT RIC for inference.
Monitor the number of cells in the system.
if dynamic event occurs **then**
    Update the Encoder's input layer:
        $model.Enc1 \leftarrow Linear(N + M, h_{l1})$
    Update the Decoder's output layer:
        $model.Dec1 \leftarrow Linear(h_{\text{lstm}}, N + M)$
    Randomize weights for newly added cells.
    Transfer previous S-LSTM weights to the new model.
end if
Retrain the updated model using Algorithm 1.

N	21	21	21	31	31	31	41	41	41
M	4	8	10	4	8	10	4	8	10
Reduction in Retraining Time	52%	36%	30%	47%	43%	40%	48%	42%	40%

TABLE II: Reduction in retraining time across different initial numbers of cells N and dynamically inserted base stations M.

V. Conclusions

In this paper, we addressed the predictive handover management problem by proposing an adaptable architecture that estimates the optimal BS based on temporal UE channel quality measurements. We formulated the problem as a multi-class classification task, minimizing the error between the predicted and actual BS. Evaluation results showed that our algorithm achieved over 95% accuracy, outperforming state-of-the-art methods, and remained effective even in large, rapidly changing networks with minimal retraining time required.

References

[1] W. Saad, M. Bennis and M. Chen, "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems," in IEEE Network, vol. 34, no. 3, pp. 134-142, 2020.

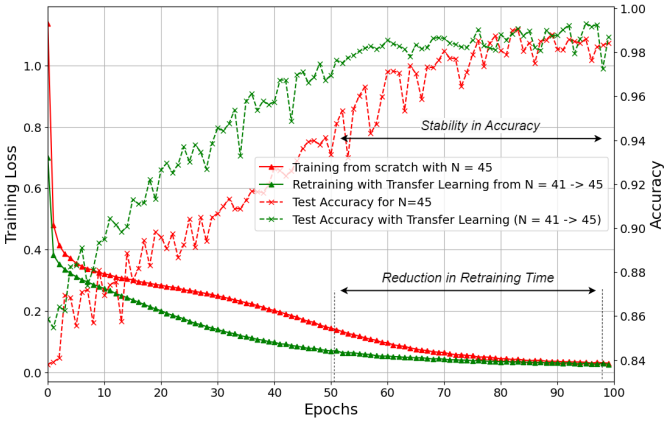


Fig. 3: Reduction in our model's retraining time.

[2] M. Polese, L. Bonati, S. D'Oro, S. Basagni and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," in IEEE Communications Surveys & Tutorials, vol. 25, no. 2, pp. 1376-1411, 2023.

[3] M. E. Morocho-Cayamcela, H. Lee and W. Lim, "Machine Learning for 5G/B5G Mobile and Wireless Communications: Potential, Limitations, and Future Directions," in IEEE Access, vol. 7, pp. 137184-137206, 2019.

[4] M. S. Mollel et al., "A Survey of Machine Learning Applications to Handover Management in 5G and Beyond," in IEEE Access, vol. 9, pp. 45770-45802, 2021.

[5] O-RAN Alliance, "O-RAN Use Cases and Deployment Scenarios: Towards Open and Smart RAN," White Paper, 2020.

[6] Z. -H. Huang, Y. -L. Hsu, P. -K. Chang and M. -J. Tsai, "Efficient Handover Algorithm in 5G Networks using Deep Learning," in IEEE Global Communications Conference, Taipei, Taiwan, pp. 1-6, 2020.

[7] C. Lee, H. Cho, S. Song and J. -M. Chung, "Prediction-Based Conditional Handover for 5G mm-Wave Networks: A Deep-Learning Approach," in IEEE Vehicular Technology Magazine, vol. 15, no. 1, pp. 54-62, 2020.

[8] N. Uniyal et al., "Intelligent Mobile Handover Prediction for Zero Downtime Edge Application Mobility," in IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, pp. 1-6, 2021.

[9] A. Masri, T. Veijalainen, H. Martikainen, S. Mwanje, J. Ali-Tolppa and M. Kajó, "Machine-Learning-Based Predictive Handover," in IFIP/IEEE International Symposium on Integrated Network Management (IM), Bordeaux, France, pp. 648-652, 2021.

[10] B. Shubyn, N. Lutsiv, O. Syrotynskyi and R. Kolodii, "Deep Learning based Adaptive Handover Optimization for Ultra-Dense 5G Mobile Networks," in IEEE International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraine, 2020, pp. 869-872

[11] O. Orhan, V. N. Swamy, T. Tetzlaff, M. Nassar, H. Nikopour and S.Talwar, "Connection Management xAPP for O-RAN RIC: A Graph Neural Network and Reinforcement Learning Approach," in IEEE International Conference on Machine Learning and Applications (ICMLA), Pasadena, CA, USA, pp. 936-941, 2021.

[12] F. Rinaldi et al., "Non-Terrestrial Networks in 5G & Beyond: A Survey," in IEEE Access, vol. 8, pp. 165178-165200, 2020

[13] S. Alraih, R. Nordin, A. Abu-Samah, I. Shayea and N. F. Abdullah, "A Survey on Handover Optimization in Beyond 5G Mobile Networks: Challenges and Solutions," in IEEE Access, vol. 11, pp. 59317-59345, 2023.

[14] ETSI, "ETSI TR 138 901 V15.0.0 (2018-07): 5G; Study on channel model for frequencies from 0.5 to 100 GHz (3GPP TR 38.901 version 15.0.0 Release 15)," Technical Report, July 2018.

[15] ETSIA, "Procedures for the 5G System," European Telecommunications Standards Institute (ETSI), Technical Specification (TS) 123 502, June 2018, version 15.2.0 Release 15