# Intelligent Mobile Handover Prediction for Zero Downtime Edge Application Mobility


Navdeep Uniyal*, Anderson Bravalheri*, Walter Featherstone†, Xenofon Vasilakos*,
Shangbin Wu†, Daniel Warren†, Reza Nejabati* and Dimitra Simeonidou*
*Smart Internet Lab, Department of Electrical & Electronic Engineering,
University of Bristol, Bristol, Clifton BS8 1UB, UK
Email: name.surname@bristol.ac.uk
† Samsung R&D Institute UK, Communications House, South St, Staines TW18 4QE, UK
{walter1.f, shangbin.wu, dan.warren}@samsung.com



***Abstract*—Ultra-reliable low-latency communication services are intrinsically challenging to deliver, with many 5G and future services, including mobile game streaming, adding further complexity by demanding *zero service downtime* in high-mobility scenarios. Solving these challenges is essential and *must* be addressed beyond mobile gaming to realise a multitude of current and future services like Virtual Reality or holoportation in mobile scenarios. Multi-access Edge Computing brings services "closer" to user consumption with evident advantages yet at the cost of maintaining a *zero downtime* guarantee when user handovers (HOs) are prevalent due to the decentralisation of services towards the network edge. In this work, we design and evaluate *intelligent HO prediction* models between radio 5G Base Stations. The motivation for *timely* user HO prediction lies in being a vital presupposition for path steering and other MANO control actions in contemporary programmable 5G networks to deliver a zero downtime perception during HO events. Our meticulous simulation and actual testbed evaluation results show that effective HO prediction can be achieved using a combination of Long Short-Term Memory (LSTM) or gradient boost regression with classification models, with the latter filtering out any Reference Signal Received Power (RSRP) prediction input outliers for predicting the serving cell.**



***Index Terms*—handover, mobility prediction, 5G network, multi-access edge computing, machine learning**


## I. Introduction

Ultra-reliable Low-Latency Communication (URLLC) 5G services pose significant challenges to the existing wireless networks in terms of strict quality of service (QoS) requirements. Amongst these requirements, *zero downtime* poses as one of the most severe and essential challenges that *must* be addressed in 5G and beyond to enable seamless functioning of URLLC services. At the same time, service delay susceptibility per se can be as extreme as in the order of a few dozen milliseconds(ms), such as for mobile gaming or eX-tended/Virtual Reality (XR/VR). This leads to considering multi-access edge computing (MEC) service deployments [24], as the "closer" a service is hosted to users, the lower the latency [18] as well as the congestion likelihood [23] with fewer hops over the data path between users and servers. Not only that, MEC deployments favour machine learning (ML) models that are envisioned to assist most modern URLLC services. This is due to their intrinsic combined offering of extremely low latency and data locality, hence overcoming any performance or privacy/security considerations or restrictions.

Nevertheless, the benefits above come at a dual cost: as identified in the zero downtime edge application mobility research project[1,2] over a mobile gaming use case, decentralising services to MEC deployments and, particularly, to hosts collocated with the Radio Access Network (RAN) sites, demands (i) to maintain a *zero downtime* guarantee over mobile handovers (HOs), while (ii) keeping users' state always in sync, i.e. within near-realtime delay margins. And all that irrespective of the users' mobility intensity and frequency.

As a result, the motivation for *timely* user *HO prediction* is vital, for it is the cornerstone enabler of all necessary processes prior to HOs to guarantee zero downtime. These *proactive* actions include (i) Software-Defined Networking (SDN) path steering and other control actions (e.g., increasing link capacity over a 5G slice); and (ii) Network Function Virtualisation (NFV) Management and Orchestration (MANO) such as increasing/allocating edge computational resources (e.g., GPU vCores) or fine-tuning user state synchronisation between service points of presence (PoPs).

In our prior work of [24], we presented a proof-of-concept mobile game streaming evaluation over an actual testbed setup that involved a kubernetes-based Edge service and highlighted the benefits of *timely* synchronising Edge service containers rather than migrating or using checkpoint and restore (CRIU)[3]. Specifically, we showed a two-order magnitude reduction of service downtime induced by a HO, from 5487 ms to an *incredible 25 ms*. Nonetheless, the former is an easy –even insignificant– conclusion without timely HO predictions. To cover this gap, the current work contributes a meticulous study for the design and evaluation of *intelligent HO prediction* models between radio 5G base stations (BSs) as a first step and as a cornerstone of advanced orchestration decisions for achieving a zero service downtime perception. In brief:

- **Regression & classification techniques for mobility prediction in 5G:** We present regression and classification techniques for serving cell prediction in a 5G radio network. We design two ML models, namely (i) *eXtreme Gradient Boosting (XGBoost)* [11] and (ii) *Long Short-Term Memory (LSTM) based Neural Network*, and compare them against a linear regression benchmark for predicting cellular RSRP future values after 2 sec, 8 sec and 16 sec periods. XGBoost is

[1] www.bristol.ac.uk/engineering/research/smart/projects/zerodeam/

[2] research.samsung.com/news/SRUK-and-University-of-Bristol-s-Smart-Internet-Lab-commence-5G-MEC-research-collaboration

[3] https://criu.org/Docker

proven to be a very efficient mobility prediction ML model with an accuracy of 90% [15]. However, the performance of LSTM is also proven in various automotive use-cases [14]. This work investigates the two models to predict user mobility in a 5G network to support URLLC use-cases.

Also, we present a classification algorithm based on XG-Boost, Extra Trees and Random Forest to predict the serving BS based on the output of the RSRP prediction. Our results show the error in regression is minimized by the classification to predict the serving cell. Note that the aforementioned periods in the future span both the ultra-delay critical synchronisation requirements and the migration time needs between edge-served Virtual Network Function (VNF) instances to support zero service downtime perception for mobile handovers [24].

• **Simulation-assisted Transfer Learning:** We use a purpose-built simulation environment designed to be compatible with our outdoor testbed [9]. This enables us to apply the *transfer learning* method [20] for developing-training-testing our models within a faster life-cycle and through various scale scenarios before moving, deploying and studying them to our full-scale actual 5G testbed at Bristol. Alongside our prediction models per se, transfer learning poses the cornerstone for further researching SDN and NFV MANO actions before HO events to enhance dynamic resource allocation and VNF placement and/or chaining.

• **3GPP RSRP model-based evaluation:** In radio networks, RSRP plays a major role in deciding which cell would be serving a connected user. The RSRP based HO decision is based on a complex relationship described in section III. We use 3GPP RSRP model-based values as training and validation datasets and conduct a meticulous evaluation. Our evaluation results show that effective HO prediction can be achieved using a combination of either (i) LSTM or gradient boost regression, and (ii) classification models, with the latter filtering out any RSRP prediction outliers for predicting the serving BS. While we have used the LSTM and gradient boost regression techniques to predict the RSRP values of all the potential serving cells, the classification techniques utilize the predicted RSRP values to futher predict the radio cell which would serve the user. The corresponding serving cell probabilities can be fed to MANO to deliver a zero downtime user perception during HO events.

For the rest of this paper, Sec. II discusses related work on mobility prediction and intelligent approaches, followed by the details of our simulation environment in Sec. III, and our ML models in Sec. IV. Finally, we present our performance evaluation in Sec. V before concluding with Sec. VI.

## II. Related work

Mobility prediction is not a new concept. It has been used previously to improve QoS support during HOs in cellular networks since the early 2000s, e.g. in [13], [22], and even before. Starting from LTE networks, multiple studies have been performed to establish mechanisms for efficient HO of users in a cellular network. More recently, with the advent of 5G networks, efforts have been increased to either establish a better cellular HO mechanism or to predict the future HO to support various QoS sensitive applications [3]–[5], [12], [19]. Some of the existing works rely on the use of ML mechanisms, while a few attempts have been made to address the issue without using ML methods and by utilising heuristics [3] instead. Most of the work has been carried out while targeting a specific scenario, or in some cases, a pattern such as a user movement is known along with the cell locations to establish an efficient HO mechanism.

In this work, we target MEC-enabled URLLC use cases covering a wide range of related use case scenarios spanning from high frame-per-second (fps) multiplayer game streaming to remote desktop; augmented reality; industrial internet-of-things (IIoT); autonomous driving; and XR/VR or holographic services. Moreover, unlike other efforts, we do *not* study tracking or predicting users' exact mobility path, but stress instead on the critical target of *timely* and *accurate HO prediction* between serving BSs and, hence, the service HO between the corresponding MEC service PoPs paired to BSs. To support mobility in such a scenario, game players expect to have a smooth HO from one BS to another without any perceived downtime [24]. Keeping user privacy under consideration, we assume, there is no user location data available to the monitoring system while it only relies on the radio signal information like RSRP. Additionally, we assume unpredictable user walking mobility patterns in our 5G testbed while interacting in small timescales with the streaming video gaming service.

As with prior works [21], [25], the more recent work of [3] relies on the predictable user pattern as the path is fixed (e.g., commute between home and office). The BSs and distances are known to the HO prediction using historical data, making it unsuitable for our case. Similarly, [5] and [4] uses the GPS data to predict the direction of user movement, which is not in line with the assumptions we made in this work.

Regarding works employing ML modelling solutions, the work of [14] investigates user mobility prediction in automotive scenarios with the use of LSTM recurrent neural network configurations. The authors show that LSTM can provide accurate mobility predictions leading to a balanced use of distributed resources through service scale migration or replication actions. The authors of [12] use channel state information along with the user HO history to employ supervised ML to predict the future HO. One drawback to this technique is the reliability of user equipment to report back the channel gain to the base station periodically, and hence the prediction could be delayed or corrupted. Another neural network-based work [19], states the effect of HO on various parameters ranging from resource utilisation to the user experience. It also uses the user location as an input to the problem and hence differs from the current work's assumptions.

## III. Data generation and simulation

The developed simulation environment consists of an aggregation of mathematical models representing the overall system behaviour and uses fixed-increment time progression. The primary purpose of this simulator is, given a specific BS arrangement covering a $L \times L$ square area, to generate a sequence of RSRP values, a description of which physical cell identifier (PCI) the user's equipment (UE) is connected to at each step, and a sequence of time values related to HO intervals.

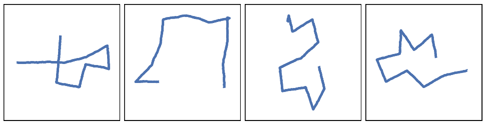

Fig. 1: Example of generated trajectories inside a $1.13 \times 1.13\ km^2$ area.

The *trajectory generation* assumes an urban scenario where users move alongside roads or paths forming lines and perform angular turns. This behaviour, illustrated in Fig. 1, is obtained by splitting the trajectory in a set of N segments with a variable number of steps. Each segment is generated by randomly selecting a target point inside the $L \times L$ square according to a uniform probability. The UE then moves in the direction of the target for a certain number of steps (given by the segment's length); however, noise with normal distribution is added to the displacement direction to account for non-rectilinear motion. The duration of each step is determined by the configured velocity of the displacement (which is assumed constant). The *RSRP computation* uses the generated trajectory to calculate the distance ($d_{\mathrm{3D}}$) between the macrocell and the UE at each point, which in turn is used to obtain the path loss ($PL$) for a given carrier frequency ($f_c$) based on the *urban macro* model defined [1] in Eq. (1). For simplicity, the UE is positioned 1.5 m off the ground. The path loss values are then used to derive RSRP according to Eq. (2), where $P_{\mathrm{TX}}$ represents the transmitted power of the BS, $G$ represents the antenna gain, and $\mathcal{N}(0, \sigma^2_{\mathrm{shadow}})$ is a random variable accounting for shadow fading, chosen according to a normal distribution with 0 mean and $\sigma^2_{\mathrm{shadow}}$ variance. Eq. (3) considers $\phi_{\mathrm{3dB}} = 65°$, a maximum attenuation value $A_{\max} = 30$ dB and the angle between user terminal and the cell orientation ($\phi''$). Since the computed values are very noisy, two stage filtering is applied: the first stage uses a rolling average for the last 40 samples of RSRPs and the second stage uses the layer 3 filtering technique [2] with filtering coefficient 4.

$$PL = 13.54 + 39.08 \log_{10}(d_{\mathrm{3D}}) + 20 \log_{10}(f_c) \tag{1}$$

$$RSRP = P_{\mathrm{TX}} + G - PL - \mathcal{N}(0, \sigma^2_{\mathrm{shadow}}) \tag{2}$$

$$G = \min\{12(\phi''/\phi_{\mathrm{3dB}})^2, A_{\max}\} \tag{3}$$

Finally, the *HO decision* and choice of S-PCI at each step follows the timer-based A3 algorithm described in 3GPP document [2]. According to A3, the UE enters a pre-condition for HO when the value of the RSRP for a neighbour BS is larger than the value of the RSRP for the currently connected BS (except for pre-configured values of offset and hysteresis). If the UE satisfies this pre-condition for a sufficient amount of time, then the HO is triggered. In our implementation, the offset values are assumed 0 dB, hysteresis is configured to 3 dB, and the time to trigger is set to 1024 ms. The initial BS cell is chosen according to the highest RSRP value.

## IV. Handover Prediction Models

To support URLLC user mobility, we must predict future user HO between BSs. The objective is to know the probability of the user getting served by a macrocell via accurate prediction of the serving PCI (S-PCI). The most crucial factor in determining user connection to a particular BS is the signal strength determined by the RSRP values. As shown in Fig. 2, we have divided the problem into two parts: First, to predict the RSRP value of each cell (with a potential to serve the user) and then to use the predicted RSRP values to predict the S-PCI. The former is a time-series regression problem, while the latter is a classification problem. The output of the whole system is deemed to be a matrix expressing the probability of the user being served by a PCI. This method makes the system *dynamic* and flexible to add multiple BSs in the prediction. In our experiments, we have implemented two different ML models and one benchmark heuristic algorithm to compare the RSRP predictions while using a multi-classification method to get the S-PCI probabilities.

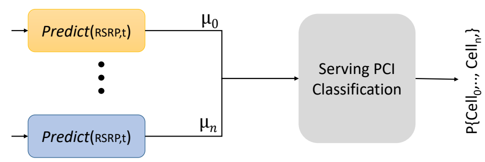


Fig. 2: Handover prediction methodology

### A. RSRP prediction

RSRP values follow a change pattern after user movement by *increasing with a decrease in distance between the user and the cell* as per Eq. (1) and (2). However, it is recommended to avoid direct access to user's location parameters due to privacy concerns, implying an additional challenge. As an alternative, we have implemented instead the following prediction techniques based on short-term history of RSRP values and S-PCI:

- **Ensemble and eXtreme Gradient Boosting:** Accurate predictions are achieved by training a single robust ML model over multiple iterations. An ML algorithm's objective can also be achieved if multiple ML models work together to find an optimum. Such a technique to sequentially build a robust predictive model is called ensemble learning [17]. Boosting is an ensemble method where multiple weak models (or learners) are combined to create a single robust model to improve the overall prediction accuracy. Gradient boosting aims to minimize the loss function by adding various weak learners using gradient descent over multiple iterations to find the direction where the loss decreases fastest. As shown in eq. 4, each weak learner is fit upon the derivative of the loss in the previous iteration with $\eta$ being the learning rate. If $\eta$ is large, the minima will be overrun easily, and a smaller $\eta$ would lead to longer training time.

$$F_{\mathrm{n}} = F_{(\mathrm{n}-1)} + \eta * \left(-\frac{\partial L}{F_{(\mathrm{n}-1)}}\right) \tag{4}$$

XGBoost is a highly efficient and one of the most recent tree boosting algorithms. As shown in eq. 4, it uses gradient descent to minimize the objective function with combined loss and regularization [11]:

$$\mathcal{L}^t = \sum_{(i=1)}^{n} l\left(y_i, f\left(x + \Delta x\right)\right) + \Omega\left(f_t\right) \tag{5}$$

XGBoost is a function of functions with $l$ in eq. 5 is a cart of decision trees. It uses a greedy algorithm to minimize the loss function $\mathcal{L}$ over time t and the regularization parameters $\Omega$ prevents overfitting of data. Additionally, the parallel processing of loss minimization makes it faster than most gradient boosting algorithms.

• **Long Short-Term Memory**: LSTM is a special kind of recurring Artificial Neural Network (ANN) that uses memory to store previous computations and an advanced four-stage design that learns about the contributions of these stored values to the final output.

This ability makes LSTM particularly suitable for processing time series and forecasts and is achieved via a "gate" mechanism. The purpose of each gate is to calculate a score between 0 and 1 based on the current and previous inputs. These scores are then used to modulate the values being either added to or retrieved from the cell memory. The four stages in LSTM are: *forget gate*, *input gate*, *candidate memory*, and *output gate*.

A new value is generated based in a combination of current and previous input by the *candidate memory*. This candidate is meant to be added to the previous value store in memory, however its final contribution is regulated by the *input gate*. Similarly the contribution of the previous stored value is modulated bye the *input gate*. Finally, the output gate determines how much of the value stored in memory will be used as output of the LSTM cell.

Since each gate is dynamically activated by the current and past inputs, LSTM can learn how to discard less relevant information and focus on the important values. This is done by optimising the weights and biases used in the linear combinations of current and past values of input used to calculate the gate scores and the candidate.

• **Linear Regression**: A simple linear regression can also be used to estimate future RSRP values based on the assumption that the prediction interval is small enough so that the users do not change the overall behaviour of their movement. This technique consists of fitting the last $N$ acquired RSRP values as a linear function of the sampling time – $RSRP(t) = at+b$ – and extrapolate the results using the obtained angular coefficient ($a$) and the intercept value ($b$). Predictions using extrapolation via linear regression are easily computed and do not require training. Nevertheless, they are constrained and generally only applicable for a short prediction time interval.

### B. Serving cell prediction

As shown in Fig. 2, the output of regression is given to the classification module to predict the S-PCI accurately. Our model uses a multi-classifier voting technique including XGBoost, random forest and extra trees ML models. The classifier's objective is to predict the current serving BS accurately while minimizing the previous step's prediction error (i.e. RSRP prediction). The training data is provided to all the algorithms, and a voting classifier [10] with "soft-voting" is used to determine the best classification model. The probabilities denote the likelihood of a user being served by a particular BS across each cell, and soft voting predicts the class label based on the argmax of the sums of the predicted probabilities. For $N$ base classifiers $b_i$, we assume $f_i^j(x_n)$ is the output of $i^{th}$ classifier for input $x_n$ of class $j^{th}$. Soft voting uses a weighted approach with a better classifier is given greater weight.

$$F(x_n) = \omega^{\mathcal{T}} f^j(x_n) \tag{6}$$

$$\text{where,} \quad f^j(x_n) = \left(f_1^j(x_n), ...., f_N^j(x_n)\right) \tag{7}$$

$\mathcal{T}$ is the transpose of a matrix and $\omega$ represents the weight matrix. Voting is based on the maximum weight among the competing classifiers. The selected classifier is used to predict the probability of a user being connected to a specific S-PCI.

## V. Performance Evaluation

To evaluate the objectives listed in section I, we performed a three-step evaluation by: (i) simulating mobility predictions using two macro-cells matching our actual testbed experimental setup; (ii) transferring and validating the trained models to our testbed, hence performing *"simulation-assisted transfer learning"*; (ii) conducting a *scalability study* by increasing the number of cells from two to four. Finally, we show the results of the classification to predict the S-PCI.

### A. Simulation & Experimentation setups

The simulation tool discussed in Sec. III was used to generate data corresponding to 300 random trajectories over a varying time duration between 2000 and 3500 seconds. From each trajectory a dataset was derived including the RSRP values of all BSs and the S-PCI (corresponding to the connected serving cell). We use 200 datasets for training and 100 for validation. Two scenarios were considered: *two macrocells* and *four macrocells*. Experiments where carried out at *Bristol's millennium square*, which is an open public space *testbed* covered by two 5G macro-cells, one facing 167°east configured with a transmitting power of 28 dBm and the other facing -86°west configured with 35 dBm. Volunteers were asked to carry 5G-enabled devices running a monitoring app and perform random trajectories lasting between 15 and 30 minutes each. Fig. 3 shows the testbed's BS arrangement and some of the recorded trajectories. For each trajectory, a dataset was created based on the logs generated by the monitoring app. Additionally, 10 extra datasets were created for adjusting the models during the *transfer-learning* process, by inputing the GPS coordinates of experimental trajectories into the simulation environment.

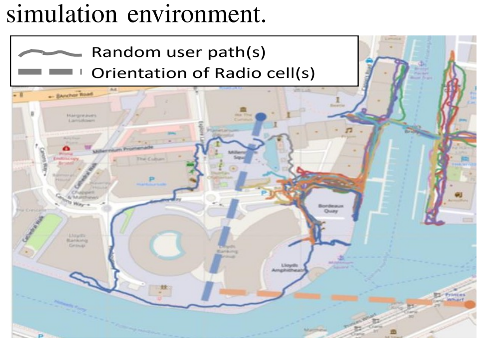


Fig. 3: 5GUK Testbed site at Bristol Millennium Square. Notice the annotated random user paths and the two differently coloured dashed lines denoting the orientation of each of the two cellular radio heads used during our RSRP validation experiments.

### B. Performance analysis

We evaluate our ML models against a benchmark in *three* future period scenarios according to the conclusions of our past work in [24] to predict RSRP values and S-PCI: *(i) 2 seconds* for *near-future* (immediate) predictions targeting fine-tuning state synchronisation between service containers across the current and future service MEC site; *(ii) 8* and *(iii) 16 seconds* in the future, hence capturing the time duration distribution for migrating a service container (i.e., spawning a new instance, and check-pointing & restoring state) to another service MEC point. For RSRP predictions (Fig. 4), the training data is varying over time; hence results are presented as three series at reference times $t{+}2$, $t{+}8$ and $t{+}16$ corresponding to scenarios (i), (ii) and (iii) above. Regarding S-PCI classification (see in Fig. 5), we use training data in its original form, i.e. containing the RSRP values of each BS and one S-PCI to denote the serving cell. Input for training and prediction of RSRP consists of past 100 values, i.e. t-100 monitored RSRP data. Based on the input RSRP dataset, the goal is to predict the future values. To conduct user mobility prediction, we predict the S-PCI based on predicted RSRP values from ML models (see Sec. IV-A) passed to the classifier.

*1) RSRP predictions:* In order to achieve zero-perceived downtime with latency-sensitive applications during the HO, the network orchestrator *must* get accurate HO predictions within a stipulated time to take necessary actions (e.g., quick synchronisation between service replicas or even whole VNF migration) [24]. Fig. 4 presents both a simulation and an experimental-based evaluation of RSRP predictions. All results refer to mean values along with 95% confidence intervals. Testbed experimental results refer to 10x validation repeats, while simulation results to 100x repeats.

**Transfer Learning validation & superiority over Lin:** Graphs (a) and (b) present a comparison between experimental and simulation-based results on two cell environments, where the simulation setup is designed to match our actual testbed. The first conclusion drawn from (a) and (b) is that testbed experiments *validate* our simulation evaluation. When transferring the simulation-trained models to the testbed, performance differences are not significant and sometimes practically non-existent (see XGBoost for $t{+}16$). In any case, the performance differences and trends are consistent. This validates and enables us to exploit the advantages of *transfer learning* from our purpose-built simulation environment to realist deployment.

Graphs (a) and (b) complement each other; (b) presents the mean absolute error (MAE) between predicted and actual RSRP, while (a) shows the "coefficient of determination" ($R^2$) values expressing the *percentage variation* of model predictions against actual RSRP. It is defined as $R^2 = 1 - \frac{RSS}{TSS}$, where RSS is the sum of squares of residuals (i.e., model predictions) and TSS is the total sum of squares of the actual RSRP measurements minus their mean. $R^2$ values closer to 0 denote that a model fails to produce RSPR predictions matching the actual ones, while values closer to 1 denote good-quality predictions. Overall, both (a) and (b) denote that our ML models exceed Lin by far in terms of $R^2$ and MAE, and for all periods, with Lin MAE values spanning larger than the ML models from approx. 82% for $t+2$ to approx. 51% for $t+16$. Also, confidence intervals denote that the ML models have a more robust performance. LSTM achieves overall the best performance for all three future periods $t+2$ (need for immediate actions), $t+8$ (mid-term actions), $t+16$ (longer-term actions like state migration). In fact, LSTM and XGBoost achieve near-optimal predictions for $t+2$ in both simulation and experimental runs, according to (b). However, LSTM has generally more robustness regarding predictions further in time. Nevertheless, we stress that the experimental results for both ML models for $t+16$ perform nearly close, with their confidence intervals largely overlapping, unlike simulation multiple ruins that show a -33.7% decrease in MAE terms for LSTM compared to XGBoost.

**Scalability and Robustness evaluation:** Graph (c), on the other, evaluates the ability of all models to scale with the number of cells. Whereas the performance superiority of LSTM and XGBoost remains, we notice that increasing the number of cells reduces entropy, hence benefiting the prediction ability for all models (even including Lin) due to the richer RSRP input from four sources rather than just two.

Moreover, all graphs in the figure indicate that the performance of our models remains robust when increasing the future period prediction, as confidence intervals for $R^2$ slightly increase in (a) yet remain small regarding MAE in (b) and (c). Unlike that, Lin's MAE confidence intervals are generally much higher for all periods in (b).

*2) S-PCI prediction on predicted RSRP:* Regardless of the anticipated impact on S-PCI prediction, the above results of Sec. V-B1 pose significant merits of LSTM and XGBoost, which can be exploited for dynamic resource allocation and the placement or chaining of VNFs as analysed in a multitude of recent works on the field, such as [6]–[9], [16], [24]. Therefore, we focus next our S-PCI classification evaluation on XGBoost and LSTM.

To evaluate the classification results we use *F1 Score*, which is a denoted by the formula 8 based on true positives (TP), false positives (FP) and false negatives (FN) and varies between 0 and 1. It expresses the ability of a model to balance both between capturing an S-PCI and being accurate when it does capture it.

$$F1\ Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{8}$$

We used the same dataset as with RSRP prediction to train and validate the S-PCI classifier. The predicted RSRP values from the two ML models are given to the trained classification model to predict the S-PCI and compared against the actual values to get *F1-score*. Fig. 5 compares the classification results on metrics listed in Sec. IV-B. The overall performance of S-PCI prediction using predicted RSRP values by LSTM outperforms the ones by XGBoost. Across the three scenarios (t+2,8,16), the F1-Scores of 2 and 4 cells are very close and within the range of their confidence intervals. However, in the nearest future, t+2 S-PCI predictions are $\sim 3\%$ better than t+8, which decreases to $\sim 2\%$ further at t+16. The F1-Score shows the model's overall ability to predict the S-PCI correctly with classification on all LSTM predictions ranging between

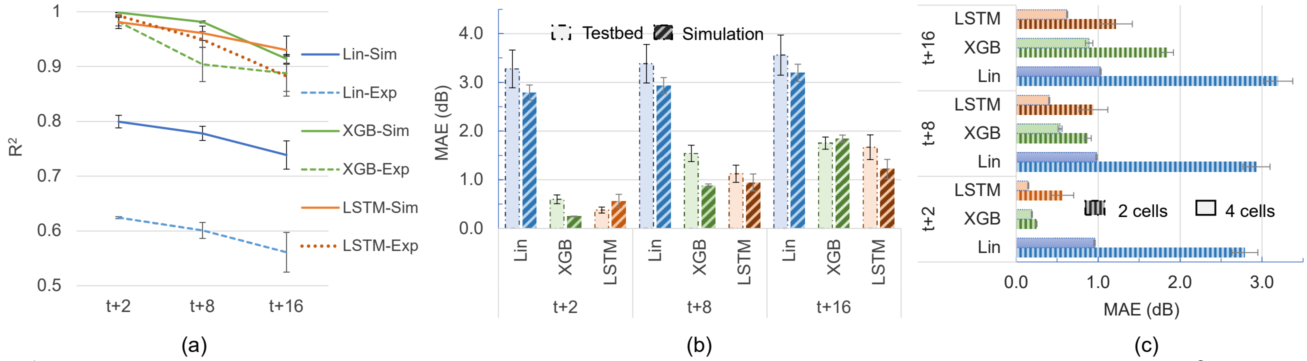


Fig. 4: RSRP prediction model performance comparison and Transfer Learning validation based on (a) Coefficient of Determination ($R^2$) and (b) Mean Absolute Error (MAE) results. Also, Graph (c) portrays a scalability and robustness evaluation by comparing two and four cell environment results.

$\sim$ 92 and $\sim$ 95% across the three scenarios. In general, the classification on LSTM predictions performs slightly better than XGBoost, indicating that the small errors observed during RSRP prediction are filtered out at this step.

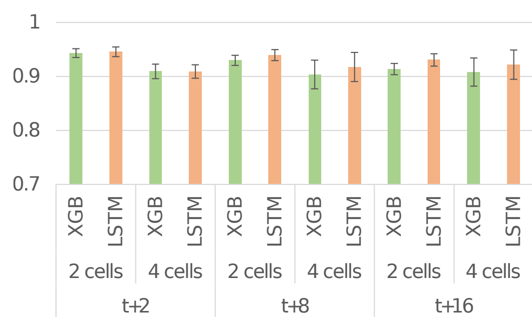


Fig. 5: F1-score on S-PCI classification on predicted RSRP values

## VI. Conclusion and Future Work

We present LSTM and XGBoost models for RSRP prediction and S-PCI classification, proving that they can achieve high and trustworthy accuracy levels regarding HOs. The exhibited performance levels for predicting the correct serving BS denote that such an intelligent HO prediction scheme can support URLLC applications to achieve zero downtime during mobile HO. We designed a purpose-built simulation environment matching the University of Bristol's 5GUK testbed facilities and prove it can be used for simulation-assisted transfer learning, posing the cornerstone for future work on SDN and MANO proactive actions for enhancing dynamic resource allocation, VNF placement or chaining. Additionally, we would like to work with the derived data and further combinations of regression and classification techniques to predict the user mobility and HO.

## Acknowledgement

This work was funded by Samsung Electronics (UK) Limited, as part of the "Zero Downtime Edge Application Mobility" research project ran by the University of Bristol's Smart Internet Lab in partnership with Samsung Electronics (UK) Limited.

## References

[1] 3GPP. TR 38.901: 5G; study on channel model for frequencies from 0.5 to 100 GHz. Technical report, 2017.

[2] 3GPP. Radio Resource Control (RRC) protocol specification (Release 15). Technical report, 2018.

[3] H. Abdah et al. Handover prediction integrated with service migration in 5G systems. In *2020 IEEE Int. Conf. on Commun. (ICC)*, pages 1–7, 2020.

[4] J. F. Abuhasnah and F. K. Muradov. Direction prediction assisted handover using the multilayer perception neural network to reduce the handover time delays in lte networks. *Procedia computer science*, 120:719–727, 2017.

[5] R. Ahmad et al. Efficient handover in lte-a by using mobility pattern history and user trajectory prediction. *Arabian Journal for Science and Engineering*, 43(6):2995–3009, 2018.

[6] S. Arora and A. Ksentini. Dynamic resource allocation and placement of cloud native network services. In *2021 IEEE Int. Conf. on Commun. (ICC)*, Montreal, 2021.

[7] Y. Bi et al. Distributed online resource allocation using congestion game for 5G virtual network services. In *2019 IEEE Global Commun. Conf., GLOBECOM 2019, Waikoloa, HI, USA, December 9-13, 2019*, pages 1–7, 2019.

[8] M. Bunyakitanon et al. Auto-3p: An autonomous VNF performance prediction & placement framework based on machine learning. *Comput. Netw., Elsevier*, 181:107433, 2020.

[9] M. Bunyakitanon et al. End-to-end performance-based autonomous VNF placement with adopted reinforcement learning. *IEEE Trans. Cogn. Commun. Netw.*, 6(2):534–547, 2020.

[10] J. Cao et al. Class-specific soft voting based multiple extreme learning machines ensemble. *Neurocomputing*, 149:275–284, 2015. Advances in neural networks Advances in Extreme Learning Machines.

[11] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016.

[12] X. Chen, F. Mériaux, and S. Valentin. Predicting a user's next cell with supervised learning based on channel states. In *2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 36–40, 2013.

[13] M. Chiu and M. A. Bassiouni. Predictive schemes for handoff prioritization in cellular networks based on mobile positioning. *IEEE J. Sel. Areas Commun.*, 18(3):510–522, 2000.

[14] U. Fattore, M. Liebsch, B. Brik, and A. Ksentini. Automec: Lstm-based user mobility prediction for service management in distributed MEC resources. In M. Aguilar-Igartua, P. Bellavista, A. A. F. Loureiro, and C. Giannelli, editors, *MSWiM '20: 23rd Int. ACM Conf. on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Alicante, Spain, November 16-20, 2020*, pages 155–159, 2020.

[15] H. Gebrie et al. What machine learning predictor performs best for mobility prediction in cellular networks? In *2019 IEEE Int. Conf. on Commun. Workshops (ICC Workshops)*, 2019.

[16] M. Golkarifard et al. Dynamic vnf placement, resource allocation and traffic routing in 5G. *Comput. Netw., Elsevier*, 188:107830, 2021.

[17] A. Natekin and A. Knoll. Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 7:21, 2013.

[18] N. Nikaein et al. LL-MEC: enabling low latency edge applications. In *7th IEEE Int. Conf. on Cloud Networking, CloudNet 2018, Tokyo, Japan, October 22-24, 2018*, pages 1–7, 2018.

[19] M. Ozturk et al. A novel deep learning driven, low-cost mobility prediction approach for 5G cellular networks: The case of the control/data separation architecture (cdsa). *Neurocomputing*, 358:479–489, 2019.

[20] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowl. and Data Eng.*, 22(10):1345–1359, 2010.

[21] V. A. Siris et al. Exploiting mobility prediction for mobility & popularity caching and DASH adaptation. In *17th IEEE Int. Symp. on A World of Wireless, Mobile and Multimedia Networks*, pages 1–8, 2016.

[22] W. Soh and H. S. Kim. Qos provisioning in cellular networks based on mobility prediction techniques. *IEEE Com. Mag.*, 41(1):86–92, 2003.

[23] X. Vasilakos et al. Cloud computing for global name-resolution in information-centric networks. In *2nd Symp. on Network Cloud Computing and Applications, NCCA 2012, London, United Kingdom, December 3-4, 2012*, pages 88–94, 2012.

[24] X. Vasilakos, W. Featherstone, N. Uniyal, A. Bravalheri, A. S. Muqaddas, N. Solhjoo, D. Warren, S. Moazzeni, R. Nejabati, and D. Simeonidou. Towards zero downtime edge application mobility for ultra-low latency 5G streaming. In *2020 IEEE Cloud Summit*, pages 25–32, 2020.

[25] F. Zhang et al. Edgebuffer: Caching and prefetching content at the edge in the mobilityfirst future internet architecture. In *16th IEEE Int. Symp. on A World of Wireless, Mobile and Multimedia Networks*, pages 1–9, 2015.