IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 27 June 2023, accepted 9 July 2023, date of publication 13 July 2023, date of current version 19 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3294990

RESEARCH ARTICLE

A New Method to Improve Frequent-Handover Problem in High-Mobility Communications Using RIC and Machine Learning

BAUD HARYO PRANANTO, (Student Member, IEEE), ISKANDAR, AND ADIT KURNIAWAN
School of Electrical Engineering and Informatics, Bandung Institute of Technology, Bandung 40132, Indonesia

Corresponding author: Baud Haryo Prananto (baud.prananto@students.itb.ac.id)

ABSTRACT The current trend of newer cellular network technology, such as 5G, is using a higher frequency spectrum that causes a smaller cell size. This will further cause a more frequent handover in the high-mobility users like the ones that happen in the high-speed train. In a cellular network, the handover process is very crucial as it may disrupt data transmission. Without a reliable handover process, the high-mobility users may experience problems like a high bit error rate (BER) or even call-drop. The traditional handover algorithm is proven reliable in ideal conditions but may not work correctly in a non-ideal condition such as the presence of a coverage hole. Machine learning can be implemented to improve the handover performance in those conditions. Open Radio Access Network (O-RAN) presents a solution to implement machine learning in the cellular network using a Radio Intelligent Controller (RIC), where we can improve a lot of functionalities in the Radio Access Network (RAN) modularly without modifying the existing RAN network element. The RIC original software is using vector autoregression to determine the target cell by predicting the throughput of each neighboring cell. In this paper, we performed two modifications to the original software: improve the vector autoregression method to consider the User Equipment (UE) movement and replace the vector autoregression method with a neural network. We also prove that these modifications present easier and better target cell determination for the environment with a coverage hole that will be useful for frequent handover in high-mobility users.

INDEX TERMS Cellular, handover, machine learning, neural network, O-RAN.

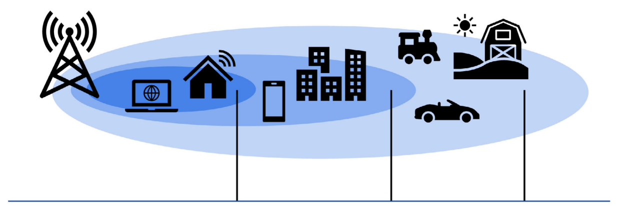
I. INTRODUCTION

One of the characteristics of the new fifth-generation (5G/New Radio - NR) cellular technology is the usage of high-frequency spectrum. While the fourth-generation (4G/Long Term Evolution - LTE) technology is standardized to use up to 3 GHz spectrum, 5G is standardized to use up to 52 GHz [1]. The impact of this high-frequency usage is the smaller cell coverage as high-frequency signals will be attenuated easier. Therefore, in the heterogenous network architecture, the higher frequency spectrum is usually deployed for pico/femtocells and used for home and stationary usage (Fig. 1). For high-mobility usage, macro cells are usually used with a lower frequency spectrum.

The main issue with the macro cell is the availability of the spectrum. In most countries, the lower frequency spectrum is already unavailable, and this forces the network operator to use a higher frequency spectrum. This spectrum scarcity is also the reason behind the standardization of higher frequency in later technology like 5G. Some different use cases also drive the operators to use the high-frequency smaller cells, like in ultra-dense high-capacity sites the higher frequency spectrum may give higher capacity. Using small cells, the operator requires to deploy more cells to cover an area. For high mobility users, such as users in a high-speed train, this condition means more handover activities (Fig. 2).

The associate editor coordinating the review of this manuscript and approving it for publication was Jad Nasreddine.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/



	Pico/Femto	Micro/Small	Macro
Transmit Power	< 1 W	1-10 W	> 10 W
Cell Size	1-10 m	10-100 m	> 100 m
Frequency	> 30 GHz (mmWave)	3-30 GHz (cmWave)	0.5-6 GHz (sub 6GHz)
Usage	Home, Enterprise, Wireless Internet, IPTV/VoD	Urban, Mobile Communication, Hotspots	Rural, Vehicular

FIGURE 1. Typical characteristic of Heterogenous Networks.

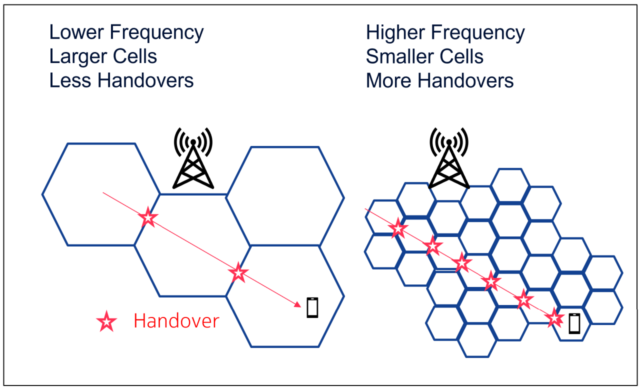


FIGURE 2. Impact of higher frequency and smaller cell.

A. HANDOVER ISSUES

Handover is the procedure that changes the serving cell of a User Equipment (UE) in its connected mode [2] and usually happens when the UE moves its position. This procedure is very crucial in cellular telecommunication as it may interrupt the data transmission [3]. Every handover instance is a potential disruption in continuous data transmission since it may not always be successful. In a successful handover case, the UE always stays in a connected state and the data transmission continuity is guaranteed. However, if the handover is failed the UE is forced to go to idle state and this will interrupt data transmission as the UE must repeat the radio connection establishment procedure.

In the high-speed train, the operator provides an onboard mobile relay station to directly serve the devices inside the trains (Fig. 3). However, to connect to the core network it has to connect to the outdoor cellular network. Due to its extreme speed, which may reach 300 km/h, the network still has to perform frequent handovers, at least in the outdoor channel. Even though the cells are deployed with macro configuration, handover processes are still relatively frequent compared to the stationary users.

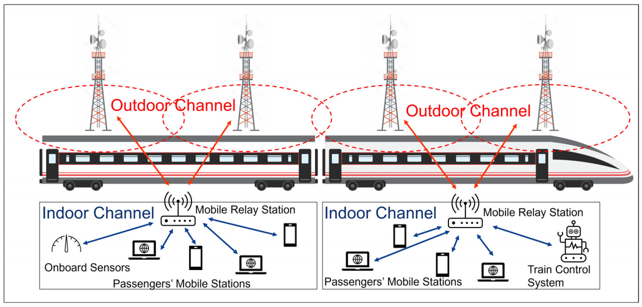


FIGURE 3. The typical network inside a high-speed train.

The UE in a high-speed train may experience degradation in radio conditions due to the doppler effect and this will impact the bit error rate (BER) experienced by the UE. The radio condition may be worsened by the handover process that becomes more frequent because of the mobility speed and typically-smaller cell size. Therefore in this high-speed train case, a reliable handover process is required to ensure transmission continuity in a degrading radio condition.

Besides the traditional handover algorithm that will be further described in Section III, a lot of algorithms are proposed including the ones based on machine learning [4]. Neural networks are one of the most widely used methods for handover improvement [5], [6], [7], [8], [9], [10]. The main issue with those proposed methods is their implementation to cellular networks since machine learning is not originally part of the cellular networks [11]. Most of those studies require major modifications in the existing cellular networks for their implementation. This will raise many problems in the implementation stage.

B. MACHINE LEARNING FOR HANDOVER: STATE OF THE ART

The traditional handover algorithm is sometimes not reliable in a non-ideal network condition like the presence of a coverage hole. This is because the radio condition, and thus the target cell, cannot be predicted solely by the UE measurements. The process of traditional handover algorithm will be described further in Section III.

Some alternative methods are required to determine the target cell in a non-ideal network condition, and one of the approaches is the predictive handover using machine learning. Several studies [4] have implemented machine learning to improve handover performance using the predictive handover method (i.e. predict the target cell using machine learning).

There are several approaches to machine learning: supervised learning, unsupervised learning, and reinforcement learning [12]. Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs [13]. Unsupervised learning looks for previously undetected patterns in a data set with no pre-existing labels and with a minimum of human supervision [14]. Reinforcement learning is concerned with how software agents ought to take action in an environment in order to maximize the notion of cumulative reward [15].

Supervised learning is widely used for handover improvement. The neural networks (NN) method is one of the most popular techniques used in several studies [5], [6], [7], [8], [9], [10]. Some studies use support vector machine [16] and K-nearest neighbor [17], [18]. Unsupervised learning techniques are also used by some studies, for example, K-means [19], [20] and long short-term memory [21]. Reinforced learning is used by some researchers that usually employ Q-learning algorithms [22], [23].

The neural networks method seems popular in mobility management improvement studies. The basic idea behind these studies is to use the concept of neural networks to learn a mobility-based model for every user in the network and then make predictions of which cell the user is most likely to be next [4].

C. RESEARCH MOTIVATION AND CONTRIBUTION

The limitation of those existing studies is they perform simulations on improving base station algorithms by adding machine learning techniques. Therefore, to implement machine learning in RAN, the base station software must be heavily modified.

With the emergence of Open Radio Access Network (O-RAN), especially the introduction of Radio Intelligent Controller (RIC), there is a possibility to implement machine learning to RAN without modifying the existing network elements. The machine learning algorithm can also be implemented modularly without disturbing the existing base station software.

O-RAN consortium [24] introduces the Near Real Time Radio Intelligent Controller (Near-RT RIC), a new additional network element in the radio access network (RAN) that can host any application to control base stations. Using this Near-RT RIC, a machine learning algorithm for improving the handover process can be implemented modularly without major modifications to the existing cellular networks.

Our research implements machine learning in Near-RT RIC because of its modularity aspect. The machine learning algorithm can be implemented modularly outside the base station without modifying the current software of the base station.

The motivation of this research is to solve the handover reliability issue in a non-ideal network by using Near-RT RIC where the machine learning algorithm to control the handover process can be implemented modularly without major modification of the existing network elements.

The contribution of this research is designing and implementing a machine-learning-based handover algorithm in Near-RT RIC to control the handover process modularly. In this paper, we performed modifications in Near-RT RIC original software. We have done two modifications: adapt

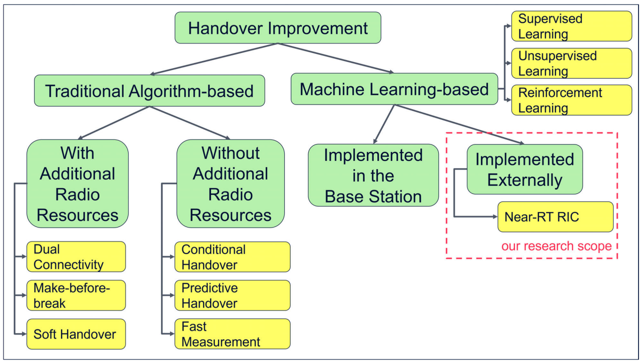


FIGURE 4. **Solution Taxonomy for Handover Improvement.**

the vector autoregression (VAR, the original algorithm used in Near-RT RIC) to consider the UE movement (described in Section V) and replaced the vector autoregression with neural network (described in Section VI). We have done simulations to test the performance of those methods and we also studied the effect of training data amount on the handover performance (described in Section VII). Finally, we compared the performance of our two proposed methods with the traditional handover algorithm and showed that the machine-learning-based handover in Near-RT RIC performs better in a non-ideal condition, in this case, a network with a coverage hole (described in Section VIII).

D. RESEARCH SCOPE AND LIMITATION

This research focuses on the usage of a machine learning algorithm in Near-RT RIC to control the handover process. In this study, we compare the performance of the handover control in Near-RT RIC with the baseline traditional handover algorithm.

As described in Fig. 4, there are various solutions to improve the handover performance. In our research, we focus only on the machine-learning-based solution that is implemented externally for modularity reasons. The usage of another innovation on top of the traditional handover algorithm such as soft handover, conditional handover, make-before-break, is not considered and not compared with our proposed machine-learning-based algorithm in Near-RT RIC. The state-of-the art machine learning algorithms for handover improvement are also not considered.

II. SYSTEM AND NETWORK MODEL

As defined by 3GPP [25], the cellular network is divided into the Radio Access Network (RAN) and Core Network (Fig. 5). In this research, we are working mainly in the RAN. The RAN consists of base stations that provide cells to serve the User Equipments (UE). As the UE moves, it may be served by another cell from another base station and this involves the handover process. The handover process will be further described in Section III.

Additionally, as defined by O-RAN, Radio Intelligent Controller (RIC) can be connected to the base stations. RIC will

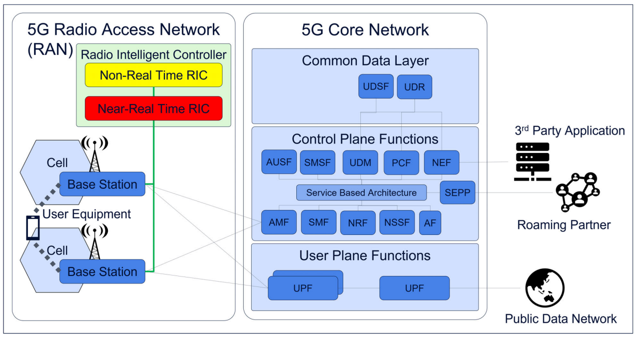


FIGURE 5. 5G NR System and Network Model [25].

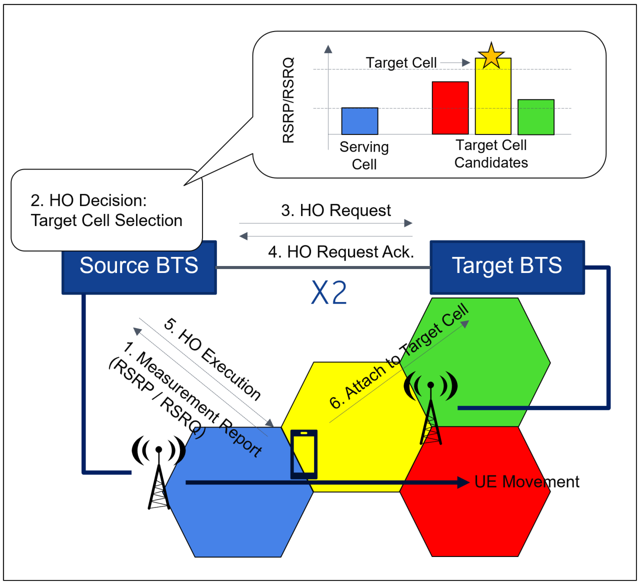


FIGURE 6. Traditional Handover Algorithm [2].

provide additional functions including the machine-learning-based handover algorithm. More detailed explanation about O-RAN RIC can be found in Section IV.

The functions run by base stations, cells, and UE are simulated using the NS3 network simulator [26]. The RIC is implemented as an application on an Ubuntu-based virtual machine. The simulation design and process will be explained in detail in Section VII.

III. TRADITIONAL HANDOVER ALGORITHM

In a traditional handover algorithm [27], the UE sends measurement reports about the serving cell and neighbor cells to the serving base station. This measurement report typically contains information about the cell's signal strength (Reference Signal Received Power - RSRP) and/or signal quality (Reference Signal Received Quality - RSRQ). The measurement reports will be analyzed by the serving base station to determine the target cell for the handover destination, typically the best-measured neighbor cell (Fig. 6).

The handover process introduces some interruptions in the data transmission since it disconnects the UE from the serving cell (thus temporarily stopping the data transmission) and connects again to the target cell. These interruptions must be minimized and the performance of the handover process can be improved by minimizing the Mobility Interruption Time (MIT). 3GPP defines MIT as the shortest time duration supported by the system during which a user terminal cannot exchange user plane packets with any base station during transitions [28]. MIT can be calculated as [3]:

$$T_{\mathrm{MIT}} = \{(1 - P_{\mathrm{HOF}}) \times T_{\mathrm{HIT}}\} + \{P_{\mathrm{HOF}} \times T_{\mathrm{HOF}}\} \quad (1)$$

T_{MIT} = Total MIT

P_{HOF} = Probability of either a handover failure (HOF) or a radio link failure (RLF) during handover

T_{HIT} = Handover Interruption Time, MIT in a successful handover

T_{HOF} = Handover Failure Time, MIT in a HOF or RLF

In LTE Network, T_{HIT} is reported around 50 ms while T_{HOF} ranges from several hundred milliseconds to a few seconds [29]. Since T_{HOF} contributes a more significant portion in total MIT (T_{MIT}), reducing the T_{MIT} can be better done by reducing P_{HOF}, i.e. avoiding unnecessary handovers or handover to a wrong cell. Therefore, target cell determination is very important in the handover process to minimize MIT and ensure network connectivity.

In this paper, we performed simulations to test the reliability of the traditional handover algorithm. Based on our study, the traditional handover algorithm is reliable in ideal conditions, where the real condition of the neighbor cells can solely be determined by RSRP/RSRQ measurement, thus the target cell is always the best cell to continue the network connection. In a non-ideal condition, such as the presence of a cell coverage hole due to an obstacle, the RSRP/RSRQ measurements may not reflect the real network condition. A UE may be handed over to a target cell with the best RSRP/RSRQ, but it enters the target cell's coverage hole after the handover, and the connection fails. In this case, the traditional handover algorithm is not reliable to determine the target cell correctly and ensuring network connectivity.

Several methods have been explored to reduce MIT by reducing P_{HOF}, for example by implementing fast measurements [30], dual connectivity [31], conditional handover [5], [32], [33], and predictive handover [5], [19], [21], [22], [34], [35].

Fast measurement enables the UE to react faster to the channel changes and can improve mobility robustness, as the source base station sends a handover command before an abrupt deterioration of the radio link to the UE. However, it increases the battery consumption of the UE.

Dual connectivity allows UE to have two separate connections to different radio resources simultaneously, such as 4G and 5G. The packet duplication in Dual Connectivity may improve mobility robustness but increases the network complexity.

Conditional handover prepares in advance multiple candidate target cells in the network. This enables the handover

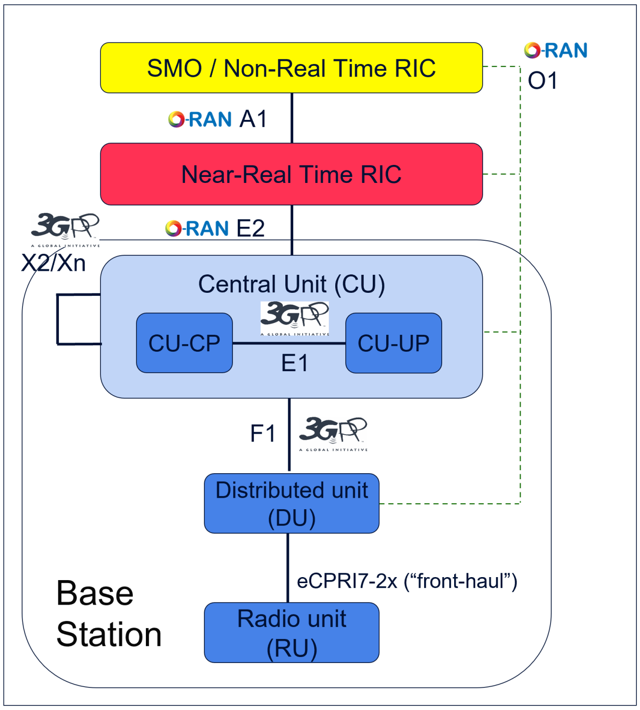


FIGURE 7. O-RAN Architecture [24].

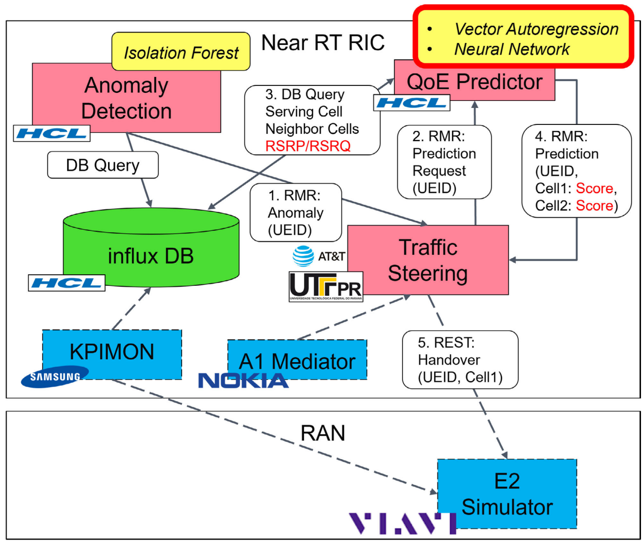


FIGURE 8. Anomaly detection use case of Near-RT RIC [45].

command to be sent to the UE earlier than at the traditional handover when the radio conditions are still good. In traditional handover, the handover command is sent when the radio conditions start to get degraded [33].

Predictive handover usually refers to an additional mechanism on top of conditional handover. In pure conditional handover, the candidate target cells are selected based on radio conditions, similar to traditional handover. In Predictive Handover, the candidate target cells may be predicted based on the user behavior and may employ machine learning techniques.

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed [36]. It studies the computer algorithms that improve automatically through experience [37].

IV. O-RAN NEAR-RT RIC

O-RAN is a relatively novel standard and currently introduces several applications for open and intelligent RAN. It standardizes new network elements called Radio Intelligent Controller (RIC) to add intelligence to the cellular radio network. There are two variants of RIC: Near-Real Time (Near-RT) and Non-Real Time (NRT) RIC (Fig.7).

As the name implies, Near-RT RIC is used to host applications that require immediate or real-time response, including mobility management applications like handover control. Due to this response requirement, Near-RT RIC must be implemented in an Edge Cloud, a virtual environment that is placed physically near the radio network.

NRT-RIC is used to host applications that do not require immediate response such as network monitoring and optimization. This network element can be implemented in Central Cloud and typically colocated with the existing network management system.

O-RAN Alliance already describes some use cases [38] to be implemented in O-RAN to provide RAN openness and intelligence, for example, Context-Based Dynamic HO Management for V2X, Flight Path-Based Dynamic UAV Radio Resources Allocation, QoE Optimization, and Traffic Steering. However, the exact implementation of the use case is given to specific vendors. For example, Nokia prioritizes Traffic Steering and Network Anomaly Detection use case for its RIC solution [39].

Several studies already use O-RAN RIC architecture for many applications such as connection management [40], mobility management [41], and scheduling policy optimization [42]. Various machine learning algorithms are implemented in RIC including reinforcement learning [43].

Near-RT RIC can be implemented in any virtualized environment. In our research, we installed it on an Ubuntu-based virtual machine by installing the open-source software provided by the O-RAN Software Consortium (SC) [44].

The Near-RT RIC use case example from O-RAN SC that is mostly corresponding to our research need is the Anomaly Detection use case [45]. However, we have to perform some modifications to fit our simulation scenario.

The Anomaly Detection Use Case consists of three applications (called xApp): Anomaly Detection, Traffic Steering, and Quality of Experience (QoE) Predictor. They work together by exchanging messages in RMR protocol, the Near-RT RIC internal communication. Currently, in this research, the Near-RT RIC works stand alone without any connection to the RAN, and all simulation data is stored in the database.

The use case begins with Anomaly Detection xApp detects an anomalous UE, for instance, the UE experiencing degradation of RSRP. In this research, this information is obtained from the database but in the real implementation, this information is notified by RAN (i.e. base station). The Anomaly Detection xApp then informs the anomaly to the Traffic Steering xApp.

Traffic Steering xApp then consults to QoE Predictor xApp by sending the identity of the UE experiencing an anomaly. QoE Predictor xApp predicts the score of QoE of the UE if the UE is placed in the neighboring cells. In the original software, this score is the throughput of the data transmission and is predicted using the vector autoregression (VAR) method. Therefore, QoE Predictor predicts the throughput experienced by the UE if it is placed in a certain cell.

This prediction is sent back to Traffic Steering xApp. Based on this prediction, it will perform some necessary actions. The action can be a handover command to the cell where the throughput prediction is the highest one.

In this research, we modified the original Near-RT RIC xApps in the Anomaly Detection Use Case to adapt to our simulation scenario. We mainly performed modifications in QoE Predictor xApp as it is the one that performs predictions that will determine the target cell. We performed two modifications to the original QoE Predictor xApp. The first modification is to adapt the original software to our simulation scenario. The prediction is still done by the vector autoregression method. The second modification is completely replacing the vector autoregression with a neural network. The neural network design is based on our previous studies [46], [47] that yield optimum results.

V. PROPOSED METHOD: MODIFIED VECTOR AUTOREGRESSION

In the original QoE Predictor xApp software provided by O-RAN SC, the prediction is done using the VAR method. However, the original software is not immediately usable for our research case. Our research aims to predict the target cell in a non-ideal network containing a coverage hole. This target cell is determined by the movement of the UE that is reflected in the RSRP/RSRQ measurements.

Vector Autoregression (VAR) is a statistical time series model used to analyze the relationship between multiple variables. In a VAR model, each variable in the system is modeled as a function of its past values and the past values of all the other variables in the system. A VAR model of order p, denoted as VAR(p), is a set of linear equations that relate each variable in the system to its own past values and the past values of all the other variables in the system up to p lags. The equations can be written in matrix form as:

$$Y_t = A_1 Y_{t-1} + A_2 Y_{t-2} + \ldots + A_p Y_{t-p} + u_t \qquad (2)$$

where Y_t is a k-dimensional vector of the current values of the k variables in the system, $A_1, A_2, \ldots, A_p$ are $k \times k$ matrices of coefficients that capture the dynamic relationships between the variables at lags 1 to p, and u_t is a k-dimensional vector of error terms that represent the unexplained part of the system at time t.

The unmodified original QoE Predictor xApp determines the target cell by predicting the throughput of each cell using time-series throughput data in the training data. However, this software only considers the position of the UE, i.e. what the neighbor cells are. It does not consider the movement and the trajectory angle of the UE. If we use the unmodified original software and training data, the prediction will always give the same target cell for all simulation cases.

Our proposed modified method using VAR can be expressed in the following pseudocode (Algorithm 1) and can be illustrated in (Fig. 9). The *italic* expression in Algorithm 1 indicates our modification.

Algorithm 1 Predict the Throughput of All Cells

Require: list of all cells serving and neighboring UE *and the RSRP measurements of those cells*
  for all cells in list **do**
    Query throughput of cell over time from Training Data *(where RSRP measurement is similar with the one reported by UE)*
    Remove outliers of the query result
    Predict the next throughput of the cell using Vector Autoregression
  end for
  Report the throughput prediction of all cells in list to Traffic Steering xApp

The QoE Predictor xApp works based on the request from the Traffic Steering xApp that sends the UE identity experiencing anomaly such as RSRP/RSRQ degradation. The QoE Predictor xApp software checks the neighbor cells of the UE in the current position and queries the training data to get the throughput trend of that cell. The training data used in the original software contains time series throughput data of all available cells in the network. The throughput trend of all cells is increasing but at a different rate (Fig. 10). This training data is suitable if the target cell is only determined by the UE's current position only. Using VAR, the QoE Predictor xApp can predict the throughput of those neighbor cells in the future and send this prediction to the Traffic Steering xApp. Traffic Steering xApp will then execute handover to the cell that has the highest predicted throughput.

To adapt the original software to our simulation scenario, we reconstructed the software and training data to put the UE movement into account. The UE movement and its trajectory angle can be reflected by the RSRP measurement variations. From the training data generation process described in Section VII, we construct the new training data that considers the UE movement to predict the next throughput by evaluating RSRP values. In this training data, the trend of throughput of each cell is not always the same but depends

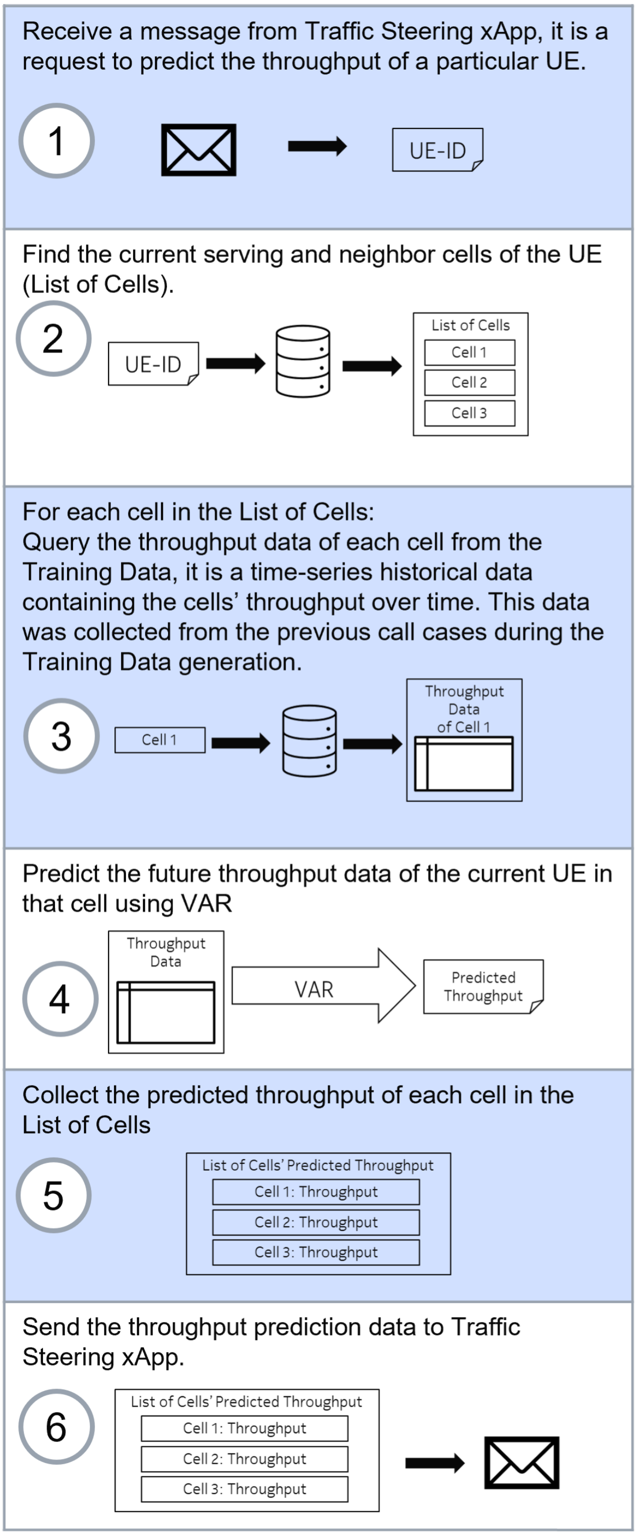


FIGURE 9. Workflow of the VAR-based QoE Predictor xApp.

on the RSRP difference of each cell. In our case, we only use two neighbor cells (Cell 2 and Cell 3) and we can use the RSRP difference between those two neighbor cells to reflect the UE movement (Fig. 11).

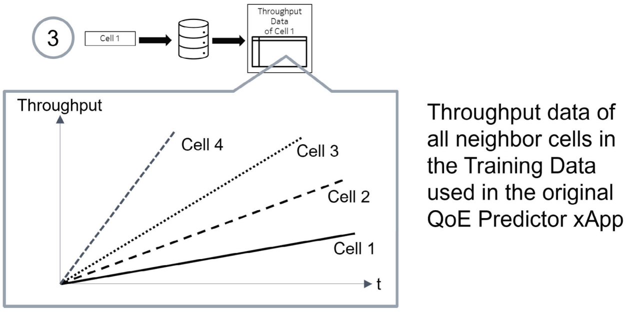


FIGURE 10. Training data in the unmodified QoE predictor xApp.

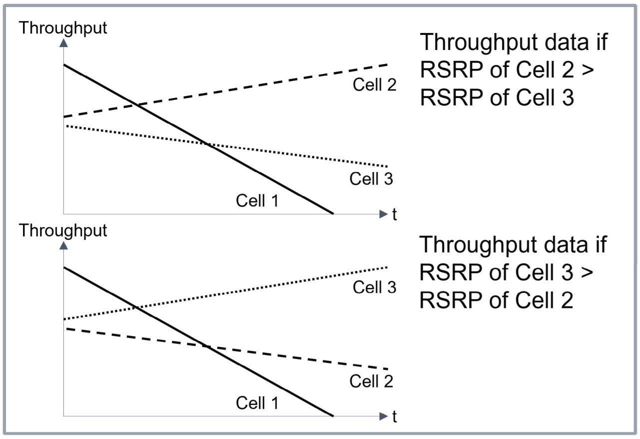


FIGURE 11. Training data in the modified QoE predictor xApp.

Using this modified training data, the target cell is determined by the UE movement, not only the UE position like the original QoE Predictor xApp.

VI. PROPOSED METHOD: NEURAL NETWORK

For our second proposed method, we completely replaced the VAR in the QoE Predictor xApp with a neural network. In this preliminary stage, we use a very simple neural network regression model to predict whether the download is successful or failed using RSRP and RSRQ samples as input. In the current model, even though the input is time-series data, our neural network does not treat it as time-series data but collects a few samples and puts them as the model input at once.

We designed a neural network containing 18 input nodes, 4 hidden nodes, and 1 output node (Fig.12). The inputs are the last 3 samples of RSRP and RSRQ measurements from all the 3 cells. The output is whether the download is successful or not, represented by the number 0 (failed download) or 1 (successful download). When the neural network is used, the output node is actually a floating point continuous number between 0 and 1 that can be used as the prediction score. The cell with the highest score will be the target cell. As a result, we do not need throughput data to determine the target cell in this modification.

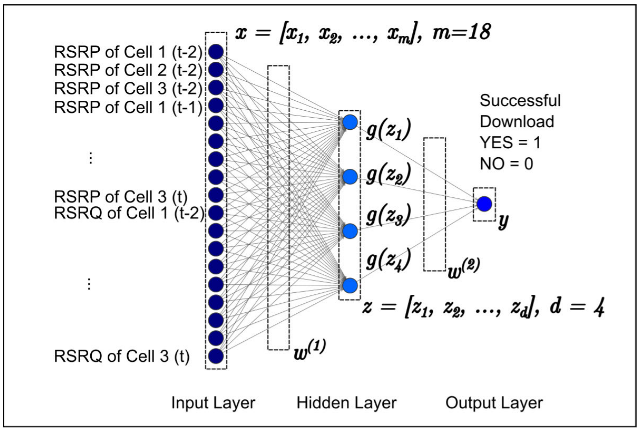


FIGURE 12. Neural network design.

The neural network design can be represented mathematically as follows. The input is represented as x, a vector of 18 variables ($m = 18$) representing RSRP and RSRQ measurements. The hidden nodes are represented as z, in this case, a vector of 4 variables ($d = 4$). The output is y which value is between 0 (representing failed download) and 1 (representing successful download). The matrix $w^{(1)}$ is representing the weight of the connection between input and hidden nodes, and matrix $w^{(2)}$ is for the connection between hidden nodes and output.

The value of hidden nodes is defined as:

$$z_i = w^{(1)}_{0,i} + \sum_{j=1}^{m} x_j w^{(1)}_{j,i}, m = 18 \tag{3}$$

While the value of the output is defined as:

$$y = g\left(w^{(2)}_{0,i} + \sum_{j=1}^{d} g(z_j) w^{(2)}_{j,i}\right), d = 4 \tag{4}$$

Here g is the activation function to introduce non-linearity in the neural network. In our case, we use the sigmoid function:

$$g(z) = \frac{1}{1 + e^{-z}} \tag{5}$$

During the training process, the z, $w^{(1)}$, and $w^{(2)}$ values are generated from the known x and y in the training data. The obtained values then are saved in the neural network model to be used to predict the output y from the input x.

The neural network implementation is using Tensor Flow Keras API. In the current implementation, the training process is done with 150 times iterations through the whole training data (*epoch = 150*), and the model is updated every 10 training data (*batch size = 10*).

VII. SIMULATION DESIGN AND DATA COLLECTION

In this research, we created an environment containing three cells, one moving UE, and a building creating a coverage hole (Fig.13). This environment is built using NS3 LTE network simulator [26] based on previous studies [18], [48]. The simulation parameters are reusing the previous work as described in Table 1.

TABLE 1. NS3 simulation parameters.

Parameter	Value
System bandwidth	5 MHz
Inter-site distance	500 m
Adaptive Modulation and Coding Scheme	MiErrorModel
Simulation area	2000×2000 m^2
Number of base stations	3
Transmit Power	46 dBm
Number of UEs	3
Velocity of UE1	16.6667 m/s
Path Loss Model	Cost 231
Antenna Height	30 m
Obstacle Height	35 m
File Size	15 MB

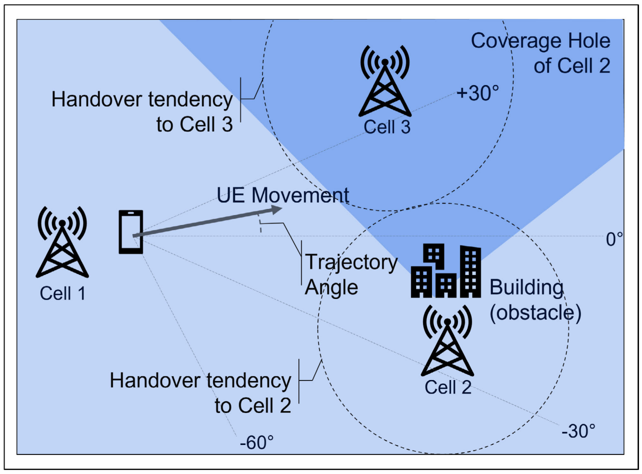


FIGURE 13. Environment for simulation.

On each simulation, the UE moves to the right side of the network with a random trajectory angle. Due to this movement, the UE needs to perform a handover from Cell 1 to either Cell 2 or Cell 3, depending on the trajectory angle. The UE also downloads files during the movement and in the end, the download may be successful or may not. For every simulation, we noted down the target cell, the download success status, and the RSRP/RSRQ measured by UE.

To create the training data, we ran 200 simulations. The first 100 simulations are deterministic handover cases where the UE is forced to perform handover to Cell 2 regardless of the trajectory angle. The next 100 simulations are also deterministic handover cases but this time to Cell 3. The flow chart of this training data generation is described in Fig. 14. In this flow chart, the amount of target cells is 2 (Cell 2 and Cell 3), thus *n=2*. We ran 100 simulation runs for each cell, thus *m=100*. The RSRP, RSRQ, and download success status of each simulation run are then saved as Training Data.

To compare the performance of the handover algorithms, we ran another 1000 simulations of non-deterministic handover. The flow chart can be seen in Fig. 15, and

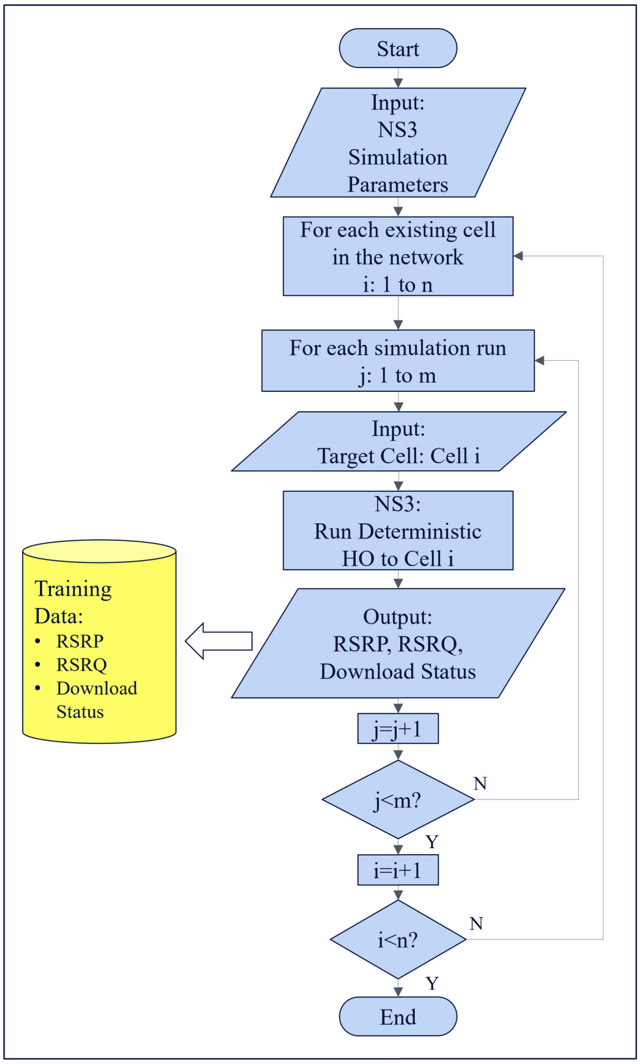


FIGURE 14. **Flow chart: Training data generation.**

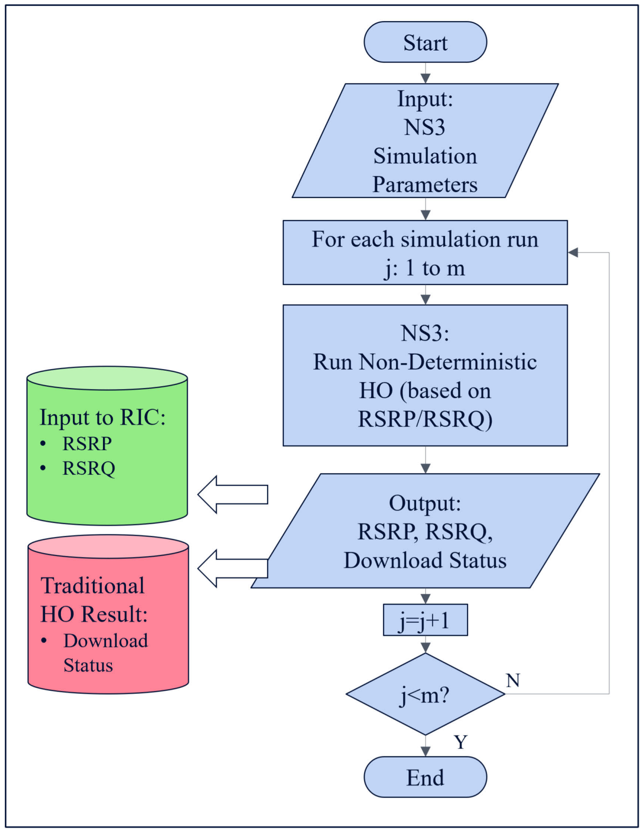


FIGURE 15. **Flow chart: Traditional handover simulation.**

in this case *m=1000*. In these simulations, the UE may perform a handover to either Cell 2 or Cell 3, using a traditional handover algorithm, based on the RSRP/RSRQ measurements. The result of these traditional handover simulations (download success status and RSRP/RSRQ measurement) is used as a baseline to be compared with machine-learning-based algorithms run in Near-RT RIC. The RSRP/RSRQ measurement for these simulations is also used as input for RIC-based handover algorithm.

The flow chart in Fig. 16 describes the RIC-based handover simulation. First, we train the model using the result of the previous process (training data generation, Fig. 14). For each simulation run, we performed prediction using the machine-learning-based algorithm in Near-RT RIC by providing RSRP and RSRQ measurement of the same simulations that we ran in the traditional handover algorithm process (Fig. 15). The algorithm in Near-RT RIC would then get the score of each existing neighbor cell in the network. The cell with the highest score is then chosen as the target cell. This process is the same for both our proposed method (modified VAR and NN).

From Near-RT RIC we only obtained the target cell, but not yet the download success status. Therefore, we need to perform verification using NS3 to check if the download is successful or not, given the target cell from Near-RT RIC (Fig. 17). Next, we performed deterministic handover again using NS3 but using the target cell obtained by Near-RT RIC. From here we get the download success status if the handover is controlled by Near-RT RIC.

We choose download success rate as the main performance metric. As stated in Equation 1 in Section III, the handover process is best improved by reducing the probability of handover failure, thus avoiding unnecessary handover and handover to a wrong cell. Based on this statement, we focus on the target cell determination process. We decide the performance metric as download success rate if we use a certain method to select the target cell: Traditional Algorithm, Near-RT RIC using Modified VAR, and Near-RT RIC using neural network.

VIII. SIMULATION RESULT AND DISCUSSION

As described in Section VII, we performed three sets of simulations: the traditional handover, Near-RT RIC handover using VAR (first modification), and Near-RT RIC using neural network (second modification). The traditional

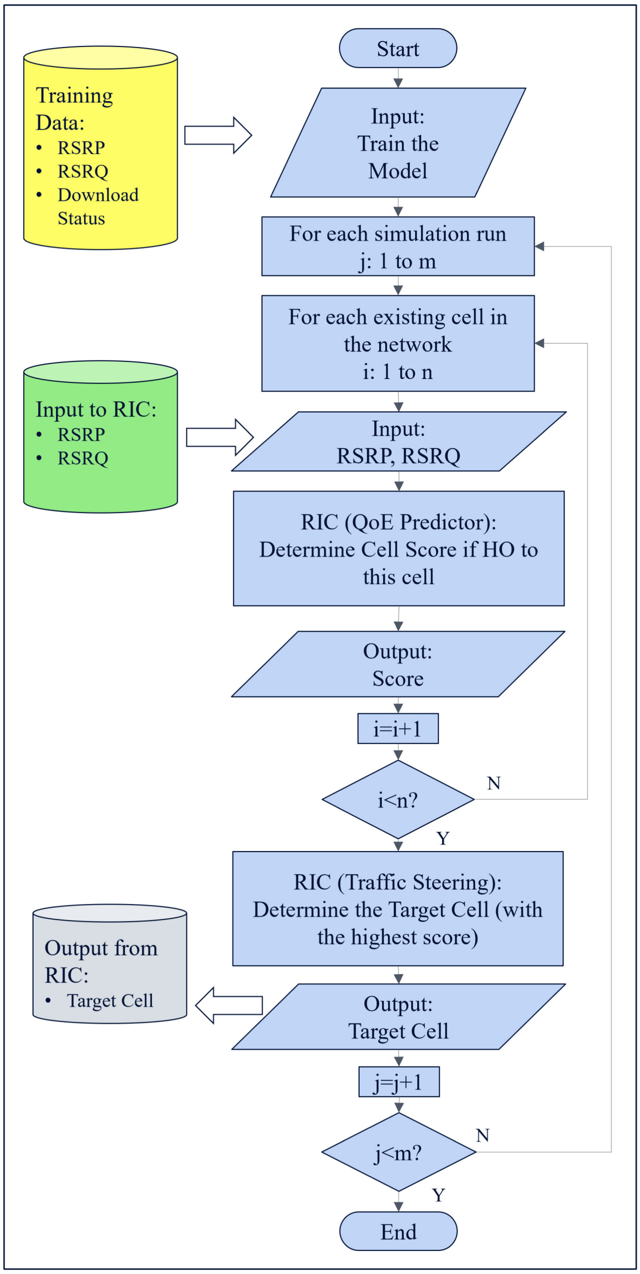


FIGURE 16. Flow Chart: RIC handover simulation.

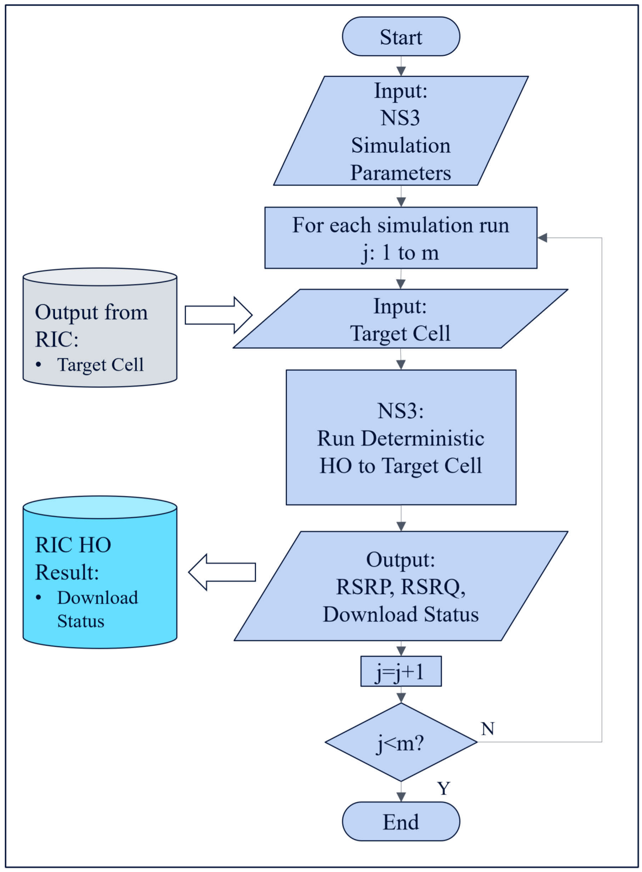


FIGURE 17. Flow chart: RIC target cell testing.

handover was done 1000 times using the NS3 simulator and we noted down the RSRP/RSRQ measurement and the handover results (target cell and download success status). The RSRP/RSRQ measurement of those simulations was used as input in Near-RT RIC handover simulations. After that, we compared the download success rate of all simulations among the three methods (Fig.18).

The successful download rate for the traditional handover algorithm is 86.2%, not 100% due to the presence of the coverage hole. All of the simulations with failed downloads happen when the UE was handed over to Cell 2 (based on the best RSRP/RSRQ measurement) but it entered the coverage hole behind the building after the handover. If it was handed over to Cell 3 instead of Cell 2, the download may be successful because Cell 3 was not obstructed by the building. This result shows that sometimes the traditional handover algorithm is not reliable in a non-ideal condition.

When we determined the target cell using Near-RT RIC, the successful download rate is mostly increasing, depending on the method and the amount of training data. If the QoE Predictor xApp uses vector autoregression (VAR), modified in our first modification, the success rate can reach 95.3% using all 100 available training data, 94.1% with 50 training data, and 92.7% with only 25 training data. If we use NN in our second modification, the success rate is slightly lower but still higher than the traditional algorithm in most cases. Using NN, the download success rate can reach 91.9 % using all 100 available training data and 88.4% using only 50 training data. However, the performance plummeted to only 58.8% if we only use 25 training data (even lower than the traditional handover algorithm).

As mentioned earlier, we intend to implement this machine-learning-based solution for ultra-dense small-cell environments to support high-speed users. Therefore we also test this method for two UE speed variation. If the original simulation is simulating the UE speed of 16.6667 m/s, i.e. 60 km/hour a typical car speed, we add the simulation with

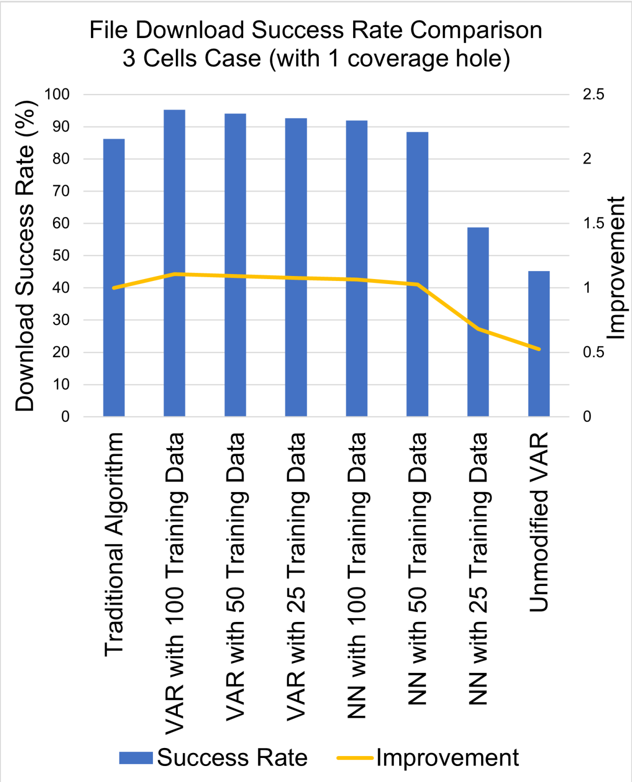


FIGURE 18. **Simulation result comparison.**

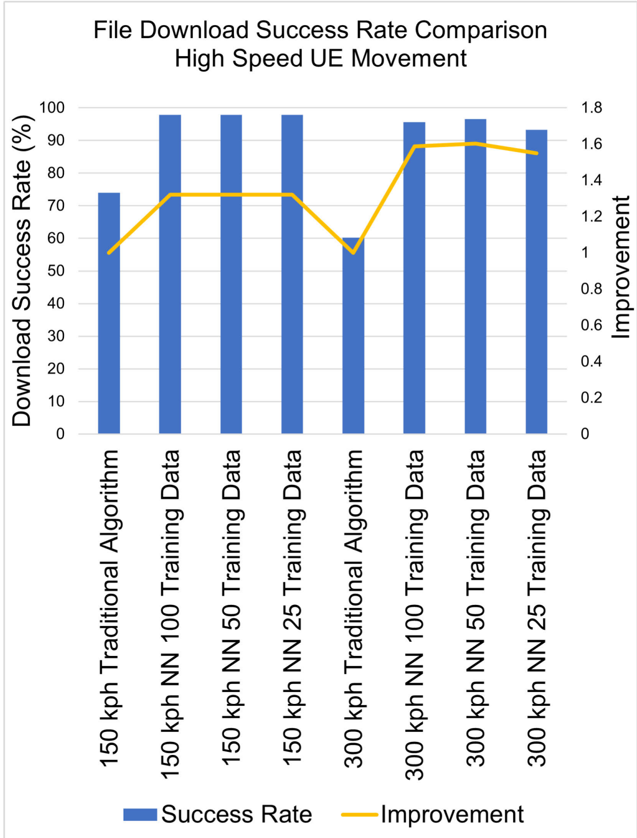


FIGURE 19. **Simulation result comparison for high speed cases.**

150 km/hour (user speed in a typical train) and 300 km/hour (a user in a high-speed train). We performed simulations using our proposed neural network method and the result can be seen in Fig. 19. As can be seen in the graph, the neural network method can improve significantly to the download success rate, even with a small amount of training data.

It is shown that machine-learning algorithms can provide better handover performance by determining the correct target cell in a non-ideal network condition. However, this performance is determined by the amount of training data, the more training data, the better the performance. From the simulation result, the first proposed method using modified VAR performs better than the neural network method.

Besides the successful download rate, we also compare the processing time of the two proposed methods in QoE Predictor xApp. This is the time required by the QoE Predictor xApp to predict the performance or score of each neighbor cell and thus determines the target cell. The time is calculated from the moment the QoE Predictor xApp receives the request from Traffic Steering xApp until it gives the result. This processing time varies among simulation cases and we took the average time from several cases.

From Fig. 20 can be seen that the processing time of QoE Predictor xApp, when we use the VAR and neural network method, varies slightly. The neural network still performs slightly slower than the VAR but it is still comparable. The time required to predict the QoE using the proposed neural network is still reasonable and similar to the VAR. Some optimizations are still required to improve the speed of our neural network.

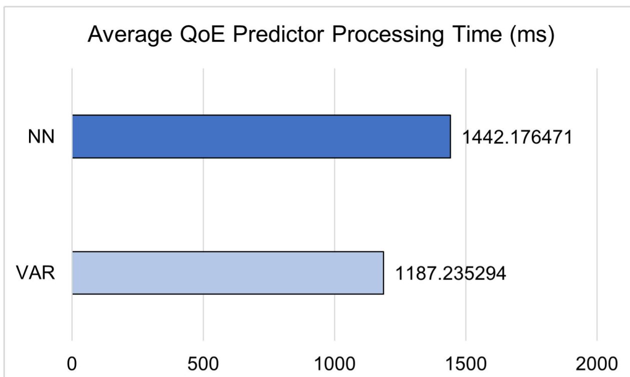


FIGURE 20. **QoE predictor processing time comparison.**

Our neural network method may perform less than the VAR method but it has some advantages in the simplicity issues. The VAR method determines the target cell based on the throughput data on each cell. This will require intensive measurements during training data generation. Our neural network method will determine the target cell based only on the RSRP/RSRQ measurement, the metrics that are already available by default in the RAN system. The advantage of

TABLE 2. Comparison of the original Near-RT RIC method (VAR) and the proposed method (NN).

	VAR	NN
Training Data Components	RSRP/RSRQ, Time series throughput	RSRP/RSRQ, Download status
Training Data Usage	Queried from the database on each simulation	Only during the NN model creation (first simulation)
Prediction Method	Query the time series throughput data (filtered with RSRP/RSRQ) from the Training Data and predict the next throughput for each cell	Provide RSRP/RSRQ as input to the NN model and obtain successful download probability (score) from each cell

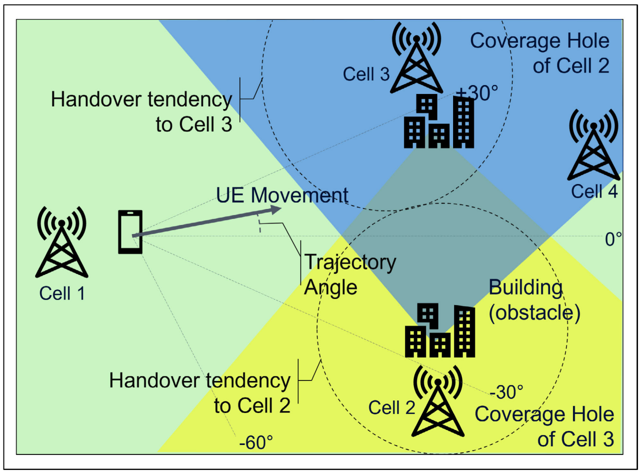


FIGURE 21. Simulation design with 2 coverage holes.

RSRP/RSRQ measurement is it can be measured by the UE without connecting to the cell, unlike the throughput measurement that requires the UE to connect to the concerned cell.

The neural network method does not require throughput measurement to generate the training data. From this RSRP/RSRQ measurement, the neural network will determine the score of each cell which also means the probability of a successful download. Another advantage of using a neural network is that the model can be saved to be used later. This will simplify and speed up the prediction process. In the VAR method, the software needs to query the time-series throughput information from training data for each case in order to predict the next throughput (Table 2).

IX. EXTENDED SIMULATION: NETWORK WITH 4 CELLS AND 2 COVERAGE HOLES

In addition to the current simulation, we performed another simulation to further prove the usability and effectiveness of Near-RT RIC over the traditional handover algorithm. Here we cr eated a new network environment with 4 cells and two buildings creating two coverage holes (Fig. 21).

The methods used in this simulation are the same, the modified VAR and the neural network. Obviously, the models must be adapted to the current network model. The neural network input is modified to 24 nodes, but still uses 4 hidden nodes and one output node.

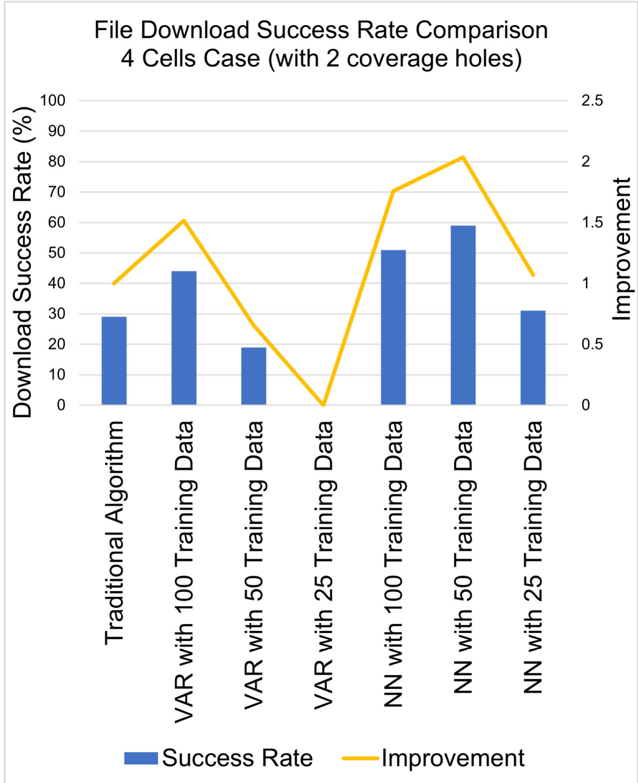


FIGURE 22. Simulation result with 2 coverage holes.

Fig. 22 shows the result of this simulation scenario. It can be seen in this more complex non-ideal network, the traditional handover algorithm performance is very low. The download success rate using the traditional handover algorithm is only about 29%.

Using Near-RT RIC can bring improvement to the handover performance, but due to the complexity of the network, the download success rate is relatively low compared to the previous one-coverage-hole scenario. However, it still brings improvement over the traditional handover algorithm. The modified VAR method can only bring improvement only if using 100 training data. If the training data is 50 or less, it does not improve the handover performance. The NN method is proven more robust in improving handover performance. Interestingly, 100 and 50 training data using NN do not give significant difference and sometimes 50 training data has better performance.

X. CONCLUSION AND FUTURE WORK

The usage of the higher frequency in the newer cellular technology will cause smaller cell size and this will further cause frequent handover in a high mobility user like the ones in the high-speed train. The handover process may cause an interruption in the data transmission, moreover in a high-mobility condition where the radio condition may worsen because of the user speed. Increasing the probability of successful handovers, such as making sure to perform handover to the correct target cell, can minimize this interruption. Therefore, target cell determination is very important in the handover process.

In this paper, we presented the result of our two proposed methods, the modified VAR and neural network, using O-RAN Near-RT RIC to determine the target cell in the handover process. From the simulation result, it can be concluded that machine-learning-based algorithms as in our two proposed methods can be used and are proven better to determine the target cell compared to the traditional handover algorithm. The performance of the algorithms depends on the method and the amount of training data.

The proposed neural network may currently underperform the VAR method in Near-RT RIC software but it has a simpler implementation since it uses only RSRP/RSRQ measurement without a throughput measurement. This neural network method can be faster as the model can be saved and reused without necessarily querying the training data on each prediction. However, some optimizations are still required to speed up the process. In a more complex scenario like the presence of two coverage holes, the neural network method is proven more robust than VAR.

In the future, we will improve the neural network to get better performance. We plan to test this method in another non-ideal network environment other than the coverage hole case.

REFERENCES

[1] *NR; User Equipment (UE) Radio Transmission and Reception; Part 2: Range 2 Standalone-Release 16*, document TS 38.101-2, V16.2.0, 3GPP, 2020.

[2] *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Overall description Stage 2 (Release 10)*, document TS 36.300, V16.0.0, 3GPP, 2013.

[3] H. Park, Y. Lee, T. Kim, B. Kim, and J. Lee, "Handover mechanism in NR for ultra-reliable low-latency communications," *IEEE Netw.*, vol. 32, no. 2, pp. 41–47, Mar. 2018.

[4] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2392–2431, 4th Quart., 2017.

[5] C. Lee, H. Cho, S. Song, and J. Chung, "Prediction-based conditional handover for 5G mm-wave networks: A deep-learning approach," *IEEE Veh. Technol. Mag.*, vol. 15, no. 1, pp. 54–62, Mar. 2020.

[6] Z. Ali, N. Baldo, J. Mangues-Bafalluy, and L. Giupponi, "Machine learning based handover management for improved QoE in LTE," in *Proc. IEEE/IFIP Netw. Operations Manag. Symp.*, Apr. 2016, pp. 794–798.

[7] N. Sinclair, D. Harle, I. A. Glover, J. Irvine, and R. C. Atkinson, "An advanced SOM algorithm applied to handover management within LTE," *IEEE Trans. Veh. Technol.*, vol. 62, no. 5, pp. 1883–1894, Jun. 2013.

[8] N. M. Alotaibi and S. S. Alwakeel, "A neural network based handover management strategy for heterogeneous networks," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2015, pp. 1210–1214.

[9] M. Rihani, M. Mroue, J. Prevotct, F. Nouvel, and Y. Mohanna, "A neural network based handover for multi-RAT heterogeneous networks with learning agent," in *Proc. 13th Int. Symp. Reconfigurable Commun.-Centric Syst.-Chip (ReCoSoC)*, Jul. 2018, pp. 1–6.

[10] B. Shubyn, N. Lutsiv, O. Syrotynskyi, and R. Kolodii, "Deep learning based adaptive handover optimization for ultra-dense 5G mobile networks," in *Proc. IEEE 15th Int. Conf. Adv. Trends Radioelectronics, Telecommun. Comput. Eng. (TCSET)*, Feb. 2020, pp. 869–872.

[11] G. Masini, Y. Gao, and S. Sirotkin. (2021). *Artificial Intelligence and Machine Learning in NG-RAN: New Study in RAN3*. [Online]. Available: https://www.3gpp.org/news-events/2201-ai_ml_r3

[12] N. Abramson, D. Braverman, and G. Sebestyen, *Pattern Recognition and Machine Learning*, vol. 9, no. 4. Berlin, Germany: Springer, 1963.

[13] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2010.

[14] G. Hinton and T. Sejnowski, *Unsupervised Learning: Foundations of Neural Computation*. Cambridge, MA, USA: MIT Press, 1999.

[15] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, no. 9, pp. 237–285, May 1996.

[16] X. Chen, F. Mériaux, and S. Valentin, "Predicting a user's next cell with supervised learning based on channel states," in *Proc. IEEE 14th Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2013, pp. 36–40.

[17] L. Yan, H. Ding, L. Zhang, J. Liu, X. Fang, Y. Fang, M. Xiao, and X. Huang, "Machine learning-based handovers for sub-6 GHz and mmWave integrated vehicular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4873–4885, Oct. 2019.

[18] T. C. D. B. Guerra, "Machine learning based handover management for LTE networks with coverage holes," Ph.D. dissertation, Centro de Tecnologia, Univ. Federal do Rio Grande do Norte, Natal, Brazil, 2018.

[19] S. Kumar, S. Vanmathi, B. Sanjay, S. Bharathi, and M. Meena, "Handover forecasting in 5G using machine learning," *Int. J. Eng. Technol.*, vol. 7, pp. 76–79, Jan. 2018.

[20] L. L. Vy, L. Tung, and B. P. Lin, "Big data and machine learning driven handover management and forecasting," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Sep. 2017, pp. 214–219.

[21] C. Wang, L. Ma, R. Li, T. S. Durrani, and H. Zhang, "Exploring trajectory prediction through machine learning methods," *IEEE Access*, vol. 7, pp. 101441–101452, 2019.

[22] Y. Koda, K. Yamamoto, T. Nishio, and M. Morikura, "Reinforcement learning based predictive handover for pedestrian-aware mmWave networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2018, pp. 692–697.

[23] V. Yajnanarayana, H. Rydén, and L. Hévizi, "5G handover using reinforcement learning," in *Proc. IEEE 3rd 5G World Forum (5GWF)*, Sep. 2020, pp. 349–354.

[24] O-RAN. (Oct. 2018). *Towards an Open and Smart RAN. White Paper*. [Online]. Available: https://www.o-ran.org/s/O-RAN-WP-FInal-181017.pdf

[25] *System Architecture for the 5G System*, document TS 23.501, V16.0.0, 3GPP, Mar. 2019.

[26] NS3. *Design Documentation—Model Library*. Accessed: Jan. 5, 2023. [Online]. Available: https://www.nsnam.org/docs/models/html/lte-design.html#handover

[27] J. Agrawal, P. Mor, J. M. Keller, R. Patel, and P. Dubey, "Introduction to the basic LTE handover procedures," in *Proc. Int. Conf. Commun. Netw. (ICCN)*, Nov. 2015, pp. 197–201.

[28] *Study on Scenarios and Requirements for Next Generation Access Technologies (Release 16)*, document TR 38.913, 3GPP, 2020.

[29] *Evolved Universal Terrestrial Radio Access (E-UTRA) Study on Latency Reduction Techniques for LTE (Release 14)*, document TR 36.881, 3GPP, 2016.

[30] A. Chincholi, M. Menon, and L. Hsu, "Measurement gap enhancements for BL/CE UEs," WO Patent 2019/094 977 Al, Qualcomm Incorporated, San Diego, CA, USA, 2019.

[31] M. Polese, M. Giordani, M. Mezzavilla, S. Rangan, and M. Zorzi, "Improved handover through dual connectivity in 5G mmWave mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2069–2084, Sep. 2017.

[32] H. Martikainen, I. Viering, A. Lobinger, and T. Jokela, "On the basics of conditional handover for 5G mobility," in *Proc. IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2018, pp. 1–7.

[33] I. L. Da Silva, C. Eklof, J. Muller, and R. Zhohov. *This is the Key to Mobility Robustness in 5G Networks*. Accessed: Jan. 5, 2023. [Online]. Available: https://www.ericsson.com/en/blog/2020/5/the-key-to-mobility-robustness-5g-networks

[34] R. V. Akhpashev and V. G. Drozdova, "Spatial interpolation of LTE measurements for minimization of drive tests," in *Proc. 19th Int. Conf. Young Specialists Micro/Nanotechnologies Electron Devices (EDM)*, Jun. 2018, pp. 6403–6405.

[35] A. Masri, T. Veijalainen, H. Martikainen, S. Mwanje, J. Ali-Tolppa, and M. Kajo, "Machine-learning-based predictive handover," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag.*, May 2021, pp. 648–652.

[36] (2020). *What is Machine Learning? A Definition—Expert System | Expert.Ai*. [Online]. Available: https://www.expert.ai/blog/machine-learning-definition/

[37] T. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.

[38] A. Akman, "O-RAN working group 1 use cases detailed specification," V04.00, O-RAN ALLIANCE e.V., Alfter, Germany, Tech. Rep., 2020.

[39] *Nokia Launches First Commercial Service Enablement Platform to Drive Open RAN Innovation*. Accessed: Jan. 5, 2023. [Online]. Available: https://www.nokia.com/about-us/news/releases/2021/03/11/nokia-launches-first-commercial-service-enablement-platform-to-drive-open-ran-innovation/

[40] O. Orhan, V. N. Swamy, T. Tetzlaff, M. Nassar, H. Nikopour, and S. Talwar, "Connection management xAPP for O-RAN RIC: A graph neural network and reinforcement learning approach," in *Proc. 20th IEEE Int. Conf. Mach. Learn. Appl.*, Dec. 2022, pp. 936–941.

[41] Y. Li, E. Datta, J. Ding, N. B. Shroff, and X. Liu, "Can online learning increase the reliability of extreme mobility management?" in *Proc. IEEE/ACM 29th Int. Symp. Quality Service (IWQOS)*, Jun. 2021, pp. 1–6.

[42] L. Bonati, S. D'Oro, M. Polese, S. Basagni, and T. Melodia, "Intelligence and learning in O-RAN for data-driven NextG cellular networks," *IEEE Commun. Mag.*, vol. 59, no. 10, pp. 21–27, Oct. 2021.

[43] H. Lee, Y. Jang, J. Song, and H. Yeon, "O-RAN AI/ML workflow implementation of personalized network optimization via reinforcement learning," in *Proc. IEEE Globecom Workshops*, Dec. 2022, pp. 1–6.

[44] F. Cefalu. (2020). *O-RAN SC Wiki: Getting Started*. [Online]. Available: https://wiki.o-ran-sc.org/display/GS/Getting+Started

[45] D. Karnwal. (2021). *Anomaly Detection Use Case*. [Online]. Available: https://wiki.o-ran-sc.org/display/RICP/Anomaly+Detection+Use+Case

[46] B. H. Prananto and A. Kurniawan, "Study on neural network for cellular mobility management: NS3 simulation with coverage hole case," in *Proc. 15th Int. Conf. Telecommun. Syst., Services, Appl. (TSSA)*, Nov. 2021, pp. 1–6.

[47] B. H. Prananto and A. Kurniawan, "O-RAN intelligent application for cellular mobility management," in *Proc. Int. Conf. ICT Smart Soc. (ICISS)*, Aug. 2022, pp. 01–06.

[48] Z. Ali, N. Baldo, J. Mangues-Bafalluy, and L. Giupponi, "Simulating LTE mobility management in presence of coverage holes with ns-3," in *Proc. 8th EAI Int. Conf. Simul. Tools Techn.*, 2015, p. e2.

BAUD HARYO PRANANTO (Student Member, IEEE) received the B.S. degree from the Bandung Institute of Technology, Indonesia, in 2004, and the M.S. degree from the Korea Institute of Science and Technology, South Korea, in 2008. He is currently pursuing the Ph.D. degree with the Bandung Institute of Technology.

He has been a Specialist Trainer with Nokia Solutions and Networks, since 2008, delivering training related to 4G and 5G RAN Nokia equipment.

ISKANDAR received the B.S. and M.S. degrees from the Bandung Institute of Technology, Indonesia, in 1995 and 2000, respectively, and the Doctoral degree from Waseda University, Tokyo, Japan, in 2007.

He is currently an Associate Professor with the School of Electrical Engineering and Informatics, Bandung Institute of Technology. His research interest includes wireless telecommunications.

ADIT KURNIAWAN received the B.S. degree from the Bandung Institute of Technology, Indonesia, in 1986, the M.S. degree from the Royal Melbourne Institute of Technology, Australia, in 1996, and the Ph.D. degree from the University of South Australia, in 2003.

He is currently a Professor with the School of Electrical Engineering and Informatics, Bandung Institute of Technology. His research interests include antenna propagation, microwave propagation, wireless spread spectrum, and CDMA.