ML-based Traffic Steering for Heterogeneous Ultra-dense beyond-5G Networks

Ilias Chatzistefanidis*, Nikos Makris*†, Virgilios Passas*† and Thanasis Korakis*†
*Dept. of Electrical and Computer Engineering, University of Thessaly, Greece
†Centre for Research and Technology Hellas, CERTH, Greece
Email: ichatsist@uth.gr, nimakris@uth.gr, vipassas@uth.gr, korakis@uth.gr

2023 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-6654-9122-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/WCNC55385.2023.10118923

***Abstract*—As networks become denser and more heterogeneous different paths can be considered in order to reach each multi-homed UE, offering optimal performance. 5G and beyond networks feature contributions related to the dynamic programming of the network, from the operator side, in order to optimally allocate resources in the network. In this work, we consider such a case, where network access is provided to the end-users via heterogeneous (3GPP and non-3GPP) Distributed Units (DUs), converging to a single Central Unit (CU), and programmable on the fly with external interfaces. We employ Machine Learning (ML) methods in order to forecast the Quality of Service (QoS) that a wireless client will get from the network in the near future based on the Channel State Information (CSI) metric. Subsequently, we appropriately steer the traffic over the different heterogeneous DUs for ensuring that the network meets the needs of the UEs. We design, develop, deploy and evaluate our method in a real testbed environment, using emulated mobility. Our results show that the overall throughput of each UE can be drastically improved compared to existing allocation mechanisms.**

***Index Terms*—5G, Artificial Intelligence, Disaggregated RAN, HetNets, Neural Networks, Traffic Steering**

I. INTRODUCTION

5G networks introduce a wide set of new functionalities that add up to the flexibility for the network provider. Through the adoption of a virtualized system architecture and the execution of the network as cloud-native functions, assisted via the control/user plane disaggregation, the operators can have fine grained control over different parameters of the network during its lifecycle. This creates fertile ground for the further enhancement of the 5G network, through resource allocation decisions that take place dynamically during the network operation, meeting the needs of the user demand.

Such approaches are commonly supported with dedicated and standardized APIs for programming and reconfiguring the Radio Access Network (RAN), through applications hosted on the edge of the network (*xApps*) [1]. Measurement collection and network programming relies on the recent advancements of the O-RAN architecture for beyond 5G networks, through the definition of interfaces for the different layers of the stack (e.g. the A1/E2 interfaces of the O-RAN architecture). *xApps* monitor and analyze metrics collected from network functions (base station or core network), and conclude on the resource allocation. In this way, the network is considered an elastic resource, constantly adapting to meet the needs of its users. Given the fact that applications are offered over heterogeneous networks, network adaptation should span multiple technologies (e.g. traffic steering over multiple links [2]).

Such interfaces provide the capability to host applications which take advantage of these metrics in real-time, thus creating fertile ground for the efficient forecasting of load, based on historical data. ML can assist in the effective prediction of metrics that reflect the load or QoS of a client, so as to appropriately and proactively apply policies that improve the overall network behaviour. In this work, we deal with the case of sustaining the QoS that mobile clients are getting, when using multiple heterogeneous links (3GPP and non-3GPP) for network service. By extracting low-level MAC statistics from the telecom network, we conclude through ML on the mobility patterns of the users, and appropriately steer traffic to them through multiple links. In this manner, QoS is preserved regardless of the network conditions in different trajectories. The main contributions of the paper are the following:

- To proactively determine the best strategy for serving users through multiple links, in order to increase their QoS, with a dedicated *xApp* running on top of the network.
- To effectively infer with ML the QoS that a mobile client is getting, in different trajectories.
- To experimentally evaluate the contributions in a real environment, using realistic datasets and emulated mobility.

We use the Channel Quality Indicator (CQI) as the predicted metric that determines the per client QoS. CQI is a metric reported from the UEs to base stations to assist in the resource allocation on the MAC layer. We use the OpenAirInterface [3] platform for the network, and the FlexRAN controller [4] for collecting MAC level statistics for our *xApp*. As our ML solution, we use a Bidirectional Long Short-Term Memory (Bi-LSTM) stacked Recurrent Neural Network (RNN) model.

The rest of the paper is organized as follows. Section II provides an overview of related literature. In Section III, we detail the system architecture, ML model, and mobility emulation framework. In Section IV we show our findings and in Section V we conclude.

II. RELATED WORK

Augmenting the network with intelligence is a key characteristic for beyond 5G solutions, as it enables the automated provisioning and proactive resource allocation. For instance, in

The research leading to these results has received funding from the European Horizon 2020 Programme for research, technological development and demonstration under Grant Agreement Number No 101008468 (H2020 SLICES-SC). The European Union and its agencies are not liable or otherwise responsible for the contents of this document; its content reflects the view of its authors only.

[5], authors explore the potential of Artificial Intelligence (AI)-based approaches in the context of 5G mobile and wireless communications, evaluating the different challenges and open issues for future research. Authors in [6] group the different approaches based on data sources, and the role of ML/AI in making the networking systems intelligent through proactive and prescriptive self-awareness and self-adaptation, with the aim to maximize the overall QoS. In [7], the authors employ a cloud-native approach for existing networks and create a closed-control loop framework for AI-based network and service management, with respect to performance KPIs. As the wireless network is susceptible to performance fluctuations due to client mobility, several works focus on optimizing the channel allocation, using ML. For instance, authors in [8] focus on an autoencoder for feature extraction from Channel State Information (CSI), and use ML (logical regression) to decide the modulation and coding schemes. In [9], resource balancing is studied among primary/secondary users in a cognitive radio environment.

Key variables in all the allocation problems are the monitored parameters that reveal the true QoS that each user experiences. Most of the works, focus on CSI information reported by clients. In [10], authors employ ML for forming the CSI, including parameters like UE context, signal-to-interference-plus-noise ratio (SINR), and delay. In [11], CSI is estimated using Neural Network based ML methods, predicting the per client signal to noise ratio.

As networks become denser, multiple-paths for serving each client might be available for the network operator, and can be included in the allocation process. In [12], authors select the path with the higher estimated throughput. In [13], a data driven algorithm for traffic steering is proposed, using as metrics the Reference Signal Received Quality (RSRQ) from the network. In [14], authors propose an adaptive cell selection scheme in ultra-dense heterogeneous environments, showing that the QoS of moving vehicles is highly affected. Based on these, it is evident that the technology selection for traffic steering affects the overall performance experienced by wireless clients. Nevertheless, none of these works assumes convergence of heterogeneous technologies, but merely the presence of an external intelligent controller that manages the steering process.

In [15], we introduce such heterogeneity in the telecom network, by integrating non-3GPP access technologies to the LTE base station architecture. The solution is disaggregated, relying on the functional splitting of the base station stack using the 3GPP Option-2 split [16], and augments the network with heterogeneous links used for serving multi-homed UEs. As the communication between the CUs and DUs relies on stateful protocols, data flows between the two technologies are ensured to deliver traffic in an orderly manner. In this work, we use the same architecture and determine the optional splitting of the traffic among the heterogeneous links, subject to client mobility. This is one of the first works, to the best of our knowledge, to address the steering process from a common convergence point, inside the 3GPP network.

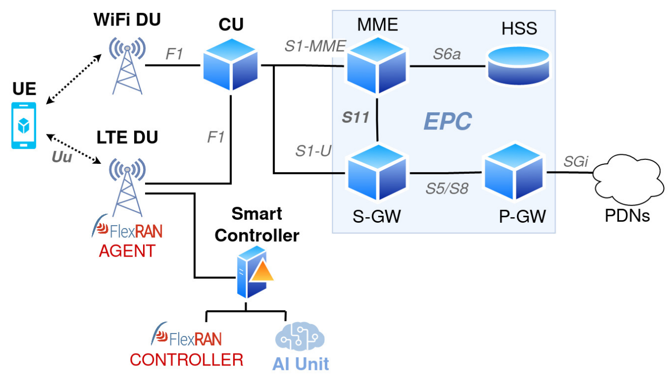


Fig. 1: The 5G Disaggregated Architecture for real-time steering over heterogeneous technologies.

Our work is considering the downlink channel and is applied from the perspective of the network operator at the base station level. Towards determining the optional split of the traffic, we retrieve MAC layer statistics for each UE, as perceived from the base station side. We rely on the CQI reported by each UE for the 3GPP network and by employing ML methods, we conclude on the split of traffic among the different technologies, for preserving the QoS of the connected clients. In the next section, we detail our system architecture and the choice of ML method that produces our predictions.

III. System Architecture

Towards developing an efficient real-time scheduling and traffic steering mechanism for heterogeneous technologies in the disaggregated heterogeneous RAN architecture, we leverage our prior work [15] based on the OpenAirInterface [3] and FlexRAN [4] platforms. Figure 1 is the main reference point depicting the functional architecture used to build our framework. The heterogeneous access is enabled through non-3GPP (WiFi) DUs that are integrated in the architecture, communicating with the CU. At first, we develop a scheduling technique on the CU side, relying on the FlexRAN internal communication (Controller/Agent). We introduce intelligence through a novel ML & AI unit responsible for the live monitoring and forecasting of the channel quality based on the reported CQI values from the UEs. Our target is the analysis of CQI values from mobile stations (car routes obtained from real commercial networks). Thus, we obtain and analyze such data from the city of Volos in Greece, where the experimental facilities are located. Specifically, we reproduce the real car routes inside a complete realistic experimental infrastructure, the NITOS testbed [17]. The NITOS Testbed includes wireless devices (Software Defined Radios, UE terminals) that are utilized for an efficient validation and evaluation of our framework. Below we list the key features that enable traffic steering and scheduling over heterogeneous DUs, seamlessly resulting in higher QoS to the end-user.

A. Management and deployment of the network functions

As a baseline topology, we utilize the 5G disaggregated architecture built with the OpenAirInterface platform. Specifically, the functional split occurs at the higher OSI stack layer 2, between Packet Data Convergence Protocol (PDCP) and Radio Link Control (RLC) layers of the base station, creating the CU and DU components, interconnected over an Ethernet-based fronthaul [18], as shown in Figure 1. The CU includes the PDCP and above layers, as well as the interface towards the Core Network, while the DU consists of the RLC and lower MAC and PHY layers. The communication between CU and DU is based on the F1 Application Protocol (F1AP) via the F1 interface. For the core network and the base station, we employ the LTE implementations of the OpenAirInterface platform due to its stability compared to the 5G-NR version at the time of writing. Nevertheless, the solution can be directly projected to the 5G implementation by interchanging the core network components with the 5G ones (HSS/UDM, MME/AMF, SPGW-U/UPF, SPGW-C/SMF) and the disaggregated eNB with a disaggregated gNB. Moreover, we use LTE and WiFi DUs to support a heterogeneous network connection to a multi-homed commercial UE. Finally, we utilize the FlexRAN platform to create the mechanism for the real-time scheduling of the DUs based on its internal slicing communication between the FlexRAN Controller and the Agent; the latter is integrated into the LTE DU.

B. Traffic Steering Mechanism Implementation

To develop an effective steering scheme for multiple technologies, we consider the CU as the coordinator, that selects the traffic load for every DU, leveraging the FlexRAN's slicing technique. Specifically, our goal is to extend slicing support, based on the resource block allocation for the 3GPP network, and steer traffic over different DUs, by a controller-defined percentage. To accomplish this, we parse the slice values (UE id and percentage) as it arrives at the DU, and develop new messages to be forwarded to the CU over the F1 interface.

The scheduling mechanism is shown in Figure 2 and is operating in a Round-Robin fashion. In particular, there are one hundred slots available for every round, each one representing 1% and used to send one packet. Every DU exploits a number of slots to transmit packets to UEs, based on the percentage value. The LTE DU is assigned initially a number of slots equal to the percentage value. Subsequently, the WiFi DU is assigned the remaining slots (100 - percentage). For instance, if the posted percentage in slice equals 60%, the LTE DU will be allocated the first 60 slots, while the WiFi DU will transmit the remaining 40 for every round. Thus, the scheduler instructs the CU to forward 60% of the downlink traffic via the LTE DU and the remaining 40% through the WiFi DU.

C. AI-Driven Architecture

Towards developing a real-time scheduling framework adaptive to the network fluctuations, ensuring the optimal network performance and QoS, we build an ML & AI unit that is able to infer on the future channel conditions. As depicted in Figure 1, it is constructed on the FlexRAN controller side for direct slicing management. Its role is to continuously monitor the channel quality by collecting CQI values. In LTE systems, CQI is reported from the UEs for assisting the allocation of modulation and coding schemes. Especially, it ranges from 0 to 15 in its value. This translates from no to 64QAM modulation, from zero to 0.93 code rate, from zero to 5.6 bits per symbol, from less than 1.25 to 20.31 SINR (dB) and from zero to 3840 Transport Block Size bits. Subsequently, the ML unit identifies patterns in the CQI data and forecasts near-future values continuously. Based on the predicted CQI values, the unit makes appropriate scheduling decisions based on a slicing allocation algorithm. Specifically, it uses the aforementioned scheduling mechanism to configure the DU scheduler to ensure enhanced end-user experience. For example, when poor LTE quality is forecast, the scheduler directs a higher traffic percentage via the WiFi DU, as long as it has better link conditions.

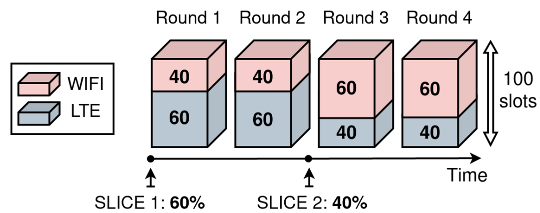


Fig. 2: Round-robin Scheduling Technique for the DUs.

D. Data Management & Analysis

This work is focused on CQI data from car routes in real networks. Below we analyze the steps followed from acquiring the data to preparing them for the model. More specifically, we describe the collected car route data and how we reproduce them in the testbed. Moreover, we explain the creation of traffic scenarios based our data, and their augmentation for emulating multiple cars traversing a specific city pathway. Finally, we present the methodology for collecting the CQI values of the emulated car routes in the testbed, and the data pre-processing for feeding them into the model.

1) Car Route Attenuation Data & Reproducibility: To emulate a realistic car route inside the testbed, we install programmable attenuators at the outputs of the Software-Defined Radio (SDR) devices. By modifying the antenna attenuation, we emulate mobility to any connected user. Specifically, we employ attenuation scenarios where the SDR attenuation is configured in reflecting actual car routes. We deploy several attenuation scenarios to reproduce many different cars traversing a specific pathway, parsing and collecting the CQI values. Noticeably, CQI is inversely proportional to attenuation; high attenuation values result in low CQI and vice versa.

2) Basic Attenuation Scenario: Towards simulating the monitoring of a specific city road that many cars traverse, we build on top of a basic attenuation scenario from real car routes. Understandably, all the cars should have a similar CQI/attenuation patterns as they travel through the same specific geographical area. Nevertheless, slight variations from car to car are foreseen, based on the driving style, the noise, and the actual UE chipset that is employed. Thus, the available scenario aims to provide the basic pattern of the cars traveling through this city pathway.

3) Attenuation Data Augmentation: We apply data augmentation techniques to create multiple attenuation scenarios representing all the cars of the pathway. Specifically, we utilize

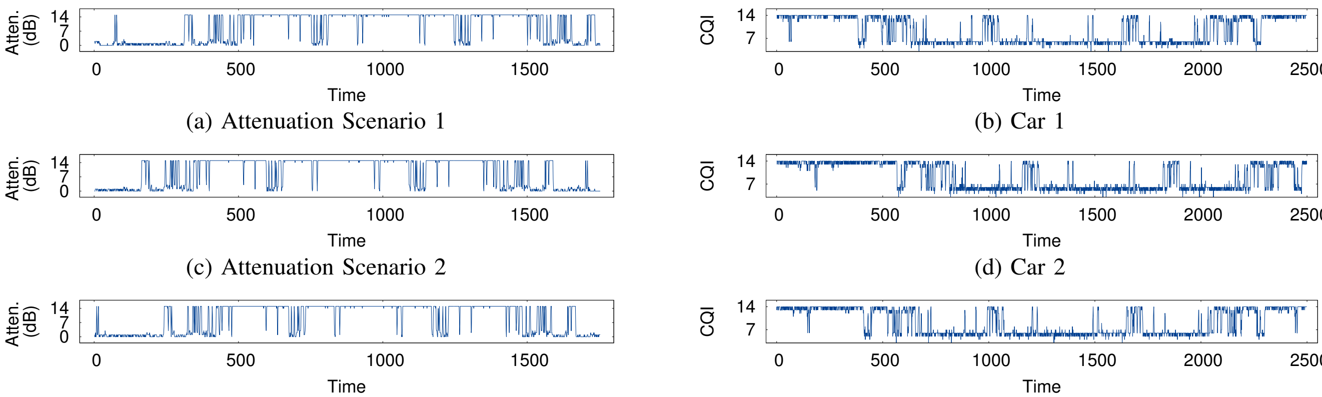


Fig. 3: Attenuation scenarios and collected CQI series.

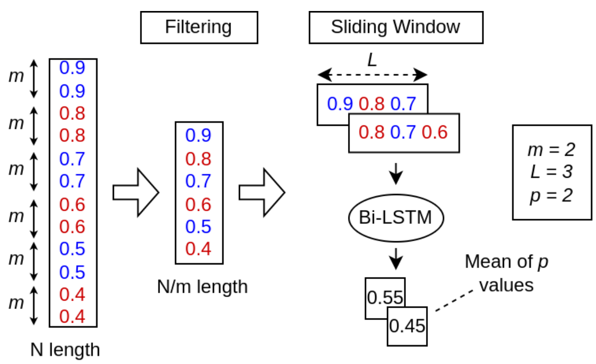


Fig. 4: Pre-processing Example with $m = 2$, $L = 3$, $p = 2$.

the Tsaug python [19] library to apply various augmenters. First, we use Dynamic Time Warping (DTW) to emulate cars with different speeds. We create cars that randomly change their speed multiple times during the driving (approx. every $1\ km$). Each augmented car reaches several points of the pathway with time differences ranging from 0 to 30 seconds from the basic car. Given these, and since the basic car is traversing with $50\ km/h$ (velocity limit), the augmented car scenarios use speeds in the range of 40 to 60 km/h. Moreover, Additive White Gaussian Noise (AWGN) with small variance is added to represent the various road conditions and chipset of the cars. Further, some scenarios are slightly cropped and scaled for more efficient training. This way, we obtain slightly altered scenarios based on the basic one, representing the different parameters of every car, as shown in Figure 3.

4) CQI Data Collection: With multiple car attenuation scenarios, we proceed to the CQI collection. We randomly choose one scenario and deploy it in our experimental topology. Meanwhile, we continuously parse the CQI values with the ML & AI unit, by periodically sending requests to the FlexRAN controller, inquiring about the base station statistics per 250 msecs. Importantly, this is also the prediction period, since the unit infers each time a new CQI value is obtained.

In total, we deployed 73 car-route scenarios, each providing about 2500 CQI values. Thus, we collect 73 univariate time series sequences with around 182.500 CQI values. Figure 3 shows three of them, showing slight changes from car to car.

5) Pre-processing: Before feeding the data into our model, it is essential to pre-process them. Initially, as shown in Figure 4, we normalize them by rescaling them in the range $[0, 1]$ to boost model training efficiency. Subsequently, we utilize a filtering technique to reduce the volume of the data by collecting the means of data batches (mean of every m values). In particular, we calculate and store the mean of every 5 CQI values. In this way, we end up with a sequence that is five times smaller but contains the same information. Following this, we implement a sliding window approach to convert the time-series forecasting problem into a supervised learning one by cropping the huge filtered sequence into multiple sub-sequences (X_i) by sliding continuously by one CQI. At this point, it is essential to choose the input (X_i) and output (y_i) shape of the model based on how the AI unit function. We concluded that the model will predict the average value of multiple (p) future CQIs and not only the next CQI value in the future. In this way, the model's predictive performance is resilient to fluctuations and outliers, and provides the general figure of the CQI values in the near-future. Taking this into account, we pre-process the output data (y_i labels) as single values that depict the average CQI for the future 17.5 seconds; that is 14 filtered values that are placed after every X_i sub-sequences. The CQI labels (y_i) of a car scenario after the pre-processing are illustrated in Figure 5. Regarding the shape of the X_i sub-sequences (L), we understand that it is vital to create input windows large enough to identify the pattern in the data, but also sufficiently small in order to bolster training time and avoid exploding/vanishing gradients; a common issue in RNN models. Thus, after extensive experimentation with various shapes, we conclude that the optimal window is to utilize 80 filtered CQI values (100 seconds) for every X_i sub-sequence. In this way, the model forecasts per 250 msecs the mean CQI for the future 17.5 secs, by analyzing the pattern in the CQI data of the past 100 secs. The data specifications are provided in detail in Table I. Figure 4 shows an example of the pre-processing procedure using $m = 2, L = 3, p = 2$.

E. Machine Learning Model

We examined our options for a deep learning model carefully, and we opted to employ a Bi-LSTM RNN utilizing

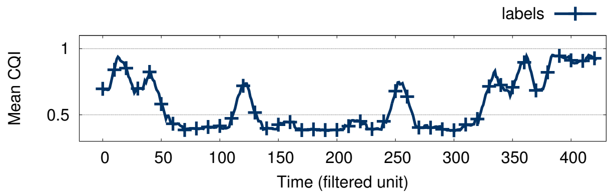


Fig. 5: Car Scenario's CQI Labels (y_i) after pre-processing.

the Tensorflow-Keras API for the real-time CQI predictions. For our decision, we considered the robust nature of LSTM NNs to cope with Time Series Forecasting (TSF) problems by employing sophisticated memory components (forget, input, output gates, cell state). In this way, they often outcompete conventional RNNs by identifying the pattern in data more accurately and usually avoiding the exploding/vanishing gradients issue. Moreover, adding bidirectional layers is linked with higher predictive performance, as it provides an additional reversed sequence learning. In fact, the key point is to combine both the reversed and original sequences, to get insight about the past as well as the future concurrently. This technique is enhancing LSTM RNNs, leading to a more powerful and adaptive model. After extensive hyperparameter tuning, we employ a stacked model of two *Bi-LSTM* layers, both utilizing 25 $neurons$ and a $ReLU$ activation function. Importantly, we recommend using at least two hidden layers to identify non linear CQI patterns. Finally, we add a $Dense$ layer with an output unit, and use the $Adam$ optimizer and the *Mean Squared Error* monitoring function (Table I).

F. Steering Algorithm

The steering scheme is responsible for obtaining the predictions of the AI unit, and appropriately steering the downlink traffic via the DUs so as to ensure high end-user QoS. Since we focus on the 3GPP part of the solution, we perform all of our experiments with excellent link quality for the WiFi DU. The steering algorithm dynamically adjusts the downlink path in real-time, ensuring optimal network performance. Specifically, the steering mechanism selects the DU based on a CQI threshold (CQI = 9). This means that while the average CQI of the future 17.5 secs is expected to be higher than 9 (16-QAM, 8.75 SINR), the scheme steers all the traffic via the LTE DU, as the link quality suffices. On the other side, when the mean CQI is about to drop below 9 in the near future, the scheme steers the traffic via the WiFi DU.

IV. Evaluation

To validate and evaluate our implementation under realistic settings, we utilize the NITOS experimental testbed [17]. In specific, we begin by assessing the deep learning model's predictive accuracy and training requirements. As evaluation metric, we utilize the *Mean Absolute Error (MAE)*. Noticeably, MAE depends on the data scale, and in our case the CQI scale $(0, 15]$. This helps us understand easily the error performance of the model. Subsequently, we test the complete ML & AI unit in the testbed, evaluating its contribution in enhancing the overall network performance and user QoS by continuously forecasting the CQI and reconfiguring the DU steering. Importantly, the basic non-augmented car scenario is employed for

TABLE I: Data & Model Specifications

Data-specific	Information
Collection/Prediction Round	Every 0.25 secs
Filtering Window (m)	5 non-filtered values
Scenario Duration	11 mins
Number of Scenarios (cars)	73
Input X_i Time-slots (L)	80 (filtered values)
Output y_i Time-slots	1 (mean of 14 filtered future values)
Training Input X Shape	(samples, L, 1) = (36269, 80, 1)
Training Labels y Shape	(samples, 1) = (36269, 1)
Hyper-parameter	**Configuration**
Layers	2 Stacked Bi-LSTM + Dense
Input Shape	(80, 1)
Bi-LSTM Neurons	25 for both Bi-LSTM layers
Dense Units	1
Activation Function	$ReLU$ for both Bi-LSTM layers
Optimizer	$Adam$
Compile loss	Mean Squared Error
Epochs	55
Batch Size	2^6
Evaluation	**Values**
Time-series CV	MAE: 0.16
Experiment	MAE: 0.157
Training Time	1 hour with TPU on Google Colab

the evaluation experiments and was excluded from the training data. The number of different scenarios and data used for predicting the network performance is large enough to justify the model's generalization for several other scenarios than the ones that we evaluate in this work.

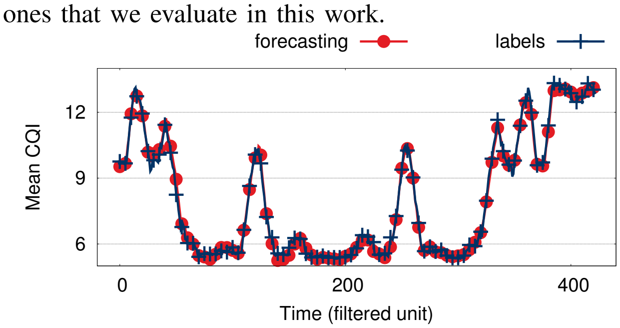
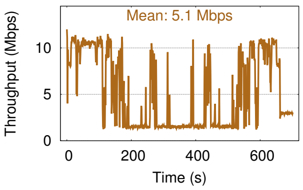


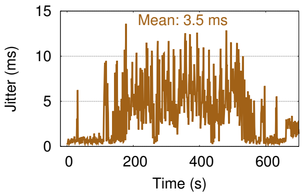
Fig. 6: CQI Forecasting in Testbed (original CQI scale).

Initially, we employ a 2-step evaluation process for the model. First, we utilize a time-series cross-validation (CV) scheme that is based on K-fold CV, but is not violating the time sequence. Specifically, the training data (36269 pre-processed X_i samples collected from the augmented attenuation scenarios) are split into multiple folds of a predefined length (≈ 500 samples). We form two sets, a training and a validation one. First, the training set is initialized with CQI data from approx. 50 cars (about 25000 samples). On every iteration, we add the next fold to the training set and use its following one as validation fold. Each time we fit the model to the training set and evaluate its generalization on the validation set (unseen data). At the end, we obtain the mean validation error from all validation sets (approx. 23 cars) to find the overall generalization error. The model generalizes remarkably well, identifying accurately the pattern in the unseen data and by having a MAE of 0.16. This error is negligible as the CQI range is $(0, 15]$.

Subsequently, as a second step, we train the model in the complete training set (all samples from the augmented scenarios) and integrate it to the AI unit to evaluate the entire efficiency of the framework on the testbed using the basic non-

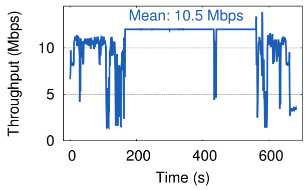


(a) Throughput in Time

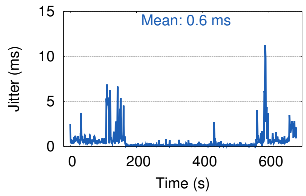


(b) Jitter in Time

Fig. 7: Default Steering



(a) Throughput in Time



(b) Jitter in Time

Fig. 8: AI-driven Steering

augmented attenuation scenario. Specifically, we design and execute two experiments. The first one, tests the QoS of the user and the network performance without utilizing the AI unit and label it as "default" network configuration since it is not aware of the link performance and thus it utilizes primarily the LTE link. On the other side, the second experiment's traffic is steered by the AI unit based on the LTE link condition (CQI predictions). The goal of the unit is to prioritize the LTE link under good link conditions ($CQI \geq 9$), and to redirect 100% of the traffic via the WiFi link when the LTE link conditions are deteriorating (poor CQI predictions, < 9) to prevent on time the QoS worsening. Figure 6 presents the CQI forecasting during the experiments showing outstanding performance with MAE of 0.157. The end-user experience is depicted in the Figures 7 and 8 and it is clear that the AI unit enhances substantially the QoS of the UE. In particular, it is evident that it predicts on time the deterioration of the LTE link at approx. 160 seconds in Figures 8a and 8b and thus, redirects the downlink traffic through the WiFi DU, which has excellent link quality, preventing the QoS plunge. Additionally, at approx. 550 seconds, the AI unit predicted on time that the LTE link quality will be enhanced and hence, steered the traffic back to the LTE DU.

V. Conclusion

In this work, we designed, developed, and evaluated a ML-driven approach for determining the optimal splitting and steering of traffic in a heterogeneous disaggregated ultra-dense network environment. A Bi-LSTM stacked model was employed for forecasting the CQI metric reported from the wireless clients of the network. Subsequently, a strategy for steering the traffic over the heterogeneous DUs was employed, for ensuring that QoS for the end-users is maximized. In the case that there is external interference on either wireless technologies, our algorithm appropriately steers the traffic percentage between them in order to ensure the highest achievable throughput. The tools and data contributions of this paper are also available online in [20]. In the future, we foresee extending our work towards determining a sophisticated approach for the dynamic traffic-steering policy among multiple DUs and extending the allocation technique towards including other decisions for the network (e.g. allocations in the transport network between DUs and CU, CU and 5G Core Network).

References

[1] M. Dryjanski, L. Kulacz, and A. Kliks, "Toward Modular and Flexible Open RAN Implementations in 6G Networks: Traffic Steering Use Case and O-RAN xApps," *Sensors*, 2021.

[2] I. A. Alablani and M. A. Arafah, "An Adaptive Cell Selection Scheme for 5G Heterogeneous Ultra-Dense Networks," *IEEE Access*, 2021.

[3] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "OpenAirInterface: A flexible platform for 5G research," *ACM SIGCOMM Computer Communication Review*, 2014.

[4] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. Kontovasilis, "FlexRAN: A flexible and programmable platform for software-defined radio access networks," in *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, 2016.

[5] M. E. Morocho Cayamcela and W. Lim, "Artificial Intelligence in 5G Technology: A Survey," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 2018.

[6] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, and F. Kojima, "Big Data Analytics, Machine Learning, and Artificial Intelligence in Next-Generation Wireless Networks," *IEEE Access*, 2018.

[7] A. Boudi, M. Bagaa, P. Pöyhönen, T. Taleb, and H. Flinck, "AI-Based Resource Management in Beyond 5G Cloud Native Environment," *IEEE Network*, 2021.

[8] Z. Dong, J. Shi, W. Wang, and X. Gao, "Machine Learning Based Link Adaptation Method for MIMO System," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018.

[9] D. Wang, B. Song, D. Chen, and X. Du, "Intelligent Cognitive Radio in 5G: AI-Based Hierarchical Cognitive Cellular Networks," *IEEE Wireless Communications*, 2019.

[10] A. Balieiro, K. Dias, and P. Guarda, "A Machine Learning Approach for CQI Feedback Delay in 5G and Beyond 5G Networks," in *2021 30th Wireless and Optical Communications Conference (WOCC)*, 2021.

[11] K. Saija, S. Nethi, S. Chaudhuri, and R. M. Karthik, "A Machine Learning Approach for SNR Prediction in 5G Systems," in *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2019.

[12] D.-Y. Kim and S. Kim, "Network-aided intelligent traffic steering in 5G mobile networks," *Computers, Materials & Continua*, vol. 65, no. 1, pp. 243–261, 2020. [Online]. Available: http://www.techscience.com/cmc/v65n1/39564

[13] C. Gijón, M. Toril, S. Luna-Ramírez, and M. Luisa Marí-Altozano, "A Data-Driven Traffic Steering Algorithm for Optimizing User Experience in Multi-Tier LTE Networks," *IEEE Transactions on Vehicular Technology*, 2019.

[14] I. A. Alablani and M. A. Arafah, "An Adaptive Cell Selection Scheme for 5G Heterogeneous Ultra-Dense Networks," *IEEE Access*, 2021.

[15] N. Makris, C. Zarafetas, P. Basaras, T. Korakis, N. Nikaein, and L. Tassiulas, "Cloud-based convergence of heterogeneous RANs in 5G disaggregated architectures," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018.

[16] L. M. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5G mobile crosshaul networks," *IEEE Communications Surveys & Tutorials*, 2018.

[17] N. Makris, C. Zarafetas, S. Kechagias, T. Korakis, I. Seskar, and L. Tassiulas, "Enabling open access to LTE network components; the NITOS testbed paradigm," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2015.

[18] N. J. Gomes, P. Chanclou, P. Turnbull, A. Magee, and V. Jungnickel, "Fronthaul evolution: from cpri to ethernet," *Optical Fiber Technology*, 2015.

[19] "Tsaug - Time Series Augmentation," [Online], https://github.com/arundo/tsaug.

[20] I. Chatzistefanidis, N. Makris, V. Passas, and T. Korakis, "UE Statistics Time-Series (CQI) in LTE Networks," 2022. [Online]. Available: https://dx.doi.org/10.21227/ec7p-xq38