Received 12 March 2024; revised 11 May 2024; accepted 3 June 2024. Date of publication 12 June 2024; date of current version 1 July 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3413590

User Classification and Traffic Steering in O-RAN

RAWLINGS NTASSAH [1] (Graduate Student Member, IEEE), GIAN MICHELE DELL'AREA [2], AND FABRIZIO GRANELLI [1,2] (Senior Member, IEEE)

[1] DISI, University of Trento, 38123 Trento, Italy

[2] Telecom Italia, 10148 Turin, Italy

CORRESPONDING AUTHOR: F. GRANELLI (e-mail: fabrizio.granelli@unitn.it)

This work was supported by Telecom Italia (TIM) in the framework of the UniversiTIM Program.

ABSTRACT The O-RAN architectural framework enables the application of AI/ML techniques for traffic steering and load balancing. Indeed, an effective steering technique is crucial to avoiding ping-pong and radio link failure. Limited observability and network complexity make it challenging to understand individual user needs. Consequently, traffic steering methods struggle to make optimal decisions, resulting in performance degradation due to unnecessary handovers. Motivated by this, we present an xApp for the RAN intelligence controller (RIC) for user equipment (UE) steering to ensure an even load distribution among cells while maintaining an acceptable throughput level. We propose an ML-aided traffic steering technique. The proposed method comprises three phases: UE classification, downlink (DL) throughput prediction, and a traffic steering (TS) technique. A support vector machine (SVM) is used for UE classification, followed by cell throughput prediction using ensemble Long Short-Term Memory (E-LSTM). The TS algorithm uses the information from the ML models to initiate handovers (HO). The SVM model identifies UEs with low throughput, while the E-LSTM predicts cell DL throughput to provide information about potential target cells for these UEs. Experimental results demonstrate that the proposed method achieves an even load distribution of UEs in 60.25% of the cells with few handovers, while also significantly improving UE throughput.

INDEX TERMS Bagging, DL throughput, load balancing, LSTM, O-RAN, supervised learning, SVM, traffic steering, xApp.

I. INTRODUCTION

NETWORK resource demand is exponentially increasing due to the number of emerging devices and applications. Hence, mobile operators and service providers are required to meet the demands of these users and the different applications. The 5th generation (5G) is designed to meet these demands and support verticals and users with high bandwidth, low latency, and reliability. To ensure that the entire network meets the key performance indicators (KPIs), both the radio access Network (RAN) and the core network must function to meet these KPIs.

The demands on the network keep increasing, hence a requirement for a more efficient RAN deployment and management. One approach proposed is the implementation of functional splitting as discussed in [1]. The core concept of this approach is to disassociate the RAN functionalities from the hardware and run them as software. As presented by the authors in [2], cloud RAN (C-RAN) demonstrates functional splitting aimed at lowering energy consumption while increasing spectral efficiency by moving the baseband units (BBUs) to a centralized location. The authors in [3] proposed an improved version of the C-RAN. The objective of these architectural frameworks is to centralize the baseband unit (BBU) in the cloud, separating them from the Radio Frequency (RF) antenna units. Functional splitting is also demonstrated in [4] where the authors presented virtual RAN (vRAN). vRAN further provides a splitting possibility of the BBU into the distributed unit (DU) and centralized unit (CU) in a virtualized environment. The vRAN provided a more traditional approach to implementing a centralized RAN architecture. C-RAN and O-RAN [5] provide open interface deployment possibilities where the interfaces are not vendor-specific.

The advent of AI/ML has changed the narrative of network management and orchestration. Network operators and service providers are developing networks that can be effectively managed by implementing AI/ML techniques. The O-RAN Alliance has proposed a noteworthy deployment of RAN (Radio Access Network) utilizing AI/ML methods [6]. They have introduced the Open RAN (O-RAN) architecture, addressing functional splitting and the incorporation of intelligence (AI/ML) for decision-making, ensuring flexibility, scalability, and open interfaces for interoperability. The functional splitting provides the avenue for the implementation of AI/ML techniques in the architecture, as presented by the O-RAN Alliance. The RAN intelligence controller (RIC) in this architecture acts as the brain of the system, where AI/ML models can be deployed for RAN management. These models are implemented as software components (xApps) for various functionalities. The deployment, management, and challenges associated with these xApps are described by the authors in [7], [8] and [9]. One major issue in the current network is traffic management and resource allocation. The complexity of assigning resources and maintaining the key performance indicators (KPIs) rises, particularly due to the diverse network demands.

Many propositions are made in the literature to address TS problems. There are TS algorithms that are either cell-centric or UE-centric. However, O-RAN has provided an avenue where an intelligent model can assist the network in TS decisions. Unlike the conventional cell-centric approach where TS is done by the cells, a smart model provides much insight into the network for better decision-making. There are also some UE-centric methods as discussed in [10] and [11]. These approaches run ML models on mobile devices for cell selection. However, much consideration is not given to the consumption presented by these approaches. In [12], a knowledge transfer and federated learning method is discussed for UE-centric traffic steering. UEs share models which could lead to private data leakage, and local model training also adds more computation on the user device. Moreover, these methods do not give details of the load on the cells. Approaches such as the one presented in [13], [14] are load-aware and ensure even load distribution but the average throughput guaranteed by the cell is unknown.

To address these challenges and test the use of xApp in O-RAN, channel state information (CSI) is used in modeling an ML-aided xApp for traffic management. The goal is to reduce congestion without prediction, as presented by the authors in [15]. After all, limited observability makes it difficult to understand individual user needs. Instead of treating a UE at a time, we want a centralized model that scans the entire network. The model then gives us the group of UEs with low throughput or poor link quality. Hence, no extra computation is done at the UE level while support is rendered to the RAN nodes for improved network performance. The algorithm presented in this work is load-aware and triggers HOs only when the load on the target cell meets the load requirement. In our previous work [16], we discussed the K-means clustering approach to cluster UEs and then predicted the throughput using LSTM. However, when there is noisy data, the model does not generate an accurate cluster. We also observed that a single LSTM performs well but the accuracy was around 85%. To tackle these challenges, we present a two-tier machine learning (ML) model designed to classify User Equipments (UEs) and predict throughput within the traffic steering framework. The UE classification model utilizes a Support Vector Machine (SVM), while cell throughput prediction adopts an ensemble Long Short-Term Memory (E-LSTM) technique. In this context, all UEs are treated equally, and the classification process relies on the individual throughput observed by each UE. Our objective is to present an ML-aided traffic steering technique using SVM and E-LSTM techniques to guarantee optimal network performance.

The main contributions of this work are as follows:

- design an SVM classification model for user classification. The classification method puts UEs into groups of respective throughput based on a threshold, unlike most classification methods where classification is based on UE behaviors and activities.
- cell throughput prediction using E-LSTM to determine the cells with the required throughput to support the UE requirement. LSTM proves to be the preferred prediction model in the literature but E-LSTM gives a much better outcome. This prediction method does not only consider traffic patterns but also the density of UEs and the correspondent throughput.
- a steering algorithm to efficiently move UEs among cells and to ensure load balancing and congestion avoidance

Due to the high number of users expected in the network, we present a classification method for the UEs. By classifying users based on their network experience, we can prioritize those with low throughput, ensuring fairness and better resource allocation. Lack of detailed user feedback hinders efficient resource allocation. Hence, classifying users based on their reported throughput provides valuable insights into their current network performance and allows for targeted steering decisions. This enables us to understand individual user experiences and make intelligent steering decisions, effectively ensuring a better overall user experience.

Similarly, by predicting the impending changes in cell throughput, the network can make more informed decisions regarding handovers. This can help minimize the risk of handovers to congested cells, leading to smoother transitions and better user QoE (Quality of Experience). Due to traffic variations, predicting cell throughput allows the steering system to adapt dynamically, continuously improving UE experience based on real-time information and ensuring an acceptable load distribution in RAN nodes.

The structure of this paper is as follows. Section II describes the related work, while Section III presents our system model and problem formulation. We discuss the

performance and evaluation in Section IV and, in Section V, the conclusions.

II. RELATED WORK

A. USER CLASSIFICATION METHODS

User classification is crucial for mobile network management and optimization. As presented by the authors in [17], users can be classified based on Internet traffic and in-app usage. They discuss an encrypted method of mobile app usage to classify users based on their traffic consumption.

Most mobile user classifications are based on user behavior and their activities as discussed by these authors [18], [19], [20], [21]. For instance, Gabrielli et al. [22] present city user classification based on behaviors and habits. They suggest that call habits can help determine whether users are residents or visitors. These classification methods are implemented to understand the behavior patterns of UEs, their areas of interest, the kind of apps they use, and the traffic type they consume.

Unlike the methods mentioned above, which classify users based on their phone applications, message types, traffic volume (low or high), and residency status, the classification method adopted for this work considers activities occurring at different time instances and the throughput observed by different UEs. This method treats all UEs equally, intending to ensure they receive adequate throughput. This way, even if a UE's behavior changes, it will still be guaranteed an acceptable throughput.

B. THROUGHPUT PREDICTION

Bui et al. [23] surveyed contemporary network performance prediction techniques, emphasizing the significance of throughput as crucial contextual network information. These throughput prediction methods are generally categorized as active and passive. Active methods necessitate UEs to be in connected mode and transmit data packets, while passive methods generate predictions with minimal or zero network disruption [24].

Furthermore, active test methods may lead to excessive network congestion due to increased sampling in dynamic wireless environments like vehicular scenarios.

Yue et al. [25] conducted an extensive correlation analysis among Radio Signals (RSs) describing Radio Frequency (RF) parameters and labeled throughput. This study encompassed various use cases, including stationary, walking, local, and highway driving scenarios. Their findings revealed a nearly linear growth in measured throughput with increasing RSRP, RSRQ, and CQI values. The study mainly concentrated on a random forest (RF) machine learning model to predict available throughput per device. The predictive features considered included RSRP, RSRQ, CQI, and historical throughput. Notably, this proposed model is fully intrusive, as it requires the UE to be in connected mode for data collection, making replication of their model and results challenging due to insufficient description. Specifically, the study lacks details on the active tests conducted, such as the selection of labeled throughput values during data collection. Additionally, the authors did not provide insights into how features were measured concerning their corresponding labeled throughput values and their alignment in terms of time resolution.

Yao et al. [26] introduced bandwidth maps, which utilize past throughput experiences to predict current throughput at specific locations. They applied their model in adaptive video streaming and audio applications to gauge mean opinion scores. Jomrich et al. [27] took a step further by incorporating positioning data with a few RF metrics to forecast throughput for moving vehicles. Leveraging geolocation has advantages in understanding factors such as path loss, cell load, and shadowing [26]. However, we have decided not to consider geolocation as a feature in our study for several reasons. Firstly, it carries the risk of bias towards specific network settings within one location, which becomes problematic with changes in network settings, topology, and environmental conditions that could render the ML model obsolete. Secondly, there is uncertainty regarding the dimensions of the geographic area required for the ML model. Thirdly, it entails an expensive training process since data must be collected from various geographic locations over extended periods. Finally, privacy is an issue of concern when applying user data containing geolocation information. The throughput prediction method without geolocational information allows for better model generalization.

Raca et al. [28] developed a machine-learning model to forecast future throughput based on RSRQ, CQI, SINR, and historical throughput measurements. Their study explored various machine learning model types, including RF, SVM, and Neural Networks (NN), similar to the approach in [25]. Similarly, Schmid et al. [29] presented deep learning models for throughput prediction using RNN. Deep learning techniques for throughput prediction are also discussed by the authors in [30], [31]. It is worth noting that most throughput prediction methods in the literature consider mostly the traffic pattern and area map, however, we only considered the historic throughput reported by the individual UEs for the prediction.

C. TRAFFIC STEERING METHODS

Providing connectivity to the users alone is not enough to ensure the QoS is maintained for all the UEs. Resource management is essential to maintain the key performance indicators (KPI) in the O-RAN. The authors in [32] presented an RL method to obtain a near-optimal traffic steering in a heterogeneous network. The objective is to analyze the performance of an RL agent in providing load distribution in a Hetnet. Gijón et al. [33] proposed a traffic steering method for QoE optimization. They presented an optimization method to handle inter-frequency handovers (IFHOs) to guarantee users an acceptable QoE.

Due to the high number of network devices and applications, most systems are designed to be energy efficient to reduce energy consumption. Math et al. [34] proposed

lightweight supervised ML techniques at the MEC to assist radio resource heads (RRHs) in traffic distribution. One of the objectives of these authors is to design an energy-efficient traffic distribution system using ML. Similarly, the authors in [35] discussed energy-efficient traffic steering. They presented a steering technique that takes into consideration the power source.

Kavehmadavani et al. [36] proposed a traffic steering technique in the Near-real time RIC to manage URLLC and eMBB traffic and to analyze their co-existence. An extension of this work is discussed in [37] where LSTM is used in traffic prediction to enhance traffic steering in the RIC.

Network management and operations are becoming sophisticated due to the diverse network applications and devices emerging recently. An increase in network devices suggests the possibility of network congestion at certain times in the network. Hence, HOs will be triggered to ensure that UEs closer to a neighboring cell (target cell) are receiving sufficient signal strength or based on any of the principles described in [38] and [39]. Although HO can guarantee UE an acceptable signal strength, signal strength alone does not indicate that a UE is receiving adequate throughput in a congested cell since interference affects throughput [40]. One way to ease HO decisions in the network is by having a prior understanding of the UEs experiencing low throughput. The number of UEs in the cells and the average predicted throughput of the cells will also facilitate easy decision-making during HOs.

To that effect, we propose a classification method for UEs in the cells. This method provides the network with a synopsis of users and the throughput they are receiving. The combination of both classification and throughput prediction approaches is not discussed in the literature. Our preliminary analysis includes an xApp deployment and an algorithm for traffic steering and load balancing in the O-RAN architecture. We execute user equipment (UE) classification using SVM to distinguish UEs with lower throughput and signal strength from the ones with higher signal strength and throughput. We then present a cell throughput prediction and handover execution to guarantee an even distribution of users in the network.

III. SYSTEM MODEL AND PROBLEM FORMULATION

In this framework, we have the O-RU, O-DU, and O-CU, which receive UE radio signals and process them. The UE metrics are then sent through the E2 interface to the Non-RT RIC for the ML model design and implementation.

A. NETWORK MODEL

According to the O-RAN Alliance specification for RAN architecture, the RAN components could be separated or bundled. The bundling of these entities determines the interfaces that will be exposed between the Near RT-RIC and the O-RAN components. In this experimental analysis, the bundling of the O-CU-CP, O-CU-UP, O-DU, and O-RU is considered, depicting the testing deployment as shown in Fig. 1.

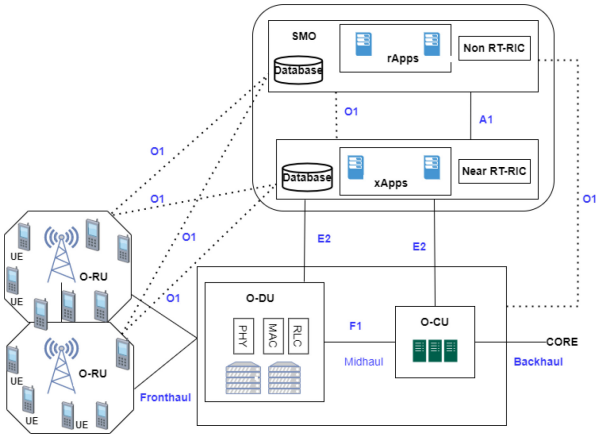

FIGURE 1. O-RAN deployment approach where UE information from the E2 nodes are constantly sent to the database for ML application.

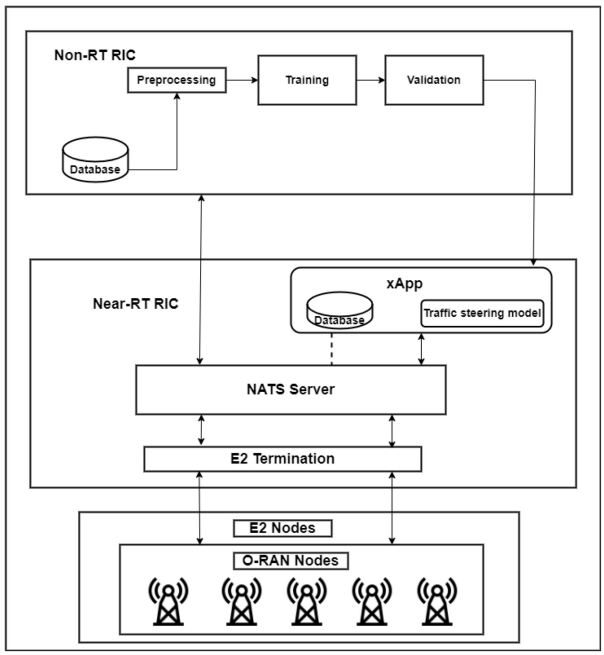

FIGURE 2. System model design and workflow.

B. WORKFLOW AND DATA GENERATION

Fig. 2 illustrates the workflow and data generation process. The E2 nodes generate network traffic containing the performance metrics of UEs, comprising both UE and cell information. This information is transmitted from the E2 nodes through the E2 termination using a UDP socket via the E2 interface. The E2 termination node is also connected to a shared data layer (SDL) Neural Autonomic Transport System (NATS). NATS is a publish/subscribe platform for information exchange between services. It provides a secure layer for data exchange between the xApp and the E2 termination. This allows xApps to receive and process data they are subscribed to, whereas E2 termination receives only the subscribed KPM sent by the xApp.

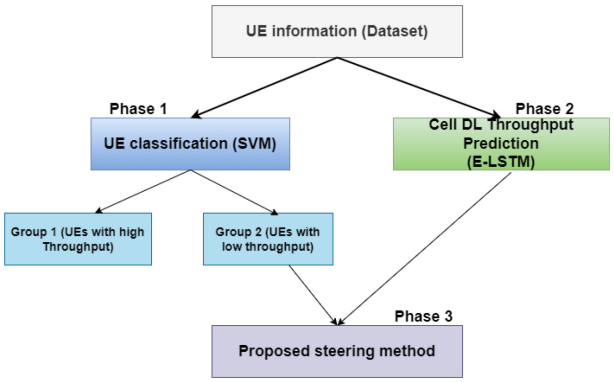

FIGURE 3. This image illustrates the method adopted for our steering technique.

The received data undergoes processing and is utilized for model training. The training and validation phases of the model occur within the Service Management and Orchestration (SMO) environment before the model is deployed in the Near-Real-Time Radio Intelligent Controller (Near-RT RIC) as an xApp. The xApp receives data for both UE classification and throughput prediction and subsequently produces outcomes, such as the generation of the "HANDOVER REQUEST" message by the TS algorithm. This message is then transmitted to the E2 nodes via NATS and E2 termination. Therefore, this work is presented in three parts:

- Triggering UE classification based on DL throughput received.
- Predicting the average throughput of all cells
- Initiating handover for UEs that require new cells.

IV. PROPOSED MACHINE LEARNING MODELS AND TRAFFIC STEERING ALGORITHM

E2 nodes generate cell and UE Information. The information provides UE and cell association. Table I illustrates UE/cell parameters and information that the xApp is subscribed to receive from the E2 nodes. The information contains physical resource block (PRB), DL throughput, channel quality indicator (CQI), DL modulation and coding scheme (MCS), the reference signal received power (RSRP), and other parameters.

The methodology adopted in this work comprises three phases. In Phase 1, we conduct UE classification based on DL throughput. Phase 2 involves cell DL throughput prediction, and the final phase, phase 3, introduces the proposed traffic steering method.

As shown in Fig. 3, the classification method we applied is an SVM to put UEs in groups based on throughput received. The E-LSTM (Bagging) method predicts cell throughput while we use the steering method to redistribute UEs.

A. SUPPORT VECTOR MACHINE (SVM)

SVM [41] is supervised machine learning for data classification using a hyperplane. SVM is an algorithm used to maximize functions in association with data. SVM is

TABLE 1. UE and cell information parameters.

E2 Termination Information	
UE information	• TTI • DL Throughput • DL MCS • Number of neighboring cells • CQI • Neighbor Cells • PRB • RSRP • RNTI • TMSI • RLC Buffer DL • RLC Buffer UL • RI
Cell information	• TTI • Number of UEs • Max PRB • DL Throughput • UE list • PCI • MMEC

* Transmission Time Interval (TTI), Downlink (DL), DL Modulation and Coding Scheme (MCS), Channel quality indicator (CQI), Physical Resource Block (PRB), Reference Signal Received Power (RSRP), Radio Network Temporary Identifier (RNTI), Temporal Mobile Subscriber Identifier (TMSI), Physical Cell ID (PCI), Rank Indicator (RI), Radio Link Controller (RLC), Mobility Management Entity Code (MMEC)

derived from [42], the concept of support vectors, denote points closer to the decision boundary. These data points are difficult to classify due to their position and closeness to the boundary (hyperplane). SVM aims to determine the hyperplane that maximizes the margin between the classes in an N-dimensional space [42], [43].

Hyperplanes depend on the number of features that exist in the dataset. The number of features determines the number of hyperplanes that can be generated. To find the hyperplane, SVM utilizes support vector classifiers (SVCs) to help determine the hyperplane. When data is difficult to separate, SVM transforms the data into higher dimensional data to determine the SVC that separates the data. SVC is determined by employing kernel functions. Kernel functions convert low-dimensional data to high-dimensional data. The versatility of the SVM is one key reason for its application in this work since you have different kernel functions to capture complex relationships in the data.

Due to the robustness provided by SVM in noisy data, we are confident of the results obtained by the classification since it is less sensitive to outliers. The reason for the choice

of this ML model is further explained in Section V-I. We perform a binary classification on our dataset X to obtain Γ and Ω. UEs in Γ are denoted and referred to as satisfied UEs since they have adequate throughput, and those in the Ω class are the unsatisfied UEs.

B. DL CELL THROUGHPUT PREDICTION

Information about the throughput UEs are receiving from cells is important. It provides a general overview of the performance of cells, as throughput could be altered by interference, cell congestion, and a low RSRP.

Therefore, predicting the DL throughput will aid in cell selection and association. E-LSTM (Bagging) is discussed in this work for DL throughput prediction. LSTM [44] is a Recurrent Neural Network (RNN) technique. RNNs are ML models used in handling sequential data where the future depends on the past. The problem with RNN is the vanishing or exploding of gradients over long dependencies [45], [46]. Vital information is lost when gradients explode or vanish, which makes RNN models perform poorly. LSTM is proposed to address the long dependency issues of losing important information over time in the RNN. LSTM applies the constant error carousel (CEC) to save errors. The saved errors enable LSTM to provide memory blocks for long periods with input and output gates [47]. These gates decide the type of information to save and the ones to forget, which allows the model to save important information.

LSTM is used in the literature for throughput prediction [16], [29], [48], [49]. The choice of LSTM is mainly based on the time series data used for this application. Based on the experimental analyses done by these authors, single model LSTM outperforms most traditional approaches and demonstrated that LSTM is robust to fluctuations mostly in time series data, hence the motivation to propose the ensemble method for performance improvement. A single model under consideration has input x_t that comprises historical throughput with features like DL MCS and PCI, RSRP, CQI, number of UEs, and the throughput. Conversely, the output y_t is the combination of the hidden and the cell state as described in the architectural framework of LSTM in [50], containing only the anticipated future throughput.

Bagging is an ensemble method of model definition where models are combined to improve prediction accuracy. In general, ensemble learning [51] methods are ML techniques where a meta-learner is used to improve predictive performance by combining other models. The principle behind ensemble is to take decisions made by the different or same models and either average or choose the maximum number predicted (maximum voting) [52]. Bagging [53] performs ensemble learning using bootstrap aggregating. This is achieved by taking bootstrap samples of the data repeatedly. The data is randomly sampled for all the predictors. The average is computed for the resulting outcome from all the predictors.

We take our dataset X to train our baseline LSTM models in our experimental scenario. A bootstrap sample of our data is taken to train and validate our base models. We choose how many base models we require for the E-LSTM technique. This gives us a list of base models needed for our bagging technique. The total number of base models *base_models* in this scenario is 10. 10 models provide the highest prediction accuracy but are not O-RAN compliant, as discussed in Section V-C. These base models are saved in the *all_models* for the ensemble method. The outputs predicted $pred_i$ (where $pred_i \in Pr$, the total predicted) from the base models are used for the E-LSTM model as illustrated in Algorithm 1. Each model does a multi-step prediction, passing these predictions through the meta_learner. The output of the meta_learner(E-LSTM model) is also a multi-step prediction hence the need for the aggregation for the final output $y_{bagging-output}$

Where; y_{e-lstm} is the output of our ensemble learning predictions, P_r is the total predictions, and $y_{bagging-output}$ is the final output.

Fig. 4 shows the bagging method described above. The original data is bootstrapped into random samples where different learners (ML prediction algorithms) are applied to the different sample data. Parallel prediction is done for all predictors, and the average is computed as the

Algorithm 1 E-LSTM Model Using Ensemble Bagging Technique

procedure E-LSTM($X_{\text{train}}, y_{\text{train}}$, base_models)
    Initialization to store LSTM base models:
$All\_models = []$
$base\_model = 10$
    for $i = 1$ to base_models **do**
        Bootstrap sample of our data: $X_{\text{bootstrap}}, y_{\text{bootstrap}} \leftarrow \text{resample}(X_{\text{train}}, y_{\text{train}})$
        Build an LSTM base model
        Train the LSTM model on the bootstrap sample: $X_{\text{bootstrap}}, y_{\text{bootstrap}}$
        Add the trained LSTM model to *All_models*
    end for
    Create an ensemble LSTM model
    Create a list to store base model outputs:
$E\_outputs = []$
    for base_model in list of *All_models* **do**
        Generate predictions on the test set using the base model
        Add predicted outputs of base models ($pred_i$) to *E_outputs*
    end for
$E - LSTM\_model = meta\_learner(E\_outputs)$
    return E-LSTM_model

    During deployment, we aggregate the output from the E-LSTM model
$y_{e-lstm} = E - LSTM\_model(E\_outputs)$
$y_{bagging-output} = (1/P_r) * \sum y_{e-lstm}$
end procedure

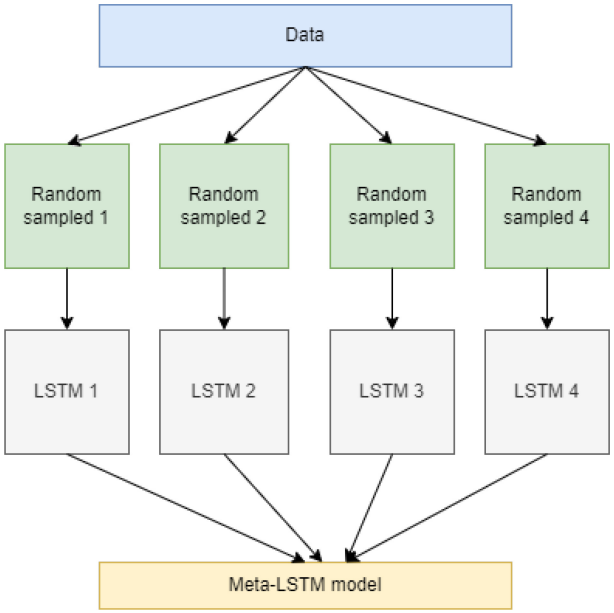

FIGURE 4. Ensemble bagging demonstrating the prediction method.

final prediction. The final prediction from the bagging model presents the DL throughput for τ (same as the total predictions steps P_r) timestamps.

Let's take k a cell amongst the list of cells (K) and its predicted throughput ($thrp_k$). The throughput predicted ($thrp_k$) is determined as illustrated in eq. (2).

$$thrp_k = \left[thrp(1), thrp(2), \ldots, thrp(\tau)\right], \quad k \in K \tag{1}$$

where $thrp(j)$ is the throughput predicted for time instance j, where j is from 1 to τ. The predicted output $thrp_k$ represents the throughput predicted by the meta model (E-LSTM model) We then compute the average of this cell's throughput $thrp_{av,k}$ ($y_{bagging\_output}$) as shown in eq. (2).

$$thrp_{av,k} = 1/\tau \sum_{j=1}^{\tau} thrp(j) \tag{2}$$

C. STEERING ALGORITHM FOR LOAD BALANCING

As mentioned earlier, this work is presented in three phases. After the classification and the throughput prediction, we present a steering algorithm. Traffic steering in the wireless network involves UE HO among cells. To trigger HO some underlying conditions are considered. These are the RSRP, RSRP threshold, and hysteresis margin during mobility [38]. Since a higher RSRP value does not necessarily translate to a higher throughput attained by a UE, it is essential to consider the possible throughput attainable to avoid ping-pong, which could lead to link failures. Hence, having an acceptable UE distribution will positively impact the throughput received by UEs since there will be low interference and fewer HOs triggered.

The objective is to initiate Handover (HO) requests for UEs with lower throughput, thereby ensuring a balanced network. During UE steering, a crucial factor is comparing the serving cell's average throughput ($thrp_{av,k_{sc}}$) with the potential target cell's average throughput. However, it is essential to acknowledge the impact of the RSRP value on throughput [54].

Let us consider a UE $x \in \Omega$, where Ω denotes all UEs with low throughput identified by the classification method. The first set of conditions is to check the RSRP of the serving cell ($rsrp_{sc}$), if it is below any of the target cell's RSRP ($rsrp_{tc}$), then the next stage is to check if the average throughput predicted for the target cell $thrp_{av,k_{tc}}$ is greater than the throughput threshold $Avthreshold$. The final condition to check is the total number of UEs in the target cell. If the number of UEs in the target cell $UE_{k_{tc}}$ is less than the threshold value μ for UEs, then the target cell (k_{tc}) is considered a potential candidate for HO.

In a system designed for its flexibility, if the initial set of conditions is not satisfied, We analyze the second set of conditions. The disparity in RSRP values between the target and serving cells is checked, ensuring it falls within the designated range δ. Furthermore, the predicted throughput of the target cell must exceed the threshold $Avthreshold$, while simultaneously accommodating UE values lower than μ as in the first set of conditions. The target cell is considered qualified to receive the UE if these conditions are satisfied. If the first two conditions are not satisfied, the last set of conditions are checked. The RSRP values and the average predicted throughput values for both the serving and target cells are compared. The difference in RSRP between the target and the serving cell must not exceed δ as in the second set of conditions. The difference between $Avthreshold$ and $thrp_{av,k_{tc}}$ is computed and the value must fall within the range ρ. However, flexibility is not granted to the total number of UEs in the target cell. A list of UEs is then sent to the E2 nodes as a list of HO request messages [msg, msg,....]. A detailed presentation of the various conditions and definitions employed in this study is provided in Algorithm 2.

V. PERFORMANCE AND EVALUATION

An O-RAN system simulator is employed to evaluate our xApp deployment. This simulator is a modified version to the one used in our previous work [16] with respect to the amount of traffic generated. The system simulator's architecture is detailed in Fig. 5. As shown in the figure, the simulator highlights separate objects for each layer (MAC/RLC/IP/Transport) specific to both gNBs and UEs. A unique identifier called "MacNodeId" is assigned to each object, allowing us to distinguish between different elements in the network (UEs, gNBs, or relays if included in the simulation). Three separate objects, PhyLayer, Channel, and ItuChanneHet3D, are used consistently across all simulations. These objects work together to define the scenario and handle packet transmission: PhyLayer simulates the physical layer behavior, Channel represents the communication channel, and ItuChanneHet3D provides a 3D model for signal propagation

Algorithm 2 Handover Request Message

INPUT: $cells(K), thrp_{av}, \Omega$
OUTPUT: List of HO request message $[msg, msg, \ldots]$
$rsrp_{tc} \leftarrow [rsrp_1, rsrp_2, \ldots, rsrp_9]$
$\delta \leftarrow [0 : 10]$
$\rho \leftarrow [0 : 10000]$
$Avthreshold = average(\Gamma)$

for all $x \in \Omega$ **do**
  if $rsrp_{tc} > rsrp_{sc}$ **then**
    if $thrp_{av,k_{tc}} > Avthreshold$ **then**
      if $UE_{k_{tc}} < \mu$ **then**
        $msg \leftarrow \{x, k_{tc}\}$
      end if
    end if
  else if $||rsrp_{tc} - rsrp_{sc}|| \in \delta$ **then**
    if $thrp_{av,k_{tc}} \geq Avthreshold$ **then**
      if $UE_{k_{tc}} < \mu$ **then**
        $msg \leftarrow \{x, k_{tc}\}$
      end if
    end if
  else if $||rsrp_{tc} - rsrp_{sc}|| \in \delta$ **then**
    if $||thrp_{av,k_{tc}} - Avthreshold||$ in ρ **then**
      if $UE_{k_{tc}} < \mu$ **then**
        $msg \leftarrow \{x, k_{tc}\}$
      end if
    end if
  else
  end if
end for

- *PhyLayer:* the role of the Physical Layer is to transport all the scheduling information from the Mac Layer to the Channel.
- *Channel:* Utilizing scheduling data that includes transmitter identities and allocated radio resources enables the computation of Effective SINR for all active radio links. Effective SINR, a condensed metric indicating radio link quality, is derived following the methods outlined in [55]. This information proves valuable for measuring CQI/RI/PMI and aiding the system simulator in determining packet reception accuracy.
- *ItuChannelHet3D:* This serves as a data structure to represent the 3D propagation channel model proposed in [56] for New Radio (NR) technology. It encapsulates all the relevant information about the scenario, including signal propagation characteristics for downlink (DL) transmission from gNBs to UEs and uplink (UL) transmission from UEs to gNBs. Key information includes fading processes and geometric details like path loss, angles of arrival (AoA) and departure (AoD), number of antennas involved, and polarization. ItuChanneHet3D captures this fading and geometric information using two indices: msIndex for UEs and bsIndex for gNBs. There's a one-to-one correspondence between msIndex and a UE's MacNodeId, and between bsIndex and a gNB's MacNodeId.

The generation of data traffic is based on [57, Ch. A.2.1.3.1], both full buffer traffic and CBR (Constant Bit Rate) traffic types are supported in the system simulator. Full buffer traffic is generated either at the MAC (Medium Access Control) layer of the gNB for DL simulations or at the MAC layer of the UE for UL simulations. In full buffer mode, a single large MAC packet is created within the MAC buffer. Because there's only one large packet, performance metrics are gathered at the MAC layer of the UE for DL simulations and at the MAC layer of the gNB for UL simulations.

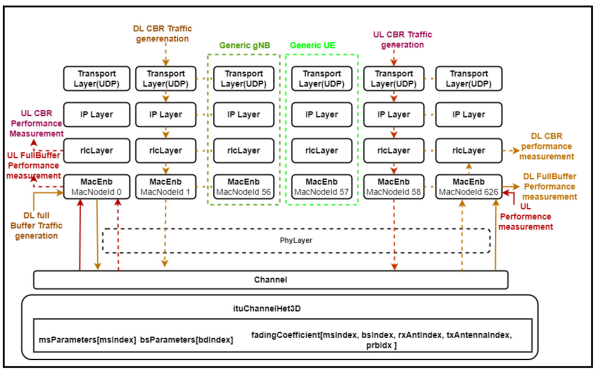

FIGURE 5. Simulator Architecture.

In contrast, non-full buffer traffic involves each UE or BS generating multiple packets during the simulation. The Transport Layer, IP Layer, and RLC (Radio link control) Layer handle the header overhead for these generated packets. Additionally, the RLC layer manages packet segmentation based on the Radio Resource allocation provided by the scheduler component of the MAC Layer.

This simulator operates by the guidelines for the Urban Macro (UMa) scenario defined in 3GPP TR 38.901. The considered release provides the specifications required to generate cell and user information. The simulator adopts the UE-RAN deployment method represented in Fig. 2. The E2 nodes generate information about the UEs and the cells (PCIs). The UEs are positioned randomly in the different PCIs (cells). 570 UEs are dropped in 57 cells with equal resource allocation and without prioritization. Each cell operates on a 10MHz channel with 50 PRBs.

The information is generated for both PCIs and UEs from the E2 nodes and received by the E2 termination. The received information is converted to a JSON file by the E2 termination and sent to the shared data layer (SDL), the NATS server. The E2 termination is subscribed to machine learning information (ml-info) and publishes UE information (ue-info) to the NATS. The xApp, on the other hand, is subscribed to ue-info and publishes ml-info to the NATS server. These components are executed in different docker containers. This setup is used to generate data for both training and validation. For clarity and simplicity, we divide Section V into two subsections: training, validation, and inference.

A. TRAINING AND VALIDATION OF MODELS

As we discussed initially, our proposed method has three phases. For clarity, we will be sure to present the results accordingly. The data received from the E2 interface is saved

TABLE 2. The table below illustrates the minimum, maximum, average, and the values used in the classification.

	RSRP (r)[dBm]	**DL Throughput** (m) [kbs]	**DL MCS** (t)	**CQI** (g)
Maximum value	-19.40	75752	27	15
Minimum value	-73.90	0	0	0
Average	-52.38	20921	19.45	12.21
Value used	-76.00	15858	20	9
Class 0	r <-76.00	m <15858	t <20	g <9
Class 1	r ⩾ -76.00	m ⩾ 15858	t ⩾ 20	g ⩾ 9

in CSV format. The simulator was run for about 48 hours and we collected about 2GB of data.

1) UE CLASSIFICATION

Redundant data such as uplink throughput and uplink MCS are excluded since they add no value in this case. An SVM classification model is defined using DL Throughput, MCS, CQI, and RSRP. During the training of the model, we first labeled the data by giving the UEs two defined classes. The classes are defined using Table 2. Each UE is considered a full buffer hence, receives a large amount of traffic. Let's consider the relationship between the mentioned parameters and the throughput. The signal strength (RSRP) determines the signal-to-interference plus noise ratio (SINR). SINR dictates the CQI experienced by the UEs and the type of modulation to adopt. MCS decides the modulation scheme to apply based on the SINR, a high SINR value could trigger 64QAM and lower SINR a QSPK. The higher the MCS, the more bits are transmitted and the higher the throughput [58], [59]. Hence, the minimum and the maximum values generated are registered and carefully selected for the classification. From Table 2, UE $x \in X$ with RSRP (r), DL Throughput (m), DL MCS (t), and CQI (g) values greater or equal to *value used*, then the UE is in *Class 1* and vice versa *Class 0*.

The data is then processed by removing redundancies and outliers. The data is then normalized using the min-max scaler. After the data preprocessing, the data is split into training and validation data of 80% and 20% respectively. We execute the classification of our data using an SVM with a linear kernel. We employed different SVM kernels for this classification, and the best for this data is the linear kernel. Some features considered in the classification are DL Throughput, DL MCS, CQI, and RSRP. Throughout the training and evaluation phases, UEs are categorized into two classes: Class 1 (same as Γ) denotes UEs with sufficient throughput, while Class 0 (Ω) signifies those with lower throughput. The ensuing classification results for the model training are depicted below. The objective of this classification method is to identify potential UEs for HO. Random forest (RF) and K-Nearest Neighbor (KNN) classification techniques are also trained on our dataset. From the observation, the RF model has high precision in classifying the Class 1 UEs, while SVM is slightly lower and KNN has the lowest. In contrast, SVM has the highest precision on low throughput UEs making it the choice for the classification method as shown in Table 3. Based on the above considerations, SVM is used for our classification model. However, RF could also be a good fit.

TABLE 3. Classification metrics of ML algorithms.

ML method	Precision		Recall	F1 Score
RF	Class 0	0.96	0.92	0.90
	Class 1	0.99	0.98	0.98
KNN	Class 0	0.78	0.67	0.76
	Class 1	0.95	0.97	0.96
SVM	Class 0	0.98	0.93	0.92
	Class 1	0.96	0.99	0.98

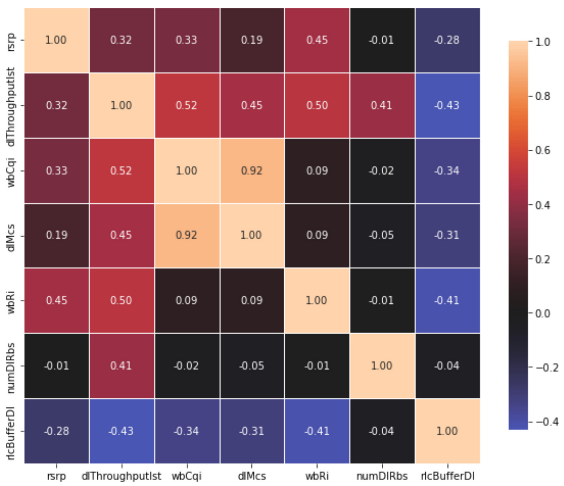

FIGURE 6. Throughput received by the two classes of UEs.

2) DL THROUGHPUT PREDICTION AND TRAFFIC STEERING

The data received from the E2 nodes are used for the throughput prediction to decide potential target cells k_{tc} for UEs. The cell information is analyzed based on the throughput each UE attached is receiving. Due to the feature engineering applied during the classification method, a correlation matrix is generated to determine the relationship between the features of our data. Fig. 6 shows the correlation matrix and the relationship between the various parameters of UE information. To enhance model accuracy during training, redundant features and parameters are eliminated. From the correlation matrix, features such as the DL MCS and CQI are highly correlated and are determining factors in the throughput received by the UE. Due to the collinearity between these two features, we dropped the CQI for the training.

TABLE 4. Simulation parameters.

Parameters	Values
Physical cell ID (PCI)	57
Number of UEs	570
LSTM layers	3
LSTM	[200, 100, 1]
Loss function	RMSE
Optimizer	Adam
Number of epochs	1000
Batch size	32
learning rate	0.01

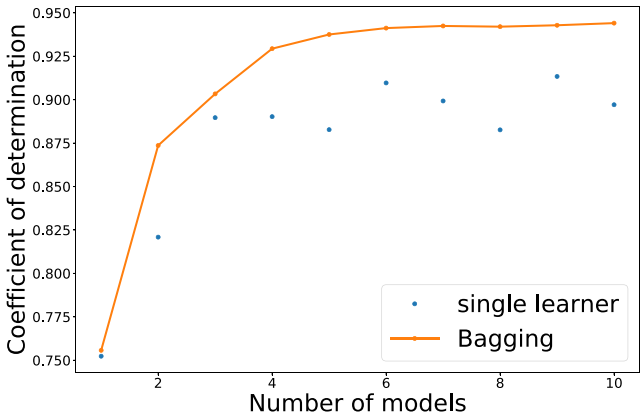

FIGURE 7. Coefficient of determination.

To design the bagging technique for our model, we establish a three-layer LSTM model. The first layer of the model comprises 200 units, 100 units for the second layer, and an output layer. The loss function applied is the root mean squared error (RMSE), and the optimizer for the individual models is adaptive moment estimation (Adam). Table 4 summarizes all the parameters used in this work.

During training, the data is split into training and validation. 80% of the data for training and 20% for validation. Data preprocessing is done just as in the classification method presented above. The data dimension is reduced with principal component analysis (PCA). These trained models are used as the based models to perform the ensemble. In this experiment, ten base models are deployed for the bagging process. We conducted a comparative analysis between the bagging method and a single model to evaluate performance. Fig. 7 illustrates the prediction coefficients between the single member and the ensemble method. The prediction coefficient of the bagging technique remains consistently high compared to that of the single models. The R^2 score demonstrates that the bagging technique outperforms the single model.

B. INFERENCE MODE AND TESTING IN THE RIC

1) UE CLASSIFICATION IN DEPLOYMENT

The setup presented earlier for data collection (Section III-B) is the same environment used for the inference test.

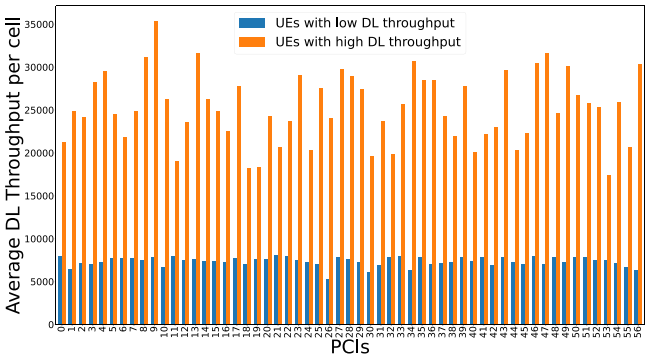

FIGURE 8. Group of UEs receiving acceptable throughput (Satisfied) and a group of UEs receiving lower throughput (Unsatisfied).

The trained SVM model and that of bagging (E-LSTM) are deployed in a microservice. The microservice (docker container) is deployed in the Near-RT RIC where data is exchanged between the docker container and the SDL. The received UE and cell information through the SDL are used for the UE classification and prediction of DL throughput. As indicated in the introductory part of this section, the subscription information for our xApp is CSI (ue-info/cell-info). This information is received through the SDL (NATS). The UE information received is used for the classification, based on the levels of throughput they receive from the cells, as depicted in Fig. 8. UE throughput can be affected by various factors such as interference, low signal strength, or the distance between the serving cell and the UE. The classification output distinguishes groups of UEs with high and low data rates.

2) THROUGHPUT ANALYSIS

In this part, we present the performance of our proposed method to existing literature. We combine the SVM method, the E-LSTM method, and the traffic steering algorithm. The objective is to analyze the performance of our proposed method, where we combine three different techniques.

In the first step, we analyze the performance of KNN, RF, LSTM, and E-LSTM using the proposed method. In the same experimental scenario, we deployed these three models separately to predict the DL throughput for the cells in our network. Fig. 9 shows the CDF distribution of the four models deployed to predict the throughput of cells. In the first quartile, RF, E-LSTM, and the single LSTM models tend to be slightly similar in the amount of throughput the UEs are observing. The single LSTM model demonstrated some improvement to the KNN and RF models. The E-LSTM proved to predict the throughput efficiently, which accounts for a better overall CDF of the throughput than the rest of the other models.

One point worth noting is that our experimental scenario is stationary, with no UE mobility considered in this scenario. The throughput received by users, therefore, does not vary that much. Hence, if the throughput of a particular UE is low, there will not be changes until the network condition

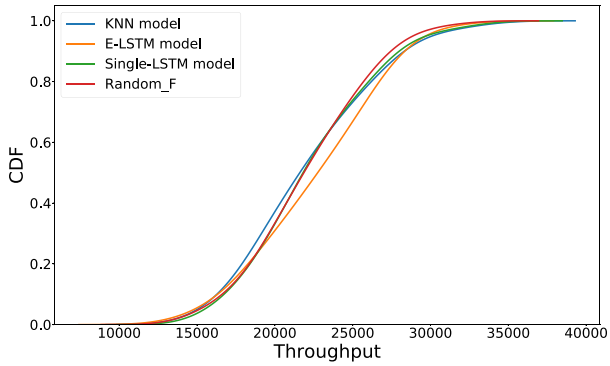

FIGURE 9. CDF of Throughput received by the UEs by the different ML models.

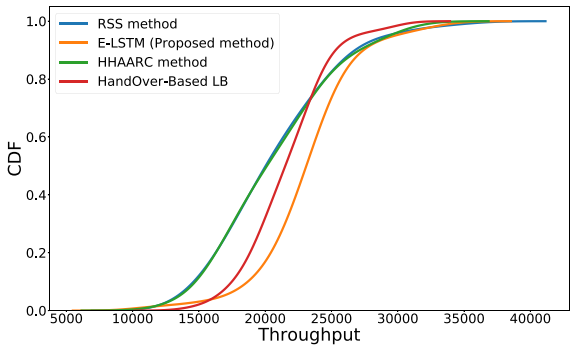

FIGURE 10. CDF of Throughput received by the UEs.

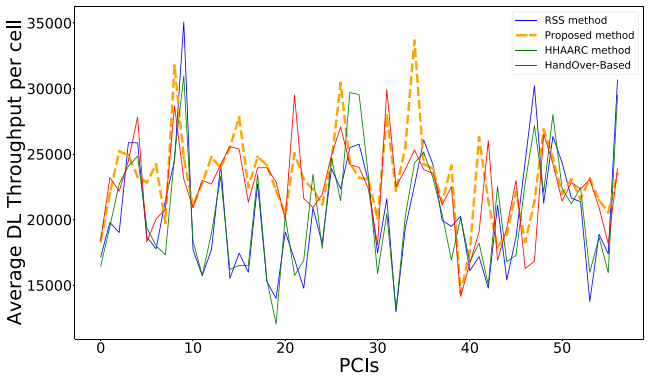

FIGURE 11. This figure illustrates the average throughput generated per cell, which is attainable by UEs. UEs in the cells can experience a minimum throughput around the average generated by the cell.

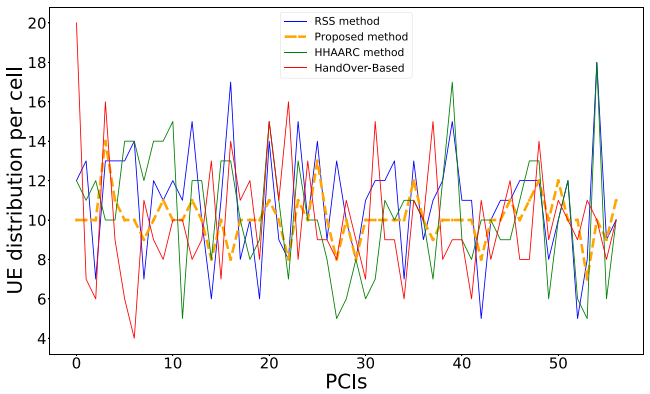

FIGURE 12. User distribution after handovers triggered in the cells for the different methods.

changes, either by reducing the load on the cell or by providing the UE with better channel conditions such as interference reduction.

This work combines three methods and is compared with the referenced signal strength (RSS) method described in [60] and [61], a Hard Handover Algorithm with Average RSRP Constraint (HHAARC) discussed in [62] and Handover-Based load balancing method (HB) [13]. Fig. 10 illustrates the CDF of the throughput in our network scenario after applying the different methods. The proposed method shows a significant improvement in the throughput the UEs are experiencing. The RSS and the HHAARC methods showed similar performance metrics compared with the Handover-Based method. The Handover-Based method showed a much improved performance but our method outperforms all the other three.

To assess the effectiveness of our model, we conducted a comparative analysis of the average throughput distribution among UEs across various cells. Fig. 11 illustrates the average throughput generated by each cell using different methods. Our proposed technique exhibited notably higher average throughput per cell than the Handover-Based method. Additionally, the Handover-Based method displayed a significant average throughput compared to RSS and HHAARC. Fig. 11 depicts the throughput improvement for the individual cells.

3) UE DISTRIBUTION ANALYSIS

UE distribution is done randomly in assigning UEs to cells. Due to the random distribution of UEs, some cells have a higher number of UEs than others. An evaluation of the distribution of UEs in the individual cells is illustrated in Fig. 12. The network has 570 UEs and 57 cells, and the objective is to achieve an equitable distribution of these UEs among the 57 cells. Our method achieved the most balanced distribution, with the load evenly distributed across the cells, as evidenced in Fig. 13. Conversely, other methods exhibit an uneven UE distribution. For example, the Handover-Based, HAARC, and RSS methods have cells accommodating approximately 20, 18, and 18 UEs, respectively.

The proportion of cells with an average of 10 UEs rose from 14.03% (simulator) to 60.25% (our method). Similarly, the HAARC method also experienced a modest increase compared to the original distribution, as depicted in Fig. 13. Our proposed method ensures a well-balanced distribution of UEs across the network, resulting in consistently high throughput levels. This is one of the objectives of our proposed method: to enhance UE distribution across cells while simultaneously maximizing throughput.

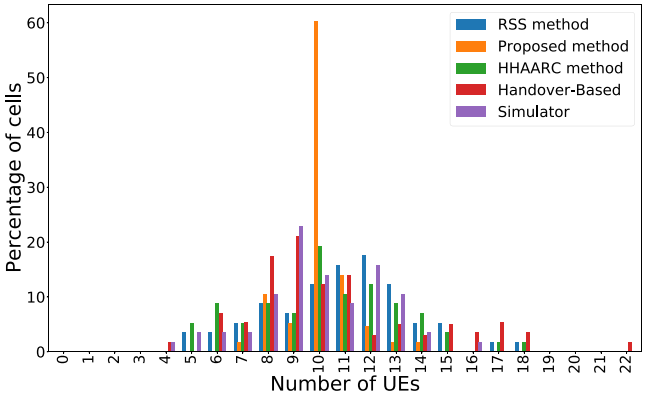

FIGURE 13. The graph shows by percentage the number of users attached to the individual cells.

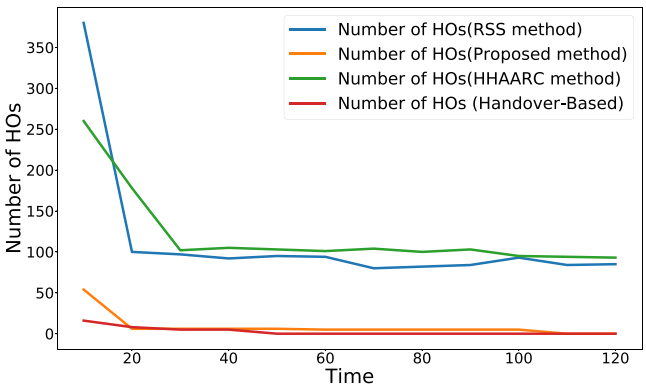

FIGURE 14. This graph presents the number of handovers triggered in the different scenarios.

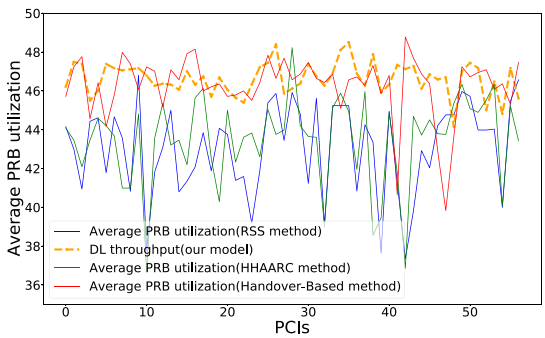

FIGURE 15. The PRB utilization per each cell.

4) HANDOVER AND PRB UTILIZATION ANALYSIS

Fig. 14 illustrates the handovers triggered by each method. When considering the total number of handovers across all four cases, the RSS and the HHAARC methods exhibited the highest number of handovers, while the Handover-Based method generated the fewest HO events. This highlights the stringent nature of the Handover-Based method, which employs a fixed threshold for handovers. However, this approach did not significantly contribute to UE distribution, with only slight improvements in throughput compared to the other methods. In contrast, our method introduces a degree of flexibility by thoroughly evaluating the system before considering a UE acceptable for handover.

We analyzed the resource utilization of all the presented approaches under high-traffic conditions in Fig. 15. As expected, due to the large number of UEs simulated in this experiment resource consumption is generally high across all approaches. In our approach, we observe minimal fluctuations in resource utilization across cells, resulting in an equitable distribution of resources. This contributes to the fair distribution of throughput observed in Fig. 11, where all cells maintain comparable performance levels.

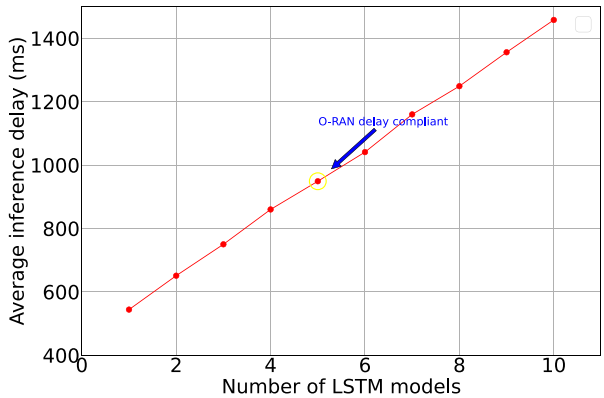

FIGURE 16. O-RAN compliant delay analysis.

C. INFERENCE DELAY ANALYSIS

The delay in transmitting information exchange is vital in understanding the performance of a proposed model. In this work, we combined multiple predictors to improve the performance of the LSTM model. However, the inference time increases as the number of models we have for the bagging increases. The observation made is that when we consider the accuracy of our predictions Fig. 7 and inference latency, we could only support 5 LSTM models to be O-RAN compliant as shown in Fig. 16. The delay metrics illustrate the total time from the data received at the E2 interface and the processing by the xApp. The base stations to be processed by the xApp depend on the classification made by the SVM model. Although 10 LSTM models (bagging) give us the highest prediction accuracy, they exceed the delay requirement of O-RAN. A good compromise is 5 models highlighted in Fig. 16.

While Fig. 17 demonstrates that our method has higher power consumption compared to others, this is due to its use of two machine learning techniques (SVM and LSTM) for UE classification and downlink (DL) throughput prediction. These functionalities introduce additional computational costs. The energy is computed using the linear model presented in [63]. However, it's important to note that optimizing power consumption is not the primary focus of this work. Future research will explore strategies to address this aspect and potentially introduce trade-offs between

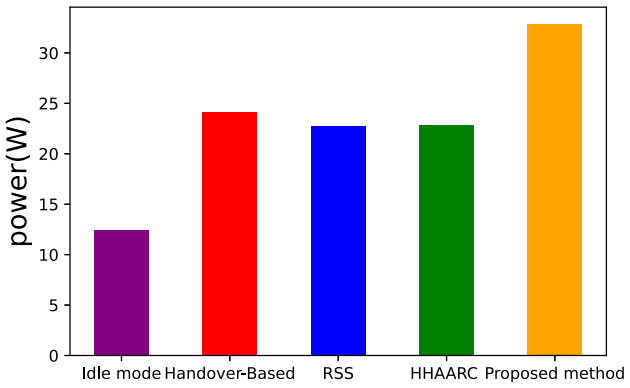

FIGURE 17. Power utilization by the various methods considered in this work.

power efficiency and the benefits provided by the proposed method.

VI. CONCLUSION

The O-RAN architecture unlocks numerous possibilities for applying AI/ML techniques to enhance network performance, particularly in optimizing UE throughput and load balancing to meet network operator KPIs. One key problem is the difficulty of making optimal traffic steering decisions due to limitations in network observability mostly in congested network areas. With this motivation, we presented a two-tier ML-aided traffic steering method for throughput optimization and load balancing. We employ the SVM method for UE classification, identifying potential candidates for handover (HO). This not only provides valuable insights into user QoS and QoE but also aids in resource allocation decisions. Understanding the average cell throughput experienced by UEs is crucial for informed handovers. Therefore, we use an ensemble Long Short-Term Memory (E-LSTM) model to predict the throughput a UE can expect in a cell before initiating handovers. This prediction allows for proactive resource management and improves overall network efficiency.

To achieve an even UE distribution throughout the network while maintaining acceptable individual throughputs, we present a traffic steering algorithm. This algorithm utilizes the insights from UE classification and cell throughput prediction to make informed decisions about handovers and optimize network load. The proposed approach achieves an acceptable and even distribution of UE throughput across different cells. The percentage of cells hosting the ideal number of UEs (10 in this case) significantly increases, from 14.03% to 60.25%, successfully aligning with our experimental goals.

Analyzing the inference delay provided insight into the number of models needed for the bagging technique to comply with O-RAN specifications.

The proposed method efficiently achieves load balancing with a minimal number of HOs, minimizing potential disruptions to user experience. This three-step approach, leveraging AI/ML, demonstrates its effectiveness in enhancing network performance through improved load balancing and UE throughput optimization, ultimately contributing to better network management and user experience.

REFERENCES

[1] P. Rost et al., "Cloud technologies for flexible 5G radio access networks," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 68–76, May 2014.
[2] "C-RAN: The road towards green RAN, Version 1.0.0," White Paper, China Mobile Res. Inst., Beijing, China, 2011.
[3] M. Hadzialic, B. Dosenovic, M. Dzaferagic, and J. Musovic, "Cloud-RAN: Innovative radio access network architecture," in *Proc. ELMAR*, 2013, pp. 115–120.
[4] S. Khatibi and L. M. Correia, "A model for virtual radio resource management in virtual RANs," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, pp. 1–12, 2015.
[5] M. Yang, Y. Li, D. Jin, L. Su, S. Ma, and L. Zeng, "OpenRAN: A software-defined RAN architecture via virtualization," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 549–550, 2013.
[6] "O-RAN: Towards an open and smart RAN," White Paper, O-RAN ALLIANCE e.V, Alfter, Germany, 2018.
[7] B. Brik, K. Boutiba, and A. Ksentini, "Deep learning for B5G open radio access network: Evolution, survey, case studies, and challenges," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 228–250, 2022.
[8] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 1376–1411, 2nd Quart., 2023.
[9] M. Dryjański, Ł. Kułacz, and A. Kliks, "Toward modular and flexible open RAN implementations in 6G networks: Traffic steering use case and O-RAN xApps," *Sensors*, vol. 21, no. 24, p. 8173, 2021.
[10] Y. Cao, S.-Y. Lien, Y.-C. Liang, K.-C. Chen, and X. Shen, "User access control in open radio access networks: A federated deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 3721–3736, Jun. 2022.
[11] X. Wang, J. Li, L. Wang, C. Yang, and Z. Han, "Intelligent user-centric network selection: A model-driven reinforcement learning framework," *IEEE Access*, vol. 7, pp. 21645–21661, 2019.
[12] H. Zhang et al., "On-device intelligence for 5G RAN: Knowledge transfer and federated learning enabled UE-centric traffic steering," *IEEE Trans. Cogn. Commun. Netw.*, vol. 10, no. 2, pp. 689–705, Apr. 2024.
[13] A. Hatipoğlu, M. Başaran, M. A. Yazici, and L. Durak-Ata, "Handover-based load balancing algorithm for 5G and beyond heterogeneous networks," in *Proc. 12th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, 2020, pp. 7–12.
[14] W. K. Saad, I. Shayea, A. Alhammadi, M. M. Sheikh, and A. A. El-Saleh, "Handover and load balancing self-optimization models in 5G mobile networks," *Int. J. Eng. Sci. Technol.*, vol. 42, Jun. 2023, Art. no. 101418.
[15] I. Tamim, S. Aleyadeh, and A. Shami, "Intelligent O-RAN traffic steering for URLLC through deep reinforcement learning," in *Proc. IEEE Int. Conf. Commun.*, 2023, pp. 112–118.
[16] R. Ntassah, G. M. Dell'Aera, and F. Granelli, "xApp for traffic steering and load balancing in the O-RAN architecture," in *Proc. IEEE Int. Conf. Commun.*, 2023, pp. 5259–5264.
[17] Y. Fu, H. Xiong, X. Lu, J. Yang, and C. Chen, "Service usage classification with encrypted Internet traffic in mobile messaging apps," *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2851–2864, Nov. 2016.
[18] D. Hu, F. Sun, L. Tu, and B. Huang, "We know what you are—A user classification based on mobile data," in *Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Social Comput.*, 2013, pp. 1282–1289.
[19] M. Yan, S. Li, C. A. Chan, Y. Shen, and Y. Yu, "Mobility prediction using a weighted Markov model based on mobile user classification," *Sensors*, vol. 21, no. 5, p. 1740, 2021.
[20] S. Alotaibi, A. Alruban, S. Furnell, and N. Clarke, *A Novel Behaviour Profiling Approach to Continuous Authentication for Mobile Applications*, Science Technol. Publ. (SCITEPRESS), Setúbal, Portugal, 2019.
[21] V. Labayen, E. Magaña, D. Morató, and M. Izal, "Online classification of user activities using machine learning on network traffic," *Comput. Netw.*, vol. 181, Nov. 2020, Art. no. 107557.

[22] L. Gabrielli, B. Furletti, R. Trasarti, F. Giannotti, and D. Pedreschi, "City users' classification with mobile phone data," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2015, pp. 1007–1012.
[23] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, "A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1790–1821, 3rd Quart., 2017.
[24] J. Xie et al., "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 393–430, 1st Quart., 2019.
[25] C. Yue, R. Jin, K. Suh, Y. Qin, B. Wang, and W. Wei, "LinkForecast: Cellular link bandwidth prediction in LTE networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 7, pp. 1582–1594, Jul. 2018.
[26] J. Yao, S. S. Kanhere, and M. Hassan, "Improving QoS in high-speed mobility using bandwidth maps," *IEEE Trans. Mobile Comput.*, vol. 11, no. 4, pp. 603–617, Apr. 2012.
[27] F. Jomrich, A. Herzberger, T. Meuser, B. Richerzhagen, R. Steinmetz, and C. Wille, "Cellular bandwidth prediction for highly automated driving," in *Proc. 4th Int. Conf. Vehicle Technol. Intell. Transp. Syst. (VEHITS)*, 2018, pp. 121–132.
[28] D. Raca et al., "On leveraging machine and deep learning for throughput prediction in cellular networks: Design, performance, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 3, pp. 11–17, Mar. 2020.
[29] J. Schmid, M. Schneider, A. HöB, and B. Schuller, "A deep learning approach for location independent throughput prediction," in *Proc. IEEE Int. Conf. Connected Veh. Expo (ICCVE)*, 2019, pp. 1–5.
[30] A. Mostafa, M. A. Elattar, and T. Ismail, "Downlink throughput prediction in LTE cellular networks using time series forecasting," in *Proc. Int. Conf. Broadband Commun. Next Gener. Netw. Multimedia Appl. (CoBCom)*, 2022, pp. 1–4.
[31] Y. Hou et al., "A study of throughput prediction using convolutional neural network over factory environment," in *Proc. 23rd Int. Conf. Adv. Commun. Technol. (ICACT)*, 2021, pp. 429–434.
[32] C. Adamczyk and A. Kliks, "Reinforcement learning algorithm for traffic steering in heterogeneous network," in *Proc. 17th Int. Conf. Wireless Mobile Comput. Netw. Commun. (WiMob)*, 2021, pp. 86–89.
[33] C. Gijón, M. Toril, S. Luna-Ramírez, and M. L. Marí-Altozano, "A data-driven traffic steering algorithm for optimizing user experience in multi-tier LTE networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 9414–9424, Oct. 2019.
[34] S. Math, P. Tam, and S. Kim, "Intelligent real-time IoT traffic steering in 5G edge networks," *Comput., Mater. Continua*, vol. 67, no. 3, pp. 3433–3450, 2021.
[35] S. Zhang, N. Zhang, S. Zhou, J. Gong, Z. Niu, and X. Shen, "Energy-sustainable traffic steering for 5G mobile networks," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 54–60, Nov. 2017.
[36] F. Kavehmadavani, V.-D. Nguyen, T. X. Vu, and S. Chatzinotas, "Traffic steering for eMBB and uRLLC coexistence in open radio access networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2022, pp. 242–247.
[37] F. Kavehmadavani, V.-D. Nguyen, T. X. Vu, and S. Chatzinotas, "Intelligent traffic steering in beyond 5G open RAN based on LSTM traffic prediction," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 7727–7742, Nov. 2023.
[38] S. De Lucia, *Optimization of Handover Algorithms for Wireless Networks*, KTH Royal Inst. Technol., Stockholm, Sweden, 2010.
[39] I. Shayea, M. Ismail, and R. Nordin, "Advanced handover techniques in LTE- advanced system," in *Proc. Int. Conf. Comput. Commun. Eng. (ICCCE)*, 2012, pp. 74–79.
[40] J. Feng and Z. Feng, "Optimal base station density of dense network: From the viewpoint of interference and load," *Sensors*, vol. 17, no. 9, p. 2077, 2017.
[41] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 144–152.
[42] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul./Aug. 1998.
[43] W. S. Noble, "What is a support vector machine?" *Nat. Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, 2006.
[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[45] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1310–1318.
[46] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, 1998.
[47] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM—A tutorial into long short-term memory recurrent neural networks," 2019, *arXiv:1909.09586*.
[48] B. Wei, W. Kawakami, K. Kanai, J. Katto, and S. Wang, "TRUST: A TCP throughput prediction method in mobile networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2018, pp. 1–6.
[49] H. Elsherbiny, H. M. Abbas, H. Abou-zeid, H. S. Hassanein, and A. Noureldin, "4G LTE network throughput modelling and prediction," in *Proc. IEEE Global Commun. Conf.*, 2020, pp. 1–6.
[50] C. Olah, "Understanding LSTM networks," 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[51] T. G. Dietterich et al., "Ensemble learning," in *The Handbook of Brain Theory and Neural Networks*, vol. 2. Cambridge, MA, USA: MIT Press, 2002, pp. 110–125.
[52] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdiscip. Rev. Data Min. Knowl. Disc.*, vol. 8, no. 4, 2018, Art. no. e1249.
[53] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[54] G. G. Lema, "Handover performance evaluation under dynamic user characteristics," *Wireless Pers. Commun.*, vol. 120, no. 4, pp. 2991–3004, 2021.
[55] "Coordinated multi-point operation for LTE physical layer aspects (release 11)," 3GPP, Sophia Antipolis, France, Rep. TR-36.819, 2011.
[56] "Study on channel model for frequencies from 0.5 to 100 GHz," 3GPP, Sophia Antipolis, France, Rep. TR 138 901, 2017.
[57] "E-UTRA: Further advancements for E-UTRA physical layer aspects," 3GPP, Sophia Antipolis, France, Rep. 36.814, 2010.
[58] F. Afroz, R. Subramanian, R. Heidary, K. Sandrasegaran, and S. Ahmed, "SINR, RSRP, RSSI and RSRQ measurements in long term evolution networks," *Int. J. Wireless Mobile Netw.*, vol. 7, no. 4, pp. 113–123, 2015.
[59] G. Basilashvili, "Study of spectral efficiency for LTE network," *Amer. Acad. Sci. Res. J. Eng., Technol., Sci.*, vol. 29, no. 1, pp. 21–32, 2017.
[60] M. Anas, F. D. Calabrese, P. E. Mogensen, C. Rosa, and K. I. Pedersen, "Performance evaluation of received signal strength based hard handover for UTRAN LTE," in *Proc. IEEE 65th Veh. Technol. Conf.*, 2007, pp. 1046–1050.
[61] Y. Li, B. Cao, and C. Wang, "Handover schemes in heterogeneous LTE networks: Challenges and opportunities," *IEEE Wireless Commun.*, vol. 23, no. 2, pp. 112–117, Apr. 2016.
[62] C.-C. Lin, K. Sandrasegaran, H. A. M. Ramli, and R. Basukala, "Optimized performance evaluation of LTE hard handover algorithm with average RSRP constraint," 2011, *arXiv:1105.0234*.
[63] S. S. Tadesse, F. Malandrino, and C.-F. Chiasserini, "Energy consumption measurements in docker," in *Proc. IEEE 41st Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 2, 2017, pp. 272–273.

Open Access funding provided by 'Università degli Studi di Trento' within the CRUI CARE Agreement