Hindawi
Mobile Information Systems
Volume 2021, Article ID 2775278, 19 pages
https://doi.org/10.1155/2021/2775278

Research Article

Using an Efficient Technique Based on Dynamic Learning Period for Improving Delay in AI-Based Handover

Saad Ijaz Majid,[1,2] **Syed Waqar Shah**,[1] **Safdar Nawaz Khan Marwat**,[3] **Abdul Hafeez**,[4] **Haider Ali**,[2] **and Naveed Jan**[5]

[1] *Department of Electrical Engineering, University of Engineering and Technology, Peshawar 25120, Pakistan*
[2] *Department of Electrical Engineering Technology, University of Technology, Nowshera 24100, Pakistan*
[3] *Department of Computer Systems Engineering, University of Engineering and Technology, Peshawar 25120, Pakistan*
[4] *Department of Computer Science and Information Technology, Jalozai Campus, University of Engineering and Technology Peshawar, Nowshera 24240, Pakistan*
[5] *Department of Electronic Engineering Technology, University of Technology, Nowshera 24100, Pakistan*

Correspondence should be addressed to Naveed Jan; naveed.jan@uotnowshera.edu.pk

Received 28 April 2021; Revised 27 July 2021; Accepted 5 August 2021; Published 25 August 2021

Academic Editor: Sungchang Lee

The future high-speed cellular networks require efficient and high-speed handover mechanisms. However, the traditional cellular handovers are based upon measurements of target cell radio strength which requires frequent measurement gaps. During these measurement windows, data transmission ceases each time, while target bearings are measured causing serious performance degradation. Therefore, prediction-based handover techniques are preferred in order to eliminate frequent measurement windows. Thus, this work proposes an ultrafast and efficient XGBoost-based predictive handover technique for next generation mobile communications. The ML algorithm in general prefers 70–30% of training and test data, respectively. However, always obtaining 70% of training samples in mobile communications is challenging because the channel remains correlated within coherence time only. Therefore, collecting training samples beyond coherence time limits performance and adds delay; thus, the proposed work trains the model within coherence time where this fixed data split of 70–30% makes the model exceed coherence time. Despite the fact that the proposed model gets starved of required training samples, still there is no loss in predication accuracy. The test results show a maximum delay improvement of up to 596 ms with enhanced performance efficiency of 68.70% depending upon the scenario. The proposed model reduces delay and improves efficiency by having an appropriate training period; thus, the intelligent technique activates faster with improved accuracy and eliminates delay in the algorithm to predict mmWaves' signal strength in contrast to actually measuring them.

1. Introduction

The next generation mobile communication promises low latency and delay. Traditionally, during a handover process [1], the user equipment (UE) measures radio parameters of a target cell called a measurement gap (MG). During the MG, data transmission stops and introduces an unwanted delay which contradicts the notion of uninterrupted high-speed data transfer in next generation mobile communications. MGs are critical time periods where transmission temporarily stops and even a call drop may occur. Despite at the cost of temporarily suspending data transmissions to execute MG, it still does not guarantee success of a handover request due to target cell radio bearings. Thus, repeated use of MGs causes performance degradation; therefore, this work proposes an ultrafast and efficient XGBoost-based algorithm which facilitates the network to enable MG only if the probability of handover success is higher than a given threshold. Thus, all MGs that are expected to return with unsuccessful handover requests are eliminated.

Previously, authors in [1] made contributions in this field; however, their work has limitations, such as their proposed model is novel in nature; however, the authors claim that their model follows channel coherence time (T_c) but there is no solution given for determination of T_c. If the samples are collected within T_c duration, then authors have not given any solution regarding how the proposed model handles the situation where the sample collection period exceeds T_c since as a recommended practice ML algorithm requires 70% of training data and 30% test data [2–5], which the authors also used for justification of results where sample collection time is set to be 0.7 × simulation time (T_s) for demonstrating results of $T_s = 40$ ms. It is further claimed that model sample collection time does not exceed T_c i.e., model is trained within T_c but it is found that, at lower T_s settings, such as below 200 ms, T_c becomes greater than T_s so how does samples' collection occurs if total T_s is less than T_c. The net results given also do not provide any reference to model training durations since T_c and $0.7 \times T_s$ are not equal time durations.

As discussed, in ML-driven algorithms, preferably 70% of data is used for training the model and remaining 30% is used as test data but this percentage might vary based upon applications. However, in mobile communications, gathering 70% of training data becomes challenging because the sample collection period should strictly follow T_c since beyond this period the channel does not remain correlated. So, if a model keeps on collecting samples beyond T_c in order to collect above required percentage of training samples, the proposed model becomes unable to predict accurate channel state information (CSI). The test results performed in this work suggest that 70% of samples can only be collected if the sample collection period remains within limits of T_c because beyond that the proposed model begins to collect unrealistic samples of the given channel since channel parameters change after T_c.

At first, this work provides solution how sample collection should be restricted to within T_c. The test results show that, for $T_s > 200$ ms, T_c becomes small enough that if 70% of samples are to be collected, then the model breaches coherence time limits. Therefore, to solve this problem, the proposed work exploits advanced error control of the XGBoost algorithm [6] where it is capable to compensate for missing samples without trading off its prediction capability or loss to HSR. Moreover, the proposed technique despite of being starved of required samples makes the model response time even far better.

1.1. Contributions.

The contributions of the proposed work are as follows:

(1) Determination of proper channel coherence time

(2) Accurate training period of the model

(3) Improved delay and the proposed algorithm activates much faster

(4) Less memory consumption because number of samples are reduced

(5) Utilization of error control in the XGBoost algorithm

(6) Eliminate oversampling

(7) Reduced learning period, which eliminates underutilization, and the model executes more handover decisions

The paper is further organized in such a manner that Section 1 introduces the problem and literature review, Section 2 details the proposed solution, Section 3 discusses results, and Section 4 provides conclusion of the proposed work.

1.2. Literature Review.

Estimating channel for handover applications is broadly classified into two major algorithms [1, 7, 8]. (1) Real-time monitoring and (2) mathematical modelling or ML-based estimations. Real-time monitoring constitutes of MGs where a UE scans the radio parameter, while the latter uses mathematical modelling or AI to estimate radio parameters. The former process is far better because it enables a UE to be smart enough in deciding whether to trigger a measurement gap or override it in contrast to physically scanning radio bearings. However, these prediction algorithms are required to be training efficient (requires lesser training data to be ultrafast) and accurate so that required inputs are processed quickly in time. The proposed work now argues some classical and, later, ML-based channel estimation techniques.

Typically, Angle of Arrival (AoA) and Angle of Departure (AoD) are used to estimate mmWave signal strength [9–14]. However, the frequency ranges in GHz make this technique susceptible to greater channel attenuation and estimation error. Thus, the technique makes design of such circuitry complex and expensive to develop. In contrast, the authors in [15–18] exploited scattering behaviour of mmWaves through correlation-based adaptive compressed sensing (CBACS) for precise estimation of AoA and AoD. This technique determines CSI through correlation between received CSI and quantized sensing vectors but it inherits resolution loss due to channel angle quantization [19].

Hybrid precoding with multiple numbers of antennas can also provide CSI; however, channel subspace sampling constraints affect its performance in digital baseband systems and also limit complete channel information [20]. Orthogonal matching pursuit (OMP) [21–25] exploits sparse properties of mmWaves which is a promising technique but it does not perform well due to its complicated nature if the number of UEs is high or in case of higher number of handovers.

Quantized AoA and AoD with an oracle estimator [22] and OMP generate a parametric model to determine CSI. However, it is not feasible for mmWave applications since the signal-to-noise ratio (SNR) is low before beam forming. A rogue technique [25–30] having dual connectivity of both 4G and 5G is also a possibility but it is not feasible because it causes immense battery drainage and might cause ping pong effects, while handing over to target cell.

The literature above highlights some of the classical techniques used to determine radio bearings of a target cell. However, all these algorithms are MG dependent, and there is no novelty where an unwanted MG can be avoided through prior prediction of mmWaves' signal strength by exploiting out-of-band information that is already available. With the advancement of computing power, ML-driven algorithms are mostly proposed to overcome limitations in classical techniques [31–39] in such scenarios. However, the ML-based technique is only effective if it is efficiently exploited. For this purpose, some commonly used ML schemes are discussed ahead.

RL (reinforcement learning) is a fluky methodology, and the algorithm is robust [40–43]; however, this architecture does not guarantee improved latencies and delays. Therefore, this technique is not recommended for applications where robust and quick decisions are required. A holistic model proposed in [44] and based on cost values for applications in handover might also be an option but such a model does not use readily available information from sub 6 GHz band. Moreover, the holistic model provides three states of information i.e., nonpredictive case, incorrect, and correct case making it rather complex for proposed applications because the proposed model works in two states only i.e., correct and incorrect prediction [1].

The kernel-based algorithm [35, 45, 46] in this research area is also debatable; however, it is only applicable where users are moving in a straight line. In mobile communications, this approach causes limitations for designing a decision model because user population density and UE movement are random in a cell. Moreover, the kernel-based algorithm does not consider geographical coordinates whereas the proposed model does consider geographical coordinates and also does not restrict UEs to move in a straight line. Dynamic Fuzzy Q-Learning is also a technique proposed in [47–49] for handover estimations where it establishes new learning thresholds and builds new fuzzy rules. However, this algorithm is not useful in case of the proposed algorithm because it is more suitable for adjusting time-to-trigger (TTT) mechanism. On contrary, the proposed algorithm predicts mmWave signal strength prior to measurement window; thus, it guarantees the success or failure of a handover. The novelty and promising factor of the proposed work are the model requires less time to train, it activates faster with improved accuracy (avoids over fitting), and the model is trained within T_c which significantly enhances effectiveness of the proposed model.

The proposed algorithm is designed to classify if the predicted estimates are valid or otherwise; therefore, it is essentially a classification challenge. For classification problems [50–53], XGBoost is a preferred method; it turns weak learners into stronger models, and its strength lies in multicore processing which tailors individual trees. These features can be effectively exploited to mimic distributed base stations of cellular networks. The algorithm also provides effective solution to find the predictor variable with less preprocessing [54] and offers lower search time. Various optimization algorithms are keys to its scalability [6, 55], and application of 2^{nd} order derivatives to the loss function makes accurate approximations. The algorithm generates advanced L1 and L2 regularization, and XGBoost offers quick training and parallel computing for robust processing. It also reduces its objective function to a convex differentiable loss function and regularization terms. Similarly, the potential and features of XGBoost are also recommended by the authors in [56–58].

1.3. Proposed ML Algorithm. As per industry standards [59], weak RSRP in comparison to set thresholds triggers event A2 leading to an MG for inter-RAT measurements; however, in the meanwhile, if UE detects better RSRP; then, it triggers event A1 to cease inter-RAT measurements, but in due course, if better mmWaves are detected, then event B2 triggers to initiate handover as given by Figure 1.

The proposed algorithm in contrast to the existing model adopts the ML-based approach where mmWaves' signal strength is computed through fractional out-of-band information [1]. The proposed work refers this ML-based scheme as fractional out-of-band information because for training the model, it will use target cell radio information, and once it gets trained, the ML algorithm starts working independently. Here, the measurement gap triggers only if the probability of success is above a set threshold. The algorithm computes mmWaves' signal strength, while remaining in sub 6 GHz band instead of physically scanning radio bearings. The proposed algorithm generates true versus false positive rates for receiver operating characteristics area under curve (ROC-AUC), thus enabling UE or base station to accept or reject its estimations. For this purpose, iterative training and cross validation are used. Table 1 shows the pseudocode, while Figure 2 demonstrates the proposed model.

2. Model Design

2.1. UE Distribution. The proposed model considers two collocated cells with circular boundaries having radius r operated at different frequencies and technologies. The UEs are smart enough to predict target cell mmWave signal strength, while remaining in sub 6 GHz band and without an MG. The mobile users are distributed in the network according to homogeneous Poisson Point Process (PPP) having intensity of ф showing expected users per unit area. The mobile coverage area is denoted by A, and the total number of users are N, and the UEs get sampled from Poisson distribution where mean equals to Њ$A =$ Њ $\times 3.14 \times r^2$, and r represents radius of cell. Therefore, the UEs are plotted such that $S(u)$ is stochastic with the process rate of λ having probability of Q in time u where B is output of an arbitrary function defined by $\lim_{j\longrightarrow 0}(B(j)/j) = 0$. Then, mathematically,

$$q_0 = (u + j) = Q\,[S(u + j) = 0], \tag{1a}$$

$$Q[S(u) - S(0) = 0,\ \mathrm{S(u + j) - S(u)} = 0], \tag{1b}$$

$$[\text{Independent Increments}] = Q\,[S(u) - S(0) = 0] \cdot Q\,[\mathrm{S(u + j)} - \mathrm{Q(u)} = 0], \tag{1c}$$

$$\begin{aligned} \text{Stationary} &= Q[S(u) = 0] \cdot P[S(j) = 0] \\ &= q_0(u)(1 - \lambda j + b(h)). \end{aligned} \tag{1d}$$

And so,

$$\frac{q_0(u + j) - q_0(u)}{j} = -q_0(u)\lambda + \frac{b(j)}{j}, \tag{1e}$$

$$\text{For } j \longrightarrow 0^{+}\ \frac{\mathrm{d}q_o(u)}{\mathrm{d}u} = -q_0(u)\lambda, \tag{1f}$$

$$\frac{\mathrm{d}q_0(u)}{q_0(u)} = -\lambda \mathrm{d}u, \tag{1g}$$

$$\text{For } S \geq 1, \tag{1h}$$

$$q_n(u + j) = Q[S(u + j) = n], \tag{1i}$$

$$\begin{aligned} &Q[S(u) = s,\ S(u + j) - S(u) = 0] + Q[S(u) = S - 1,\ S(u + j) \\ &\quad - S(u) = 1] + Q[S(u + j) = S,\ S(u + j) - S(u) \geq 2], \end{aligned} \tag{1j}$$

$$= Q[S(u) = s].\, QS(j) = 0 + QS(u) = S - 1.\, QS(j) = 1 + b(j). \tag{1k}$$

The equation simplifies to

$$= (1 - \lambda j)q_s(u) + \lambda j q_{s-1}(u) + b(j). \tag{1l}$$

For the derivation,

$$Q[S(u + j) = s,\ S(u + j) - \mathrm{S(u)} \geq 2] = \mathrm{b(j)}, \tag{1m}$$

because

$$\lim_{j \longrightarrow 0^{+}} \frac{QS(u + j) = n,\ S(u + j) - S(u) \geq 2}{j} \leq \frac{\lim_{j \longrightarrow 0^{+}} QS(u + j) - S(u) \geq 2}{j} = 0. \tag{1n}$$

Since

$$\frac{q_s(u + j) - q_s(u)}{j} = -\lambda q_s(u) + \lambda q_{s-1}(u) + \frac{b(j)}{j}, \tag{1o}$$

$$\begin{gathered} \text{For } j \longrightarrow 0^{+}, \\ \frac{\mathrm{d}q_s(u)}{\mathrm{d}(u)} = -\lambda q_s(u) + \lambda q_{s-1}(u), \end{gathered} \tag{1p}$$

$$\frac{\mathrm{d}\left(e^{-\lambda u} q_s(u)\right)}{\mathrm{d}(u)} = \lambda e^{\lambda u} q_{s-1}(u). \tag{1q}$$

This refers to the fact that any event in any interval during simulation time is Poisson distributed.

To demonstrate that any UE position is an independent and identically distributed sample from continuous uniform distribution, let X_1 denote exponential distribution having parameter λ; then for X_2, having independent increment, for number of events m,

$$Q[X_2 > u | X_1 = m] = QS(U + m) - S(e) = 0\,|S(m) - N(m^{-}) = 1, \tag{2a}$$

$$= Q[S(u + m) - \mathrm{S(m)} = 0], \tag{2b}$$

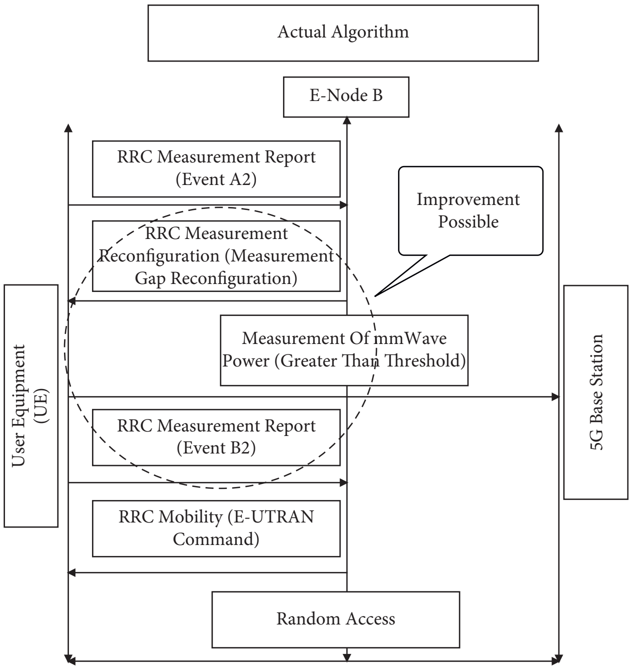

Actual Algorithm
E-Node B
RRC Measurement Report (Event A2)
Improvement Possible
RRC Measurement Reconfiguration (Measurement Gap Reconfiguration)
Measurement Of mmWave Power (Greater Than Threshold)
RRC Measurement Report (Event B2)
RRC Mobility (E-UTRAN Command)
Random Access
User Equipment (UE)
5G Base Station

Figure 1: Baseline model.

$$\begin{aligned} &= Q[S(u) = 0] \\ &= e^{-\lambda u}, \end{aligned} \tag{2c}$$

$$\text{where as, } Q[X_2 > u] = D[Q(X_2 > u \mid X_1)] = e^{-\lambda u}. \tag{2d}$$

Therefore,
X_2 distribution is exponential with parameter λ, such that

$$Q[X_2 > u] = Q[X_2 > u \mid X_1 = m], \text{ thus } X_2 \text{ and } X_1 \text{ are independent.} \tag{2e}$$

2.2. Radio Propagation and Channel Modelling. Radio propagation models play a significant role in performance of the proposed model since it is an estimation and prediction-based algorithm. The close-in free space path loss model is given by the following equation [60, 61]:

$$\mathrm{PL}[\mathrm{dB}] = \mathrm{FSPL}\ [\mathrm{dB}] + 10\,k\ \log_{10}(d) + \gamma[\mathrm{dB}] + \Delta_\sigma. \tag{3}$$

Here, PL = path loss function from free space ($FSPL$), k = exponent of path loss, d = distance, γ = constant for atmospheric attenuation, and Δ = Gaussian random variable (GV) having zero mean and standard deviation (SD) of σ.

This research work discusses following radio propagation models [60–65]:

(1) Alpha Beta Gamma (ABG)

(2) Close-in (CI) free-space reference distance model

(3) CI model with a frequency-weighted (CIF) path loss exponent

(4) Cost 231 Model

The ABG model is given by the following equation:

$$\mathrm{PL}^{\mathrm{ABG}}(f,d)[\mathrm{dB}] = 10\beta\ \log_{10}\left(\frac{d}{1\,\mathrm{m}}\right) + \alpha + 10\eta\ \log_{10}\left(\frac{f}{1\,\mathrm{GHz}}\right) + \gamma_\sigma^{\mathrm{ABG}}, \tag{4}$$

where $\mathrm{PL}^{\mathrm{ABG}}(f,d)[\mathrm{dB}]$ = path loss in dB, α and η = coefficients of dependence for path loss, β = optimized offset value, d = 3-dimensional receiver and transmitter distance, f = carrier frequency, $\gamma_\sigma^{\mathrm{ABG}}$ = zero mean GV, and σ = SD in large-scale signal fluctuation. In single frequency applications, the ABG model becomes floating intercept (AB or FI) which is already in the 3GPP standard.

The CI model is given by the following equation:

$$\mathrm{PL}^{\mathrm{CI}}(f,d)[\text{in dB}] = \mathrm{FSPL}(f,d_1)[\text{in dB}] + 10\,a\ \log_{10}\left(\frac{d}{d_1}\right) + \gamma_\sigma^{CI}, \quad \text{for } d \geq d_1, \tag{5}$$

where f = frequency (GHz), d_1 = close-in free space reference distance, a = path loss exponent, γ_σ^{CI} = zero mean GV with SD of σ in dB, d is the three-dimensional transmitter and receiver distance, and $\mathrm{FSPL}(f,d_1)$ is the free space path loss (FSPL) in dB with frequency f and distance between transmitter and receiver d_1.

The $\mathrm{FSPL}(f,\ d_1)$ is given by the following equation:

$$\mathrm{FSPL}(f,\ d_1)[\text{in dB}] = 20\ \log_{10}\left(\frac{4 \times 3.14 \times f \times d_1 \times 10^9}{3 \times 10^8}\right). \tag{6}$$

The CI model in 3GPP/ITU format is given below:

$$\mathrm{PL}^{\mathrm{CI}}(f,d)[\text{in dB}] = \mathrm{FSPL}(f,d_1)[\text{in dB}] + 10a\ \log_{10}(d_t) + \gamma_\sigma^{CI}, \quad d_t = \frac{d}{d_1}, \tag{7a}$$

$$\mathrm{PL}^{\mathrm{CI}}(f,d)[\text{in dB}] = 10\,a\ \log_{10}\left(\frac{d}{d_1}\right) + 20\ \log_{10}\left(\frac{4 \times 3.13 \times d_1 \times 10^9}{3 \times 10^8}\right) + 20\ \log_{10}(f) + \gamma_\sigma^{CI}. \tag{7b}$$

Table 1: Pseudocode for the proposed algorithm.

Step	Algorithm
1	Input all parameters
2	Derive total UEs in respective location and calculate RSRP for 4G and mmwave with given radio parameters
3	While measurement gap open and mmwave RSRP > threshold $\longrightarrow$ estimate HSR, controlling feature $[y]_i = \mathrm{UE}_i$
4	Optimise hyperparameters using grid search on K-fold cross-validation
5	Split data for training sequence, testing, and validation; model training = channel coherence time
6	Proposed model training (XGBoost)
7	Search proposed model handover decision (0 or 1)
8	Plot ROC-AUC
9	If
	ROC-AUC > decision threshold
	Then
	Override conventional algorithm
	Else
	Do not override conventional algorithm
10	End

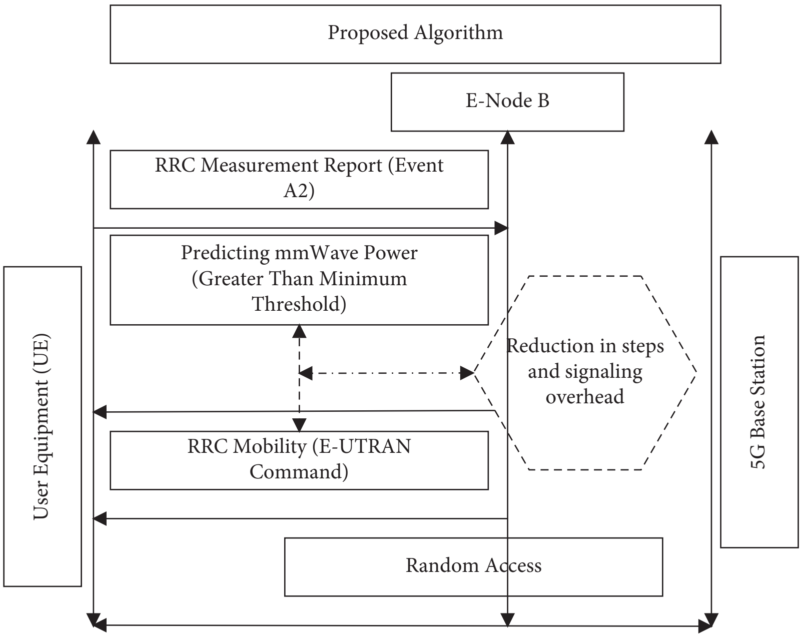

Proposed Algorithm
E-Node B
RRC Measurement Report (Event A2)
Predicting mmWave Power (Greater Than Minimum Threshold)
Reduction in steps and signaling overhead
RRC Mobility (E-UTRAN Command)
Random Access
User Equipment (UE)
5G Base Station

Figure 2: Proposed model.

For multifrequency modelling, the CIF model can be given in the simplified form by the following equation:

$$\mathrm{PL}^{\mathrm{CIF}}(f,d)[\text{in dB}] = \mathrm{FSPL}(f,\ 1\,\mathrm{m})[\text{in dB}] + 10\,a\left(1+\frac{j(f-f_1)}{f_1}\right)\log_{10}(d)+\gamma_{\sigma}^{\mathrm{CIF}},\quad \text{for } d \geq d_1. \tag{8}$$

The proposed algorithm is based upon estimates and predictions. Therefore, the proposed model remains stable, while choosing a model with low shadow factors and path loss. Thus, the preferred model for the proposed algorithm is the ABG model and is given by the following equation:

$$\mathrm{PL}[\text{in dB}] = \alpha + \beta \times 10\ \log_{10}(d) + x_{\sigma}. \tag{9}$$

The 4G path loss model is given by the Cost 231 model:

$$\mathrm{PL}[\text{in dB}] = \left(46.3 + 33.9 \times \log_{10}\left(f_c\right) + 13.82 \times \log_{10} h_{te} - \alpha\left(h_{re}\right) + \left(44.9 - 6.55 \times \log_{10}\left(h_{te}\right)\right) \times \log_{10}(d) + C\right), \tag{10a}$$

where f_c = carrier frequency, $\alpha(h_{re})$ = frequency correction factor, h_{te} = antenna height, d = distance, and $C = 3$ dB.

The correction factor $\alpha(h_{re})$ is given by the following equation:

$$\alpha\left(h_{re}\right) = 3.2 \times \log_{10}\left(11.75 \times h_{re}\right)^2 - 4.97, \tag{10b}$$

where h_{re} is the receiver height.

The radio parameters used in simulation are given in Table 2.

2.3. Proposed Model Training Period. Coherence bandwidth and delay spread parameters define time dispersive nature of the channel but these parameters do not describe any channel changes due to relative motion of the user or a base station or due to movement of different entities in the channel. For this purpose, Doppler spread and coherence time [66, 67] are used to specify time-varying nature of the channel. Modern communication systems rely on AI for enhanced use in contrast to traditional communication systems which rely on fixed conjectures and hence limiting its use. The notion that the channel remains invariant is only correct if relative motion is lower compared to $\lambda/2$, where λ refers to the ratio between speed of light and carrier frequency [68]. However, modern communication systems do not have mobility limitations; thus, Doppler spread and coherence time become important parameters for estimation while switching of to better frequency spectrums. Technically, T_c is time dual of Doppler spread or simply these are inverse of each other [67] and used for quantifying frequency dispersiveness relative to time-varying nature of the channel in the time domain. T_c is an important parameter for determining handoffs [69] but the proposed work also considers T_c in terms of model's training time control and computes it through mathematical equations. Moreover, up to 100 ms, the channel varies slowly [70], and the proposed model test results also support this fact by finding out that upto a range of approximately 100 ms, the channel coherence time does not significantly affect simulated results; however, beyond 100 ms, it introduces major impact and is discussed in Section 3.

The following equations provide T_c [67, 69, 71]:

$$T_c \cong \frac{1}{f_m}, \tag{11a}$$

$$f_m\,(\text{maximum Doppler shift}) = \frac{v}{\lambda}. \tag{11b}$$

T_c gives time duration where two received signals have strong potential for amplitude correlation. As a general rule, the threshold level is defined as $1/(\sqrt{2})$; thus, the coherence time becomes as given by the following equation:

$$T_c = \frac{9}{16\pi f_m}. \tag{12}$$

In practice, the time duration by equation (11a) suggests a period where signal amplitude may fluctuate more but equation (12) suggests a too restrictive period. Therefore, practically, geometric mean of these two equations is used which is given by the following equation:

$$T_c = \sqrt{\frac{9}{16\pi f_m^2}}. \tag{13}$$

Although, there are multiple ways to determine a more realistic value of T_c but those techniques require live monitoring of their respective parameters and special enhancements of radio circuitry, while in contrast, the proposed work requires prediction-based value which provides enough time window for the proposed model to get trained where signal amplitude fluctuations are wilder to effectively simulate mmWaves' behaviour. Secondly, the proposed model requires simple estimation of T_c because it only requires an estimated time limitation (without involving massive computing resources of UE or BS) upto which the model should be trained effectively without introducing significant delays and loss in efficiency of the model, while keeping the radio circuitry simple. Therefore, the proposed model T_c is given by equation (11a) due to reasons mentioned above.

According to [2–5], the ML algorithm should get trained for the period of $T_{Train} = 0.7 \times T_{sim}$; however, the proposed work suggests that T_{Train} should never breach T_c. This makes the estimator even more efficient, faster, and accurate in making predictions and eliminates overfitting. Therefore, T_{Train} mathematically is given by the following equations:

Table 2: Radio parameters for simulation.

No.	Parameter	Value
1	Radio propagation models	4G cost 231 model and 5G floating intercept model
2	Floating intercept model values (α, β, σ)	116.77, 0.41, 5.96
3	RRC events' time to trigger (TTT)	0 s
4	Event A1 (RRC) trigger	125 dBm (negative)
5	Event A2 (RRC) trigger	130 dBm (negative)
6	Event B2 (RRC) trigger	95 dBm (negative)
7	LTE center frequency	2100 MHz
8	LTE bandwidth	20 MHz
9	Cyclic prefix (LTE)	Normal
10	5G mmWave center frequency	38 GHz
11	5G mmWave bandwidth	100 MHz
12	UE physical status	Nonstatic
13	Simulation time (ms)	900 to 100 with step size of 100 ms
14	Cell coverage radius (r)	350 m
15	Base station power (5G)	46 dBm
16	Base station power (LTE)	46 dBm
17	Radio propagation (antenna)	Omnidirectional
18	Height (antenna)	36 m
19	Sub 6 GHz (antenna gain)	17 dBi
20	mm-wave (antenna gain)	24 dBi
21	Cellular coverage pattern	Circular
22	User height	1.5 m
23	Poisson point parameter (μ)	2×10^{-3}

$$T_{\text{Train}} = \text{T}_1 = T_{\text{Coherence}}, \quad \text{For } 0.7 \times \text{T}_{\text{sim}} > T_{\text{Coherence}}, \tag{14}$$

$$T_{Train} = \text{T}_2 = 0.7\, x\, T_{Sim}, \ \text{For } 0.7 \times \text{T}_{\text{sim}} \leq T_{\text{Coherence}}. \tag{15}$$

2.4. XGBoost Algorithm. Let $T(x)$ denote the gathered parameters, as given in Table 2, such that the parameter set is given by equation (16); $T(x)_n$ denotes individual parameters:

$$T(x) = [T(x)_1, T(x)_2, \ldots, T(x)_n]. \tag{16}$$

For output y, let F_0 denote the first predictive model. If the predictive models are m in number, then

$$y_m = f(F_m). \tag{17}$$

$F_0 \longrightarrow (y - F_0)$ associates with residuals.

A fresh model h_1 is produced to fit the former step. Thus, h_1 together with F_0 generates F_1 as a boosted F_0, given by equations (18a)–(18c), and F_m value can be determined for any value of m in this fashion.

Therefore,

$$F_1 = F_0 + h_1, \tag{18a}$$

$$F_2 = F_1 + h_2, \tag{18b}$$

$$F_m = F_{m-1} + h_m. \tag{18c}$$

The preceding functions are not affected by additive learners. However, the newest model bears lower error in comparison to predecessor. Thus, m^{th} step error is

$$M_1 < M_0, \tag{19a}$$

$$M_2 < M_1, \tag{19b}$$

$$M_m < M_{m-1}. \tag{19c}$$

For the given algorithm, the decisive aspect for F_m is h_m. Thus, the value of h_m determines F_m.

Therefore,

$$F_m = F_{m-1} + \varepsilon h_m, \tag{20}$$

where ε corresponds to the scaling factor for h_m.

Let the proposed model build a total of N trees; then,

$$\text{For Proposed Model} = \sum_{m=0}^{M} f_m. \tag{21}$$

The prediction at data point j in l^{th} step for feature vector x_j is given by

$$y_j(l) = \sum_{m=1}^{l} f_m(x_j). \tag{22}$$

The following equation provides the objective function for the proposed algorithm:

$$\text{Obj}(\theta) = \text{Я} + \text{¥}, \tag{23}$$

where Я represents linear and logistic regression, and $\text{¥} = \alpha B_1 + 0.5\lambda B_2^2 + \gamma J$, while vector $B =$ leaf weights and $J =$ number of leaves [6]:

$$
\begin{aligned}
\text{Logistic} &= -\frac{1}{M}\sum_{h=1}^{M}\left(k_h \ \log\left(g_h\right) + \left(1-k_h\right)\log\left(1-g_h\right)\right),\\
k_h &= h^{\text{th}} y \text{ value for given input},\\
g_h &= h^{\text{th}} \text{ probabilistic value},
\end{aligned}
\tag{24}
$$

$$
\text{Linear regression} = \frac{1}{M}\sum_{h=1}^{M}\left(k_h - k_{\hat{h}}\right)^2, k_h = h^{\text{th}} y \text{ value for given input}, \ k_{\hat{h}} = h^{\text{th}} \text{ predicted value}. \tag{25}
$$

Hyperparameter values are given in Table 3. For tuning hyperparameters, the grid search method is used on k-fold cross-validation, as grid search offers the optimized essential values.

Table 3: Hyperparameters.

No.	Parameters	Value
1	Maximum depth	6, 8
2	Child weights	0, 10
3	Training data	0.7
4	Weights of samples	0.5, 0.7
5	Objective	Logistic and linear
6	Cross validation	5
7	Complexity control	0, 0.002
8	Weights of samples	0.5, 0.7
9	Lambdas	0, 1, 2
10	Estimator number	500

3. Results and Discussion

This section shows that it is not necessary to collect 70% of required training samples and achieve similar HSR with reduced number of samples by using the XGBoost algorithm, hence improving the model response time.

The following equations are used for calculations:

$$
\text{No. of failures} = \text{total attempts} - \text{ successful attempts}, \tag{26}
$$

$$
\text{HSR }(\%) = \left[\frac{\text{number of succesful attempts}}{\text{total number of attempts}} \times 100\%\right], \tag{27}
$$

$$
\text{Introduced delay} = [\text{incorrect training period} - \text{correct training period}], \tag{28}
$$

$$
\text{Number of attempts missed } = \text{Њ} = \text{ attempts with } T_{\text{Correct}} - \text{attempts with } T_{\text{Incorrect}}, \tag{29}
$$

$$
\%\,\text{Њ} = \left[\frac{\text{Њ}}{\text{attempts with } T_{\text{Correct}}} \times 100\right]. \tag{30}
$$

To effectively demonstrate effects of the ML training period on efficiency and response time of the model, two scenarios are arbitrarily instantiated where T_{Train} should be equal to T_c but T_{Train} is chosen to be equal to $0.7 \times T_{\text{sim}}$ which makes it collect 70% of training samples. The effects on the ML algorithm are demonstrated in Table 4. The delay here refers to wait time for the ML algorithm to become active.

Table 4 indicates that, for $T_{\text{sim}} = 900\,\text{ms}$ and $T_{\text{Train}} = 0.7 \times T_{\text{sim}}$, an unwanted additional delay of 596 ms is introduced compared to 34 ms delay which could have been avoided had it been $T_{\text{Train}} = T_c$. This additional delay causes underutilization of the model because the model activates with additional delay; thus, a smaller number of attempts are governed by the ML algorithm, where it was supposed to govern 285307 attempts compared to 89286; thus, the model remains inactive for 196021 attempts which is 68.70% of total handover attempts.

Therefore, to improve response time of the proposed model and make it efficient without the loss to HSR, T_{Train} should follow model training equations (13) and (14) according to the applicable scenario.

Similarly, for $T_{\text{sim}} = 800\,\text{ms}$, under similar conditions, an additional delay of 522 ms is introduced in comparison to what could have been 38 ms delay due to the incorrect training sequence. Hence, the proposed model only executes for 79467 attempts instead of 251186 due to the incorrect training sequence. Thus, it misses 68.36% of attempts for which it is supposed to execute.

Since the proposed model achieves significant improvement in terms of response time, therefore it is important to examine if there is any trade-off between HSR values, and Table 5 gives this information.

From Table 5, it is observed that there is no significant impact on HSR values despite the fact that two different training sequences are used; in fact, the HSR values have slightly improved because although the model collects lesser number of training samples but are accurate and are collected within T_c so now the model can accurately predict CSI. For both the simulation periods of 900 ms and 800 ms, the difference between HSR values remains

Table 4: Effects of the improper training sequence on the proposed model; number of UEs: 810.

Algorithm	T_{Sim} (ms)	T_{Train} (ms)	Delay due to incorrect training sequence (ms)	Attempts after training period	Δ attempts after training period	% attempts missed
$T_{\text{Train}} = T_c$	900	34	596	285307	196021	68.70
$T_{\text{Train}} = 0.7 \times T_{\text{sim}}$	900	630		89286		
$T_{\text{Train}} > T_c$	800	38	522	251186	171719	68.36
$T_{\text{Train}} = 0.7 \times T_{\text{sim}}$	800	560		79467		

∗Δ refers to difference in all tables.

Table 5: Difference in HSR values for $T_{\text{Train}} = T_C$ and $0.7 \times T_{\text{sim}}$. Number of UEs: 810.

Algorithm	T_{Sim} (ms)	T_{Train} (ms)	Total attempts	Attempts after training period	Δ attempts after training period	Number of failures after training	HSR (%)	ΔHSR (%)
$T_{\text{Train}} = T_c$	900	34	296035	285307	196021	6396	97.75	0.33
$T_{\text{Train}} = 0.7 \times T_{\text{sim}}$	900	630		89286		2297	97.42	
$T_{\text{Train}} > T_c$	800	38	263229	251186	171719	5841	97.67	0.14
$T_{\text{Train}} = 0.7 \times T_{\text{sim}}$	800	560		79467		1959	97.53	

0.33% and 0.14%, respectively, despite of major improvement in system response time.

Further, the tests results from Table 4 shows that efficiency and response time of the system gets limited if the training period is not chosen correctly and the proposed model remains underutilized. Since authors in [1] focused on HSR only, which demonstrates handover success rate and does not show system efficiency in terms of response time, therefore this important observation is not discussed in their work.

Oversampling of training data can become a bottle neck when 5G networks are involved with Internet of Things (IoT) because, in that case, the number of handover attempts is expected to be significantly large, and this delay in response time could cause serious performance degradation. Therefore, to eliminate this limitation, the proposed work argues the importance of the training sequence.

Similarly, it is observed from test results given in Table 6 that T_{Train} is chosen to be equal to T_c instead of $T_{\text{Train}} = 0.7 \times T_{\text{sim}}$; for $T_s \leq 200$ ms, it is observed that incorrect T_{Train} introduces a delay of 10 ms and misses 16.34% which is 3352 attempts in case of $T_{\text{sim}} = 200$ ms, and for $T_{\text{sim}} = 100$ ms, the ML algorithm does not even respond because the sample collection period exceeds T_{sim}.

Similarly, the impact on HSR is given by Table 7 below; for $T_s \leq 200$ ms, it is observed that there is no significant impact on HSR with which every training sequence is used; however, the model response time and its efficiency does get affected.

3.1. Achieving Improved Results with Less Samples. As discussed, the ML algorithm preferably requires 70% of training samples, while the proposed algorithm requires these samples to be collected within channel coherence time. This concept works best till $T_{\text{sim}} \leq 200$ ms but as soon T_{sim} gets greater than 200 ms, channel coherence time gets small enough where collection of 70% of training samples gets impossible to be collected within T_c. The contest arises that since the proposed model strictly follows T_c for sample collection, therefore it sacrifices all samples which are way beyond T_c and thus collects way less than 70% of total required samples but still the model does not show any degradation in HSR, and premature activation of the model also provides improved model response time. Therefore, this section addresses how the proposed model addresses this limitation with promising results.

$T_2 = 0.7 \times T_{\text{sim}}$, basically, refers to the concept where it uses 70% of samples as training data of an entire simulation time. However, for cases above 200 ms, this training sequence exceeds T_c, and thus, it introduces oversampling, underutilization, and overfitting of the model. However, XGBoost comes equipped with advanced error control mechanisms which can sufficiently be exploited to deal with missing sample information and thus by limiting the model to gather samples within coherence time at scenarios where the sample collection period exceeds T_c; the proposed model can still provide promising results without losing prediction accuracy. Therefore, in all given scenarios, where T_c is longer than $0.7 \times T_{\text{sim}}$, the model should collect 70% of training samples. However, where T_c is lower compared to $0.7 \times T_{\text{sim}}$, then the model should be trained with whatever samples can be gathered within T_c because the error control mechanism can deal with missing samples effectively as discussed in Sections 1 and 2. Moreover, there is no need to collect samples beyond T_c since the channel does not remain static beyond that time.

3.2. UE Level Analysis. This section demonstrates how the ML sequence is limited to within T_c at the UE level supported by graphical figures with respect to RSRP (reference signal received power) where the mobile requests for handover. In this section, the baseline algorithm refers to classical handover attempts where no AI is used, while the proposed refers to the handover mechanism proposed within this work.

Table 6: Difference in HSR values for $T_{\text{Train}} = T_C$ and $0.7 \times T_{\text{sim}}$. Number of UEs: 810.

Algorithm	Time (ms)	T_{Train} (ms)	Introduced time delay due to $T_{\text{Train}} > T_c$ (ms)	Attempts after training period	Δ attempts	% attempts missed
$T_{\text{Train}} = T_c$	200	150	10	17160	3352	16.34
$T_{\text{Train}} = 0.7 \times T_{\text{sim}}$	200	140		20512		
$T_{\text{Train}} > T_c$	100	299	229	ML algorithm does not activate because $T_c > T_{\text{sim}}$		
$T_{\text{Train}} = 0.7 \times T_{\text{sim}}$	100	70		10439		

Table 7: Difference in HSR values for $T_{\text{Train}} = T_C$ and $0.7 \times T_{\text{sim}}$. Number of UEs: 810.

Algorithm	T_{Sim} (ms)	T_{Train} (ms)	Total attempts	Attempts after training period	Δ attempts after training period	Number of failures after training	HSR (%)	ΔHSR (%)
$T_{\text{Train}} = T_c$	200	150	66213	17160	3352	396	97.69	0.05
$T_{\text{Train}} = 0.7 \times T_{\text{sim}}$	200	140		20512		462	97.74	
$T_{\text{Train}} > T_c$	100	299		ML algorithm does not activate because $T_c > T_s$				
$T_{\text{Train}} = 0.7 \times T_{\text{sim}}$	100	70	33087	10439	N/A	222	97.87	N/A

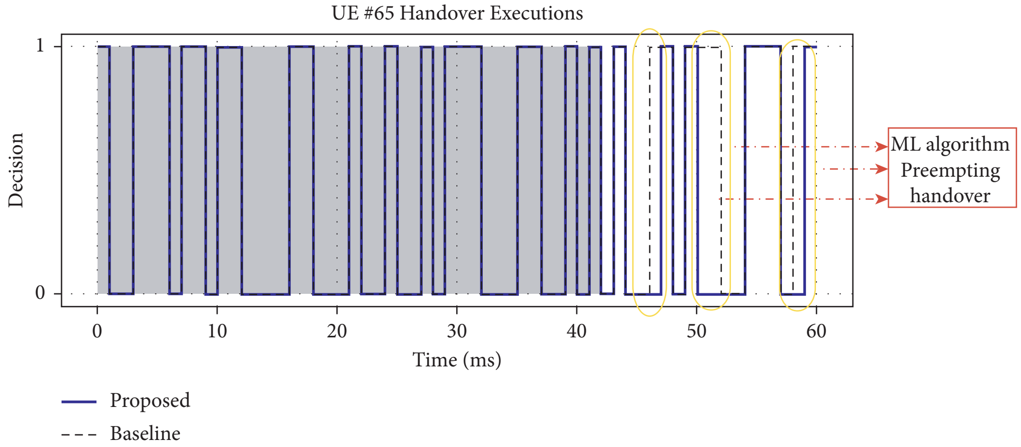


Figure 3: ML algorithm pre-empting failures (training period = 42 ms).

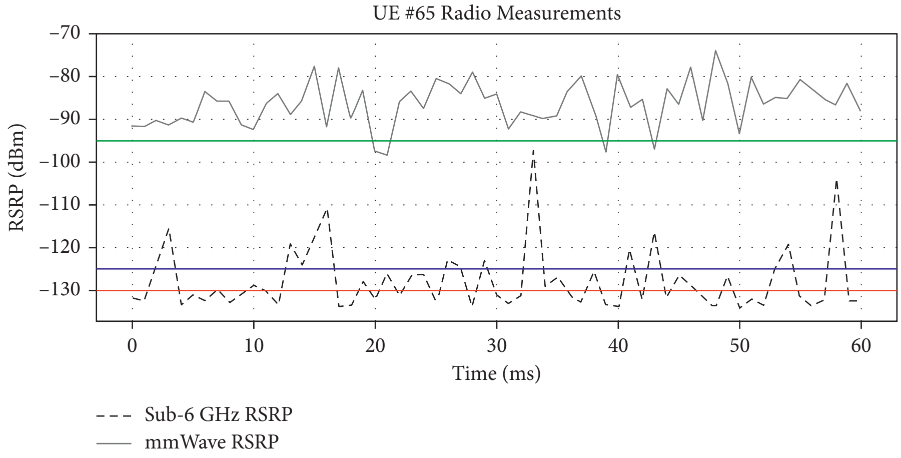


Figure 4: RSRP measurements at various time slots.

Table 8: UE level analysis (UE#65).

Model	T_{Sim} (ms)	Attempts (after ML)	Failures	HSR (%)	FPA (%)
Baseline	60	12	4	58.33	33.33
Proposed	60	7 (5 handovers pre-empted)	0	100	0

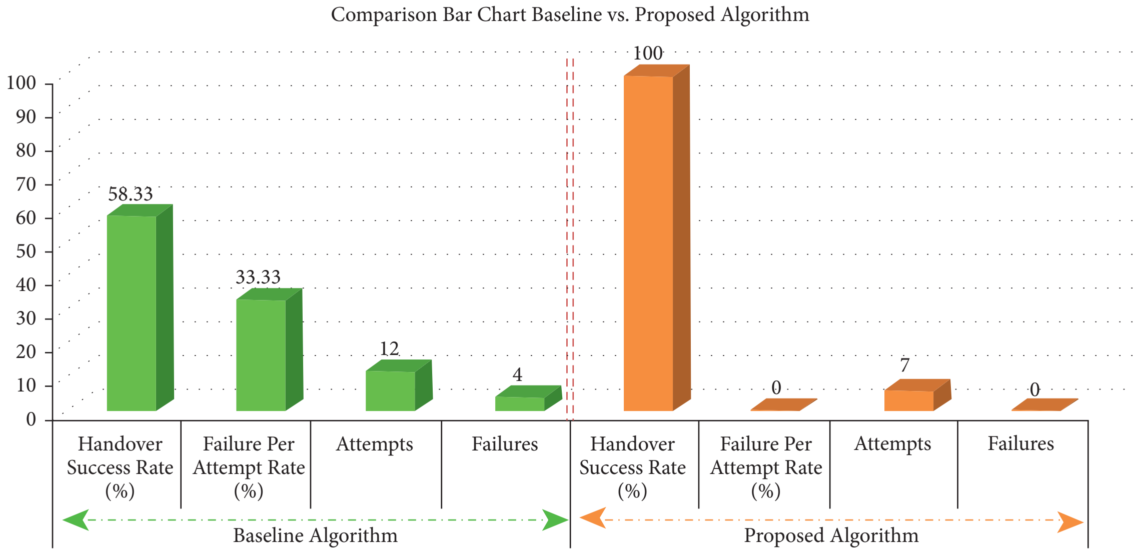
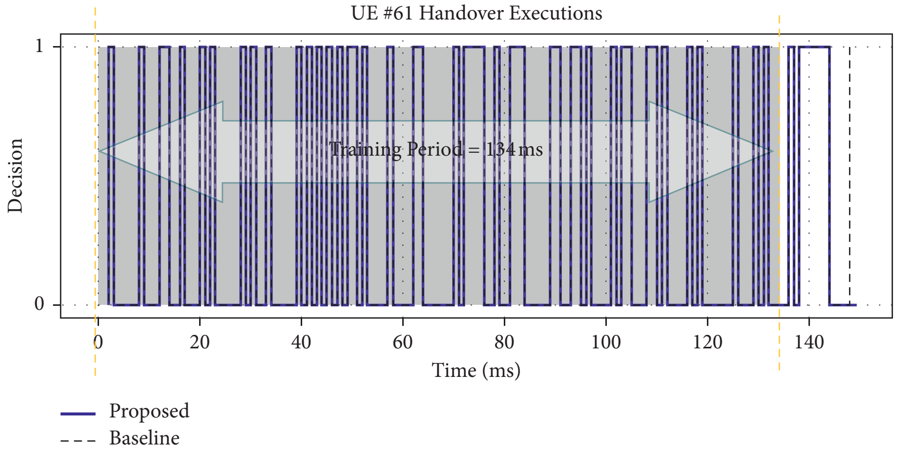
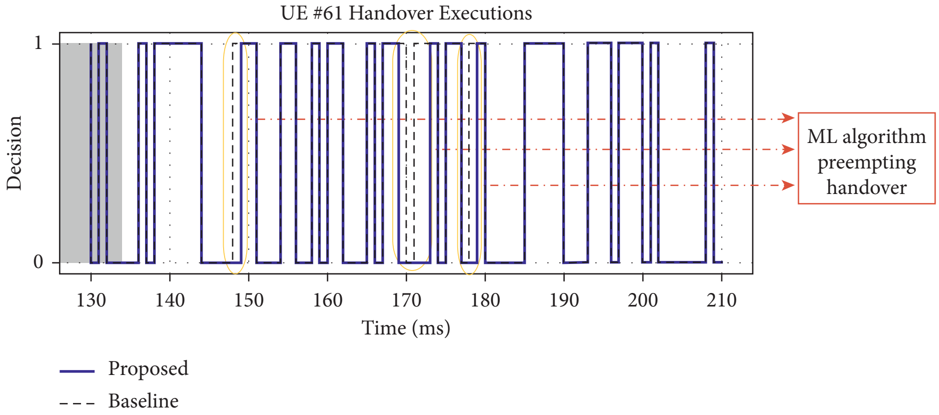


Figure 5: Comparison bar chart baseline vs. proposed algorithm UE#65.



(a)



(b)

Figure 6: (a) $T_{\mathrm{Train}} < 147$ ms (to satisfy coherence time condition). (b) $T_{\mathrm{Train}} < 147$ ms (figure split into two to accommodate).

Table 9: UE level analysis (UE#61).

Model	T_{Sim} (ms)	Attempts (after ML)	Failures	HSR (%)	Failure per attempt (%)
Baseline	210	36	5	86.84	13.15
Proposed	210	31 (5 handovers pre-empted)	0	100	0

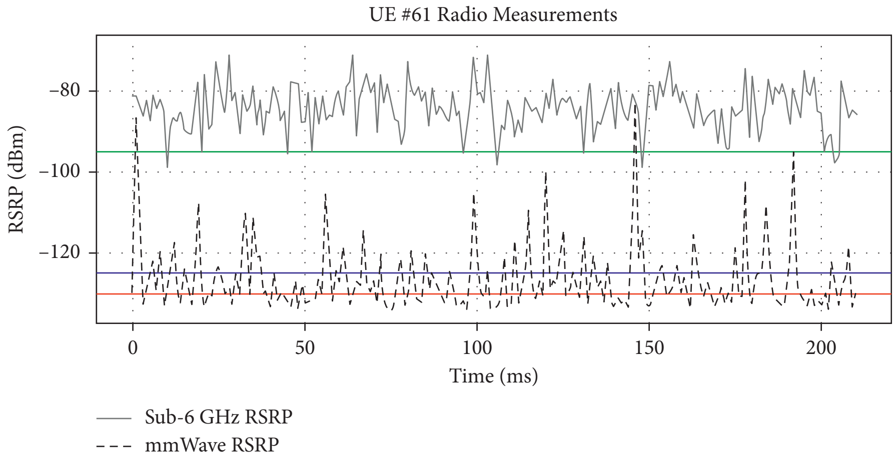


Figure 7: RSRP measurements at various time slots.

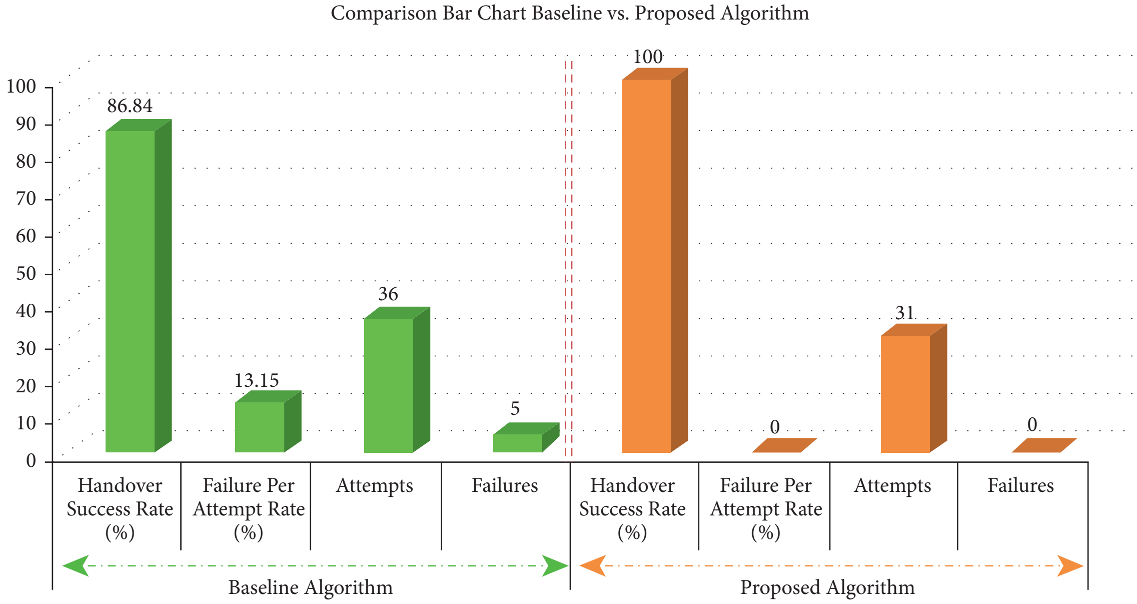


Figure 8: Comparison bar chart baseline vs. proposed algorithm UE#61.

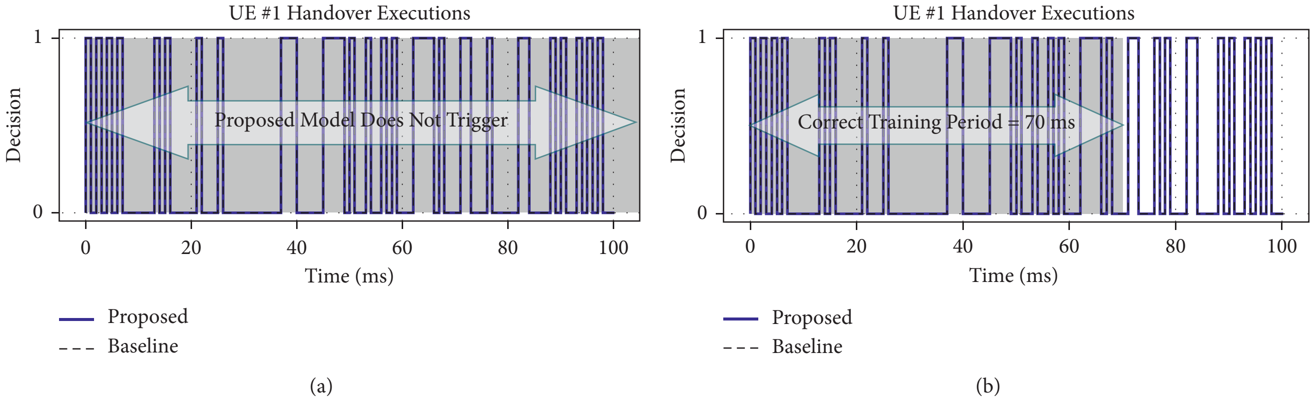


Figure 9: (a) ML algorithm does not trigger due to $T_{\text{Train}} > T_{\text{Sim}}$. (b) ML algorithm now triggers due to correct T_{Train}.

Table 10: BS level analysis. Number of UEs: 810.

Algorithm	T_{Sim} (ms)	T_{Train} (ms)	T_r (ms)	Reduction in delay (ms)	Total attempts	Attempts after training period	Δ attempts after training period	% increased attempts	Number of failures	HSR (%)	ΔHSR (%)
T_1	900	34	34	596	296035	285307	196021	68.70	6396	97.75	0.33
T_2	900	630				89286			2297	97.42	
T_1	800	38	38	522	263229	251186	171719	68.36	5841	97.67	0.14
T_2	800	560				79467			1959	97.53	
T_1	700	43	43	446	230216	216543	146836	67.80	5168	97.61	0.1
T_2	700	489				69707			1732	97.51	
T_1	600	50	50	370	197628	177164	118722	67.01	4460	97.48	0.04
T_2	600	420				58442			1492	97.44	
T_1	500	60	60	290	164680	145329	95327	65.59	3761	97.41	0.03
T_2	500	350				50002			1277	97.44	
T_1	400	75	75	205	131787	107514	67361	62.65	2597	97.58	0.03
T_2	400	280				40153			982	97.55	
T_1	300	100	100	110	98862	66448	36411	54.79	1706	97.43	0.1
T_2	300	210				30037			739	97.53	
T_1	200	150	140	10	66213	17160	3352	16.34	396	97.69	0.05
T_2	200	140				20512			462	97.74	
T_2				Not applicable for $T_{\text{sim}} \leq 100$ ms as discussed in Section 3.1							
T_1	100	299	70	—	33087	10336	—	—	343	96.68	—

*For $T_{\text{sim}} \leq 200$ ms, the 70–30% data spit condition is fine since it does not breach T_c limitations.

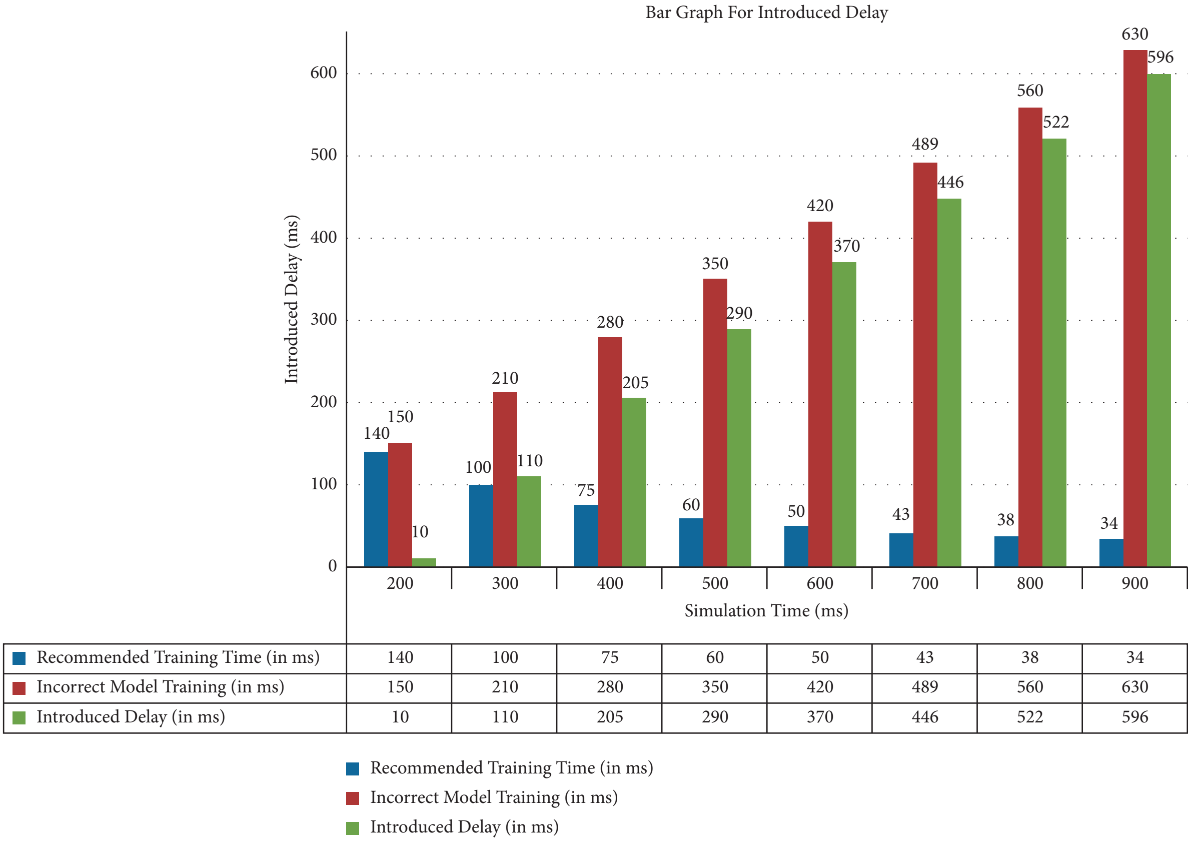
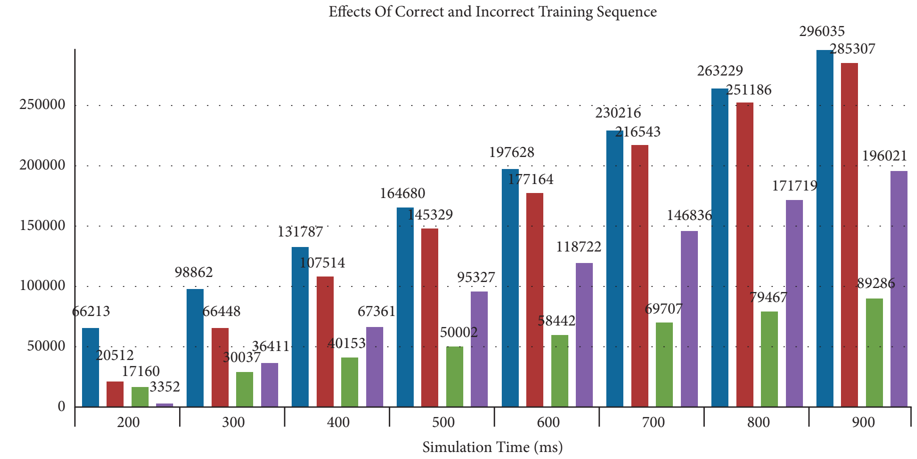


Figure 10: Introduced delay due to the incorrect training sequence.

Case 1. $T_{\text{Train}} = 0.7 \times T_{\text{Sim}}$ for $0.7 \times T_{\text{Sim}} \leq T_{\text{Coherence}}$.

For this purpose, random UE#65 is chosen; Figure 3 demonstrates that the proposed model follows a training period of 42 ms ($T_{\text{Train}} = 0.7 \times T_{\text{sim}}$), where $T_{\text{sim}} = 60$ ms. Once the model gets trained, it overrides the baseline algorithm where a handover failure is expected. Figure 3

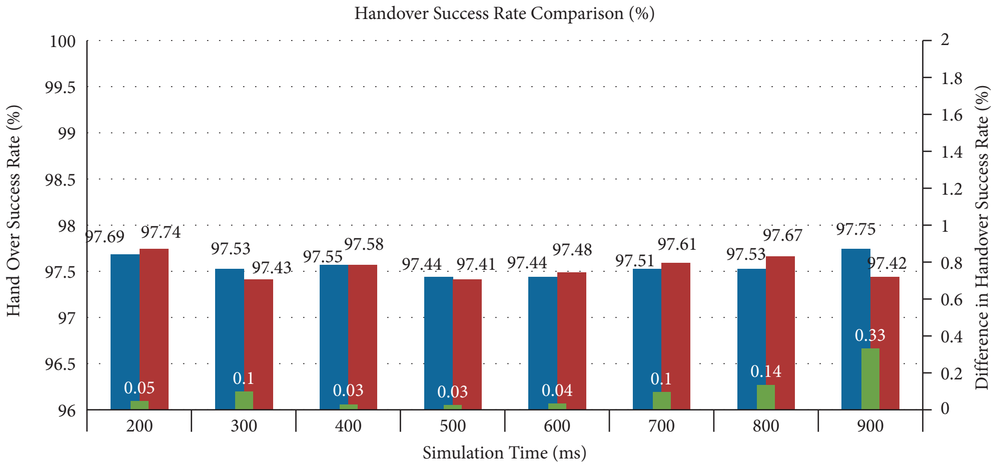


	200	300	400	500	600	700	800	900
🟦 Total Attempts	66213	98862	131787	164680	197628	230216	263229	296035
🟥 Attempts After Training Period (Correct)	20512	66448	107514	145329	177164	216543	251186	285307
🟩 Attempts After Training Period (incorrect)	17160	30037	40153	50002	58442	69707	79467	89286
🟪 Attempts Missed Due To Incorrect Training Period	3352	36411	67361	95327	118722	146836	171719	196021

🟦 Total Attempts 🟩 Attempts After Training Period (incorrect)

🟥 Attempts After Training Period (Correct) 🟪 Attempts Missed Due To Incorrect Training Period

Figure 11: Model misses to execute handover attempts due to the incorrect training period.



	200	300	400	500	600	700	800	900
🟦 HRS With Incorrect Training Sequence	97.69	97.53	97.55	97.44	97.44	97.51	97.53	97.75
🟥 HSR With Correct Training Sequence	97.74	97.43	97.58	97.41	97.48	97.61	97.67	97.42
🟩 Difference in HSR	0.05	0.1	0.03	0.03	0.04	0.1	0.14	0.33

🟦 HRS With Incorrect Training Sequence

🟥 HSR With Correct Training Sequence

🟩 Difference in HSR

Figure 12: No loss to HSR or prediction accuracy. In fact, the proposed model HSR is slightly better despite of lower number of collected samples.

shows the time slots highlighted within yellow colour where ML overrides the baseline algorithm in order to pre-empt failing handover attempt (the same concept is used for other similar figures ahead). Figure 4 illustrates RSRP levels at various time slots.

The following equation is used to compute percentage failure per attempt:

$$\text{Failure Per Attempt } (FPA)\ (\%) = \frac{\text{number of failures}}{\text{total attempts}} \times 100. \tag{31}$$

Table 8 shows that, with baseline algorithm HSR and FPA remain 58.33% and 33.33%, respectively, while with proposed, they are 100% and 0%, respectively. The attempts that are likely to fail are pre-empted with the ML algorithm.

Figure 5 shows the comparison bar chart for the above case.

Case 2. $T_{\text{Train}} = T_C$ for $0.7 \times \text{T}_{\text{sim}} > T_{\text{Coherence}}$.

Now, for $0.7 \times T_{\text{sim}} > T_C$, UE#61 is randomly selected; as per the proposed model, T_{Train} should remain in limits of T_C. Therefore, for Figure 6(a), T_{Train} is supposed to be 147 ms according to $0.7 \times T_{\text{sim}}$; however, the proposed model limits the training sequence to only 134 ms for the reason already mentioned (coherence time), as shown in Figure 6(a). Table 9 and Figure 6(b) show that, with the baseline algorithm HSR and FPA as 86.84% and 13.15%, respectively, while with proposed technique, HSR and FPA are 100% and 0%, respectively. Figures 7 and 8 show RSRP and comparison bar chart, respectively, for this case.

Case 3. $T_{\text{Train}} = T_C$ for $\text{T}_{\text{Train}} > T_{\text{Sim}}$.

If $T_{\text{Train}} = T_c$ is implemented for $T_{\text{Train}} > T_{\text{sim}}$, then the proposed algorithm never gets triggered and it remains in the learning state for time greater than the simulation period, as given by Figure 9(a), but does trigger (as shown in Figure 9(b)) if T_{Train} is chosen as per given policy in equations (14) and (15). Therefore, the proposed work argues the dynamic training period depending upon given scenarios.

3.3. BS Level Analysis. This section provides detailed analysis at the BS level; Table 10 shows that, by selecting the proposed training sequence (T_r), which is also referred to as the correct training sequence in this work, the model response time can be significantly improved without trading off HSR or accuracy by not even taking 70% of required samples. In fact, it is observed that HSR slightly improved because the samples are now gathered within T_c, and the proposed model now estimates the channel with samples that truly depict CSI because if the model keeps on collecting samples beyond T_c, the channel does not correlate to its original state. The test results also show that, since the model now activates timely, therefore it remains active for significantly higher number of attempts; thus, the proposed model does not get underutilized and can be exploited for its full efficiency. The effects of underutilization and reduction in delay are more significant after 200 ms of T_s.

The test results also show such as at $T_s \leq 200$ ms, 70% of samples can be taken without breaching T_c so the model should be allowed to do so. Precisely, the proposed model should be allowed to collect 70% of training samples if it does not by-pass T_c; otherwise, the model should be restricted to collect samples within T_c.

Figure 10 shows the comparison between two training periods T_1 and T_2 and shows that, with the incorrect training period, a significant amount of delay gets introduced in model response time which is highly undesired in 5G and next generation wireless networks. Thus, by implementing proposed technique, this unwanted additional delay can be eliminated.

Figure 11 shows the total number of attempts for complete T_s and then specifies the number of attempts executed through the ML algorithm with the correct and incorrect training sequence. It is observed that the model remains active for more attempts with the correct training sequence in comparison to the incorrect training period, thus eliminating underutilization of the model.

Figure 12 shows that although model response time is significantly improved, the model remains active for more attempts but still there is no loss to HSR or prediction accuracy of the model despite the fact that the proposed model is starved of required training samples at scenarios were collecting 70% of required samples is not possible but the XGBoost built-in algorithm for compensating missing samples keeps the proposed algorithm working. In fact, it is observed from Figure 12 that HSR is slightly improved with the correct training sequence.

4. Conclusion

It is inevitable that, for 5G and future networks, AI will dominate major network tasks in order to provide best QoS possible. Speed, accuracy, and efficiency of these intelligent algorithms are dominating factors in this field. In the ML algorithm, determining exact data split, i.e., percentage of training data and test data, is challenging because the inequilibrium between both leads to either underfitting or overfitting of the model which is highly undesirable. However, generally a data split of 70–30% of training and test data is preferred but still does not guarantee optimum results. The proposed research work exploits this uncertainty of the data split mechanism and argues to train the model within coherence time, thus keeping the data split dynamic. Thus, by saving time that is spent in collecting samples, model response time can be significantly improved; however, the ML technique should be robust enough to reduce errors caused due to missing samples.

In order to estimate CSI, it is mandatory that samples be collected strictly within coherence time because the samples gathered after this period do not provide accurate channel state; therefore, the proposed work argues to avoid sample collection after that period. Therefore, this work specifies the algorithm for switching the training sequence if the training period exceeds coherence time limits. If the training sequence is not limited to within coherence time, then it is in contradiction to model design parameters which adds high-level delays and blocks the ML algorithm to perform at full capacity because the proposed model does not activate timely in order to predict mmWaves' signal strength.

At some instances, it is observed that upto 100% improvement can be achieved at certain UEs. However, no 100% improvement might be possible at the network level because the proposed model only activates where ROC-AUC is above 0.7. Therefore, at places where the proposed model predictions are not accurate, the baseline algorithm remains active. Thus, due to this limiting factor that the baseline algorithm cannot be fully overridden makes 100% achievement at the network level almost impossible at this stage.

The proposed model also suggests that the ML algorithm should be available in both UE and base station in contrast to previous authors' work because if the base station is not working at its full processing load, then the algorithm should be processed by the base station in order to protect UE processing and battery resources; otherwise, it should be processed by UE in order to relieve the base station in case of processing overload.

For future, one possible research direction can be optimizing the XGBoost algorithm for mobile applications only in order to make it more efficient and mobile-friendly.

Data Availability

The data used to support the findings of the study are available within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] F. B. Mismar and B. L. Evans, "Partially blind handovers for mmWave new radio aided by sub-6 GHz LTE signaling," in *Proceedings of the 2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–5, Kansas City, MO, USA, May 2018.

[2] M. S. Khorsheed and A. O. Al-Thubaity, "Comparative evaluation of text classification techniques using a large diverse Arabic dataset," *Language Resources and Evaluation*, vol. 47, no. 2, pp. 513–538, 2013.

[3] Q. H. Nguyen, H.-B. Ly, L. S. Ho et al., "Influence of data splitting on performance of machine learning models in prediction of shear strength of soil," *Mathematical Problems in Engineering*, vol. 2021, Article ID 4832864, 2021.

[4] N. Leema, H. K. Nehemiah, and A. Kannan, "Neural network classifier optimization using differential evolution with global information and back propagation algorithm for clinical datasets," *Applied Soft Computing*, vol. 49, pp. 834–844, 2016.

[5] D. V. Dao, H. Adeli, H.-B. Ly et al., "A sensitivity and robustness analysis of GPR and ANN for high-performance concrete compressive strength prediction using a Monte Carlo simulation," *Sustainability*, vol. 12, no. 3, p. 830, 2020.

[6] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, San Francisco, CA, USA, August 2016.

[7] H. Yang, A. Alphones, Z. Xiong, D. Niyato, J. Zhao, and K. Wu, "Artificial-intelligence-enabled intelligent 6G networks," *IEEE Network*, vol. 34, no. 6, pp. 272–280, 2020.

[8] A. Singh, S. P. Singh, U. N. Tripathi, and M. J. i. Mishra, "Optimizing call drops in cellular network using artificial intelligence based handover schema," vol. 6, no. 1, 2017.

[9] D. Zhu, J. Choi, and R. W. Heath, "Auxiliary beam pair enabled AoD and AoA estimation in closed-loop large-scale millimeter-wave MIMO systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4770–4785, 2017.

[10] H. B. H. Dutty and M. M. Mowla, "Channel modeling at unlicensed millimeter wave V band for 5G backhaul networks," in *Proceedings of the 2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, pp. 769–773, Dhaka, Bangladesh, September 2019.

[11] X. Song, S. Haghighatshoar, and G. Caire, "Efficient beam alignment for millimeter wave single-carrier systems with hybrid MIMO transceivers," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1518–1533, 2019.

[12] S. Dahal, E. A. Stephanou, N. Talukdar, S. Ahmed, H. King, and M. Faulkner, "Millimetre wave propagation reverse measurements for 5G urban micro scenario," in *Proceedings of the 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pp. 1–5, Kuala Lumpur, Malaysia, April-May 2019.

[13] N. Garcia, H. Wymeersch, and D. T. M. Slock, "Optimal precoders for tracking the AoD and AoA of a mmWave path," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5718–5729, 2018.

[14] Q. Qin, L. Gui, P. Cheng, and B. Gong, "Time-varying channel estimation for millimeter wave multiuser MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9435–9448, 2018.

[15] Y. Haifen, F. Yutao, L. Dong, Z. Zhi, and L. Shuisheng, "Compressive sensing and prior support based adaptive channel estimation in massive MIMO," in *Proceedings of the 2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pp. 1618–1622, Chengdu, China, October 2016.

[16] J. Yang, Z. Wei, X. Zhang, N. Li, and L. Sang, "Correlation based adaptive compressed sensing for millimeter wave channel estimation," in *Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, San Francisco, CA, USA, March 2017.

[17] C. Rusu, N. González-Prelcic, and R. W. Heath, "Low resolution adaptive compressed sensing for mmWave MIMO receivers," in *Proceedings of the 2015 49th Asilomar Conference on Signals, Systems and Computers*, pp. 1138–1143, Pacific Grove, CA, USA, November 2015.

[18] S. Sun and T. S. Rappaport, "Millimeter Wave MIMO channel estimation based on adaptive compressed sensing," in *Proceedings of the 2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 47–53, Paris, France, May 2017.

[19] A. Liao and Z. Gao, "Super-resolution channel estimation for mmWave massive MIMO," in *Proceedings of the 2018 IEEE International Conference on Communications (ICC)*, pp. 1–5, Kansas City, Kansas, USA, May 2018.

[20] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and R. S. Blum, "Low-rank tensor decomposition-aided channel estimation for millimeter wave MIMO-OFDM systems," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 7, pp. 1524–1538, 2017.

[21] C. Yeh, K. Hsu, J. Chi, and Y. Huang, "Adaptive simultaneous orthogonal matching Pursuit for mmWave hybrid beam tracking," in *Proceedings of the 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, pp. 1–5, Shanghai, China, November 2018.

[22] J. Lee, G.-T. Gil, and Y. H. Lee, "channel estimation via orthogonal matching Pursuit for hybrid MIMO systems in millimeter wave communications," *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2370–2386, 2016.

[23] W. Zhong, L. Xu, Q. Zhu, X. Chen, and J. Zhou, "MmWave beamforming for UAV communications with unstable beam pointing," *China Communications*, vol. 16, no. 1, pp. 37–46, 2019.

[24] L. Weiland, C. Stöckle, M. Würth, T. Weinberger, and W. Utschick, "OMP with grid-less refinement steps for compressive mmWave MIMO channel estimation," in *Proceedings of the 2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 543–547, Sheffield, UK, July 2018.

[25] A. N. Uwaechia, N. M. Mahyuddin, M. F. Ain, N. M. Abdul Latiff, and N. F. Za'bah, "On the spectral-efficiency of low-complexity and resolution hybrid precoding and combining transceivers for mmWave MIMO systems," *IEEE Access*, vol. 7, pp. 109259–109277, 2019.

[26] V. F. Monteiro, D. A. Sousa, T. F. Maciel, F. R. P. Cavalcanti, C. F. M. e. Silva, and E. B. Rodrigues, "Distributed RRM for 5G multi-RAT multiconnectivity networks," *IEEE Systems Journal*, vol. 13, no. 1, pp. 192–203, 2019.

[27] G. Ying, M. Qingmin, and W. Xiaoming, "Energy-optimized 5G dual connectivity radio resource allocation," in *Proceedings of the 2019 IEEE 2nd International Conference on Electronics Technology (ICET)*, pp. 126–130, Chengdu, China, 2019.

[28] M. Polese, M. Giordani, M. Mezzavilla, S. Rangan, and M. Zorzi, "Improved handover through dual connectivity in 5G mmWave mobile networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 9, pp. 2069–2084, 2017.

[29] A. Ghosh, T. A. Thomas, M. C. Cudak et al., "Millimeter-Wave enhanced local area systems: a high-data-rate approach for future wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1152–1163, 2014.

[30] J. Chen, Y. Wang, Y. Li, and E. Wang, "QoE-aware intelligent vertical handoff scheme over heterogeneous wireless access networks," *IEEE Access*, vol. 6, pp. 38285–38293, 2018.

[31] E. Zeljkovic, N. Slamnik-Kriještorac, S. Latré, and J. M. Marquez-Barja, "ABRAHAM: machine learning backed proactive handover algorithm using SDN," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1522–1536, 2019.

[32] S. I. Majid, S. W. Shah, and S. N. Marwat, "Applications of extreme gradient boosting for intelligent handovers from 4G to 5G (mm waves) technology with partial radio contact," *Electronics*, vol. 9, no. 4, 2020.

[33] J.-h. Bang, S. Oh, K. Kang, and Y.-J. Cho, "A bayesian regression based LTE-R handover decision algorithm for high-speed railway systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10160–10173, 2019.

[34] D. Li, D. Li, and Y. Xu, "Machine learning based handover performance improvement for LTE-R," in *Proceedings of the 2019 IEEE International Conference on Consumer Electronics—Taiwan (ICCE-TW)*, pp. 1-2, Ilan, Taiwan, May 2019.

[35] L. Yan, H. Ding, L. Zhang et al., "Machine learning-based handovers for sub-6 GHz and mmWave integrated vehicular networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4873–4885, 2019.

[36] Y. Huang, C. Xu, C. Zhang, M. Hua, and Z. Zhang, "An overview of intelligent wireless communications using deep reinforcement learning," *Journal of Communications and Information Networks*, vol. 4, no. 2, pp. 15–29, 2019.

[37] T. Nishio, H. Okamoto, K. Nakashima et al., "Proactive received power prediction using machine learning and depth images for mmWave networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2413–2427, 2019.

[38] L. Hao and B. Ng, "Self-healing solutions for Wi-Fi networks to provide seamless handover," in *Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 639–642, Washington DC, USA, April 2019.

[39] S. Aghabozorgi, A. Bayati, K. Nguyen, C. Despins, and M. Cheriet, "Toward predictive handover mechanism in software-defined enterprise wi-fi networks," in *Proceedings of the 2019 IEEE Sustainability through ICT Summit (StICT)*, pp. 1–6, Montreal, Canada, June 2019.

[40] G. Sun, G. T. Zemuy, and K. Xiong, "Dynamic reservation and deep reinforcement learning based autonomous resource management for wireless virtual networks," in *Proceedings of the 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–4, Orlando, Florida, USA, November 2018.

[41] I. AlQerm and B. Shihada, "Energy efficient traffic offloading in multi-tier heterogeneous 5G networks using intuitive online reinforcement learning," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 3, pp. 691–702, 2019.

[42] N. Rastegardoost and B. Jabbari, "A machine learning algorithm for unlicensed LTE and WiFi spectrum sharing," in *Proceedings of the 2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–6, Orlando, FL, USA, November 2018.

[43] R. H. Puspita, S. D. A. Shah, G. Lee, B. Roh, J. Oh, and S. Kang, "Reinforcement learning based 5G enabled cognitive radio networks," in *Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 555–558, Jeju Island, South Korea, October 2019.

[44] M. Ozturk, M. Gogate, O. Onireti, A. Adeel, A. Hussain, and M. A. Imran, "A novel deep learning driven, low-cost mobility prediction approach for 5G cellular networks: the case of the Control/Data Separation Architecture (CDSA)," *Neurocomputing*, vol. 358, pp. 479–489, 2019.

[45] S. Mahfouz, F. Mourad-Chehade, P. Honeine, J. Farah, and H. Snoussi, "Kernel-based machine learning using radio-fingerprints for localization in wsns," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 2, pp. 1324–1336, 2015.

[46] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Kernel-based positioning in wireless local area networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 689–705, 2007.

[47] J. Wu, J. Liu, Z. Huang, and S. Zheng, "Dynamic fuzzy Q-learning for handover parameters optimization in 5G multi-tier networks," in *Proceedings of the 2015 International Conference on Wireless Communications & Signal Processing (WCSP)*, pp. 1–5, Nanjing, China, October 2015.

[48] T. J. Ross, *Fuzzy Logic with Engineering Applications*, Wiley Online Library, Hoboken, NJ, USA, 2004.

[49] P. Muñoz, R. Barco, and I. de la Bandera, "On the potential of handover parameter optimization for self-organizing networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1895–1905, 2013.

[50] F. M. Darus, N. A. Ahmad, and A. F. M. Ariffin, "Android malware classification using XGBoost on data image pattern," in *Proceedings of the 2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, pp. 118–122, Bali, Indonesia, November 2019.

[51] L. M. d. Silva, M. R. Silveira, A. M. Cansian, and H. K. Kobayashi, "Multiclass classification of malicious domains using passive DNS with XGBoost: (work in progress)," in *Proceedings of the 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, pp. 1–3, Cambridge, MA, USA, November 2020.

[52] J. Sun, S. Wang, and J. Du, "Research on classification model of equipment support personnel based on collaborative filtering and xgboost algorithm," in *Proceedings of the 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*, pp. 1339–1343, Dalian, China, December 2017.

[53] Z. Qi, "The text classification of theft crime based on TF-IDF and XGBoost model," in *Proceedings of the 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pp. 1241–1246, Dalian, China, June 2020.

[54] M. T. Vo, T. Nguyen, and T. Le, "Robust head pose estimation using extreme gradient boosting machine on stacked autoencoders neural network," *IEEE Access*, vol. 8, pp. 3687–3694, 2020.

[55] H. T. Weldegebriel, H. Liu, A. U. Haq, E. Bugingo, and D. Zhang, "A new hybrid convolutional neural network and eXtreme gradient boosting classifier for recognizing handwritten Ethiopian characters," *IEEE Access*, vol. 8, pp. 17804–17818, 2020.

[56] R. P. Sheridan, W. M. Wang, A. Liaw, J. Ma, and E. M. Gifford, "Extreme gradient boosting as a method for quantitative structure-activity relationships," *Journal of Chemical Information and Modeling*, vol. 56, no. 12, pp. 2353–2360, 2016.

[57] W. Lu, Y. Rui, Z. Yi, B. Ran, and Y. Gu, "A hybrid model for lane-level traffic flow forecasting based on complete ensemble empirical mode decomposition and extreme gradient boosting," *IEEE Access*, vol. 8, pp. 42042–42054, 2020.

[58] S. Georganos, T. Grippa, S. Vanhuysse, M. Lennert, M. Shimoni, and E. Wolff, "Very high resolution object-based land use-land cover urban classification using extreme gradient boosting," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 4, pp. 607–611, 2018.

[59] 3GPP, "3GPP specification series," 2021, https://www.3gpp.org/DynaReport/36-series.htm.

[60] S. Sun, G. R. MacCartney, and T. S. Rappaport, "Millimeter-wave distance-dependent large-scale propagation measurements and path loss models for outdoor and indoor 5G systems," in *Proceedings of the 2016 10th European Conference on Antennas and Propagation (EuCAP)*, pp. 1–5, Davos, Switzerland, April 2016.

[61] M. M. Lodro, N. Majeed, A. A. Khuwaja, A. H. Sodhro, and S. Greedy, "Statistical channel modelling of 5G mmWave MIMO wireless communication," in *Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1–5, Sukkur, Pakistan, March 2018.

[62] S. Sun, T. S. Rappaport, T. A. Thomas et al., "Investigation of prediction accuracy, sensitivity, and parameter stability of large-scale propagation path loss models for 5G wireless communications," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 2843–2860, 2016.

[63] P. K. Sharma, D. Sharma, and T. V. Sai, "Optimization of propagation path loss model in 4G wireless communication systems," in *Proceedings of the 2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pp. 1245–1248, Piscataway, NJ, USA, 2018.

[64] T. S. Rappaport, Y. Xing, G. R. MacCartney, A. F. Molisch, E. Mellios, and J. Zhang, "Overview of millimeter wave communications for fifth-generation (5G) wireless networks-with a focus on propagation models," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 12, pp. 6213–6230, 2017.

[65] T. S. J. M. J. Rappaport, "Wireless communications—principles and practice," *The Book End)*, vol. 45, no. 12, pp. 128-129, 2002.

[66] I. Bergel and S. Benedetto, "The effective coherence time of common channel models," in *Proceedings of the 2010 IEEE 11th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, Marrakech, Morocco, June 2010.

[67] A. Sorrentino, G. Ferrara, and M. Migliaccio, "On the coherence time control of a continuous mode stirred reverberating chamber," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 10, pp. 3372–3374, 2009.

[68] M. E. Morocho-Cayamcela and W. Lim, "Accelerating wireless channel autoencoders for short coherence-time communications," *Journal of Communications and Networks*, vol. 22, no. 3, pp. 215–222, 2020.

[69] V. Va and R. W. Heath, "Basic relationship between channel coherence time and beamwidth in vehicular channels," in *Proceedings of the 2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pp. 1–5, Boston, MA, USA, September 2015.

[70] D. Bykhovsky, "Coherence time evaluation in indoor optical wireless communication channels," *Sensors*, vol. 20, no. 18, 2020.

[71] M. Kim, T. Lee, T. Im, and H. Ko, "The analysis of coherence bandwidth and coherence time for underwater channel environments using experimental data in the West sea, Korea," in *Proceedings of the OCEANS 2016—Shanghai*, pp. 1–4, Shanghai, China, April 2016.