Random Forest Prediction of WLAN Throughput using Communication Logs and Channel Occupancy Rate

Nan Ni* and Takeo Fujii*
* Advanced Wireless and Communication research Center (AWCC), The University of Electro-Communications
1–5–1 Chofugaoka, Chofu, Tokyo 182–8585, Japan
Emails: {ni, fujii}@awcc.uec.ac.jp

***Abstract*—With the advent of high-speed and low-latency wireless communications such as 5G and Wi-Fi 6, more and more users are using them for large traffic applications such as high-resolution video streaming and cross reality (XR) services. However, the quality of wireless communications can be degraded by interference and other factors, which can reduce the quality of such services. Prediction of wireless communication quality is a useful method to deal with this problem. A method has been proposed to avoid degradation of quality of service (QoS) by predicting throughput using time-series data of throughput, which is one of the indicators of wireless communication quality. However, existing methods require constant measurement of throughput, which increases the traffic load and may degrade the quality of service. In this paper, we propose a method for predicting wireless communication quality without traffic load for measuring throughput by using communication logs. The proposed method uses received signal strength indicator (RSSI), channel occupancy rate (COR) and modulation and coding scheme (MCS) as inputs. The learning is performed using a random forest to predict the throughput that can be potentially transmitted from a node with new connection. Through experimental evaluation, we have confirmed that we can predict the potential throughput with an average error of about 16% in areas where RSSI is greater than −60 dBm by using communication logs and COR.**

***Index Terms*—IEEE 802.11ac, Random Forest, Throughput Estimation, Channel Occupancy Rate, MCS**

I. Introduction

In recent years, the fifth-generation mobile communication system (5G) and a new wireless LAN (WLAN) standard named IEEE 802.11ax (Wi-Fi 6) have enabled faster link speed and lower latency wireless communications than before. As a result, cross reality (XR) services such as virtual reality (VR) and high-resolution video streaming and viewing services have grown and developed significantly. Today, these services can be used in various locations by using wireless communications such as cellular systems and WLAN for the last hop to the user, which is the communication section from the base station, router, or access point (AP) to the end-user. Here, it is necessary to provide high-speed and low-latency communications for the comfortable use of high-resolution video distribution and viewing services and XR services. Specifically, the recommended transmission speeds for video streaming services such as YouTube are 20 Mbps or higher for 4K videos and 50 Mbps or higher for 8K VR videos [1]. Also, it is recommended that the latency becomes less than 200 ms for comfortable video calls [2].

However, the increase of wireless users at a particular location and time may cause bandwidth shortages and radio interference. Consequently, the quality of wireless communication degrades, making it impossible to maintain high-speed, low-latency communication. Then, the quality of services that require high-speed, low-latency communication will get worse. Poor quality of service (QoS), for example, affects the stable provision of live streaming services that require real-time performance and remote surgery services that require high reliability. Therefore, service suppliers need to consider how to avoid QoS degradation. If they can predict the user's wireless communication quality in advance, they can take measures such as changing the used AP or switching the communication system before QoS degradation occurs. Thus, predicting wireless communication quality can help avoid QoS degradation.

One method for predicting wireless communication quality is to create a prediction model of wireless communication quality by using machine learning to analyze a time-series dataset created based on measured data of wireless communication quality evaluation indicators. Here, the important indicators for evaluating wireless communication quality are throughput, round trip time (RTT), link speed, and packet loss rate. An existing method predicts future values of these indicators by learning time-series data of the indicators collected from each user using a graph convolutional network (GCN) [3].

Other methods have been proposed using channel occupancy rate (COR) and received signal strength indicator (RSSI) as evaluation indicators in addition to throughput and predicting future throughput using long and short term memory (LSTM) [4].

One common point among these methods is that they include past throughput as input for machine learning in order to predict throughput. However, using past throughput makes it difficult to predict throughput before data transmission begins. Furthermore, constant measurement of throughput to predict wireless communication quality may lead to an increase in traffic load and resulting degradation of communication quality due to large traffic required for measurement throughput.

Therefore, for stable provision of services that require high-speed and low-latency communication, it is necessary to develop a method for predicting wireless communication quality without extra traffic. In this paper, we propose a method for predicting wireless communication quality indexes using communication logs that are easy to obtain without additional traffic. Specifically, we take RSSI, COR, and modulation and coding scheme (MCS) as communication logs and learn using random forest to predict the potential throughput Th_p of a new connected terminal. The potential throughput is defined as the throughput that can be newly generated by the terminal. In this paper, we actually created a model that predicts potential throughput from RSSI, COR and MCS values using a random forest, and evaluated its prediction accuracy.

The structure of this paper is as follows. Section II describes the communication logs used for the prediction. Section III describes our proposed method for predicting potential throughput using communication logs. Section IV presents the results of experimental evaluation based on the proposed method, and Section V concludes the paper.

II. Communication Logs used for Prediction

A. Received Signal Strength Indicator

Received signal strength indicator (RSSI) is a value that expresses the power of the radio signal received in the wireless communication terminal. The larger the value, the higher the input voltage, and reception is more stable. The RSSI value is decreased with increasing distance between the transmitter and receiver and also decreased due to the presence of obstacles.

In 2.4 GHz band 802.11n, it has been found by measurement that there is a linear relationship between RSSI and throughput [5]. Considering that the 5 GHz band 802.11ac is basically an extension of 802.11n, a linear relationship between RSSI and throughput is expected for the 5 GHz band 802.11ac as well when the bandwidths are the same. In fact, measurement experiments have confirmed a roughly linear relationship [6].

B. Channel Occupancy Rate

IEEE 802.11 uses carrier sense multiple access/collision avoidance (CSMA/CA) to avoid frame collisions in the channel. In CSMA/CA, before transmitting data, all WLAN equipment checks whether there are any other devices in communication, i.e., whether the channel is in use by carrier sense. If there is a device in communication, in other words, the channel is in use (busy), transmission is postponed and if there is no device in communication, in other words, the channel is not in use (idle), the device waits for a certain period, called DCF Inter Frame Space (DIFS), and a random back-off period and then transmits the data after confirming again that the channel is idle.

The channel occupancy rate (COR) is defined as Eq.(1):

$$COR := \frac{t_{busy}}{t_{active}}, \tag{1}$$

where t_{active} is the active time since the move to a particular channel, i.e., the time elapsed after the move, and t_{busy} is the time during which the channel was busy. Regarding Eq.(1), the longer the other WLAN devices are communicating, the larger the value of t_{busy} and the larger the value of COR. It can also be said that the throughput of own device decreases when the COR is high because the device cannot communicate while other WLAN devices are transmitted.

Therefore, COR is an indicator of channel congestion and a value that has a strong correlation with throughput.

TABLE I: MCS index

MCS	Modulation scheme	Coding rate
0	BPSK	1/2
1	QPSK	1/2
2	QPSK	3/4
3	16-QAM	1/2
4	16-QAM	3/4
5	64-QAM	2/3
6	64-QAM	3/4
7	64-QAM	5/6
8	256-QAM	3/4
9	256-QAM	5/6

C. Modulation and Coding Scheme

The modulation and coding scheme (MCS) is an indexed combination of modulation scheme and coding rate. In the 5 GHz band 802.11ac, the MCS is given in 10 steps from 0 to 9. The correspondence is shown in Table I [7].

In the IEEE 802.11 physical layer, these values are closely related to the theoretical values of the transmission rate. The theoretical transmission rate DR [Mbps] is given by Eq.(2):

$$DR = \frac{N_{SD} \times N_{BPSCS} \times R \times N_{SS}}{T_{DFT} + T_{GI}}, \tag{2}$$

where N_{SD} is the number of data subcarriers, N_{BPSCS} is the number of coded bits per subcarrier per stream, R is the coding rate, N_{SS} is the number of spatial streams, T_{DFT} [ns] is the OFDM symbol duration, and T_{GI} [ns] is the guard interval duration.

Note in Eq.(2) that the coding rate and N_{BPSCS} have a proportional relationship with the data rate. Besides, N_{BPSCS} depends on the modulation scheme. Therefore, MCS having two pieces of information, modulation scheme and coding rate, can be considered useful for predicting throughput.

III. Potential Throughput Prediction using Communication Logs

A. System Model

Fig. 1 shows the system model we consider in this paper. We assume a situation where a target terminal and another terminal exist in the same AP coverage area. In this situation, while another terminal is in downlink (DL) communication with the AP, the target terminal starts DL communication with the same AP. The purpose is to predict the target terminal's potential throughput Th_p, or the throughput that the target terminal will be able to achieve.

B. Proposed Method

In order to predict the potential throughput, it is first necessary to know the maximum throughput that can be achieved at the target terminal in the absence of other terminals. In

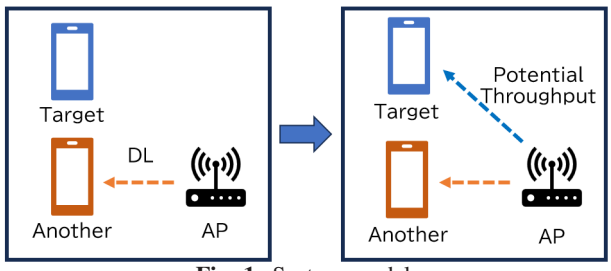
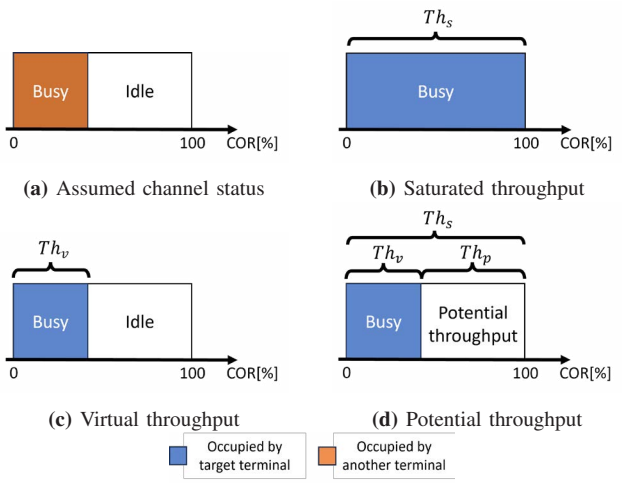


Fig. 1: System model.



(a) Assumed channel status **(b)** Saturated throughput

(c) Virtual throughput **(d)** Potential throughput

Fig. 2: Channel occupancy and throughput.

addition, determine the throughput that the target terminal could have achieved in the time that another terminal is communicating. In this paper, the former is defined as saturated throughput Th_s and the latter as virtual throughput Th_v.

We interpret saturated throughput and virtual throughput in terms of channel occupancy. The saturated throughput is the throughput that can be achieved when the target terminal occupies the entire channel. The virtual throughput is the throughput that could be achieved if the target terminal was to occupy the part of the channel occupied by another terminal. So, by subtracting the virtual throughput from the saturated throughput, the potential throughput can be obtained. Thus, the potential throughput Th_p can be obtained using Eq.(3):

$$Th_p = Th_s - Th_v. \tag{3}$$

This relationship between COR and each throughput is visually expressed as Fig. 2.

From the above, we consider creating estimation models for saturated throughput and virtual throughput to obtain each throughput. Since saturated throughput is strongly correlated with RSSI as described in Section II, it is reasonable to choose RSSI as the input to the estimation model. On the other hand, the inputs to the virtual throughput estimation model are RSSI, COR and MCS.

Therefore, the procedure for estimating the potential throughput first obtains the RSSI, COR and MCS as communication logs. Next, RSSI is input to the saturated throughput estimation model and RSSI, COR and MCS are input to the virtual throughput estimation model. Finally, for the output from each model, the potential throughput is calculated by subtracting the virtual throughput from the saturated throughput.

From the obtained communication logs, prepare a dataset to create the estimation models. The logs record the instantaneous values of the measured indicators. In actual systems, some processes such as resource allocation are based on smoothed instantaneous value [8]. Therefore, the smoothed version of the communication logs should be used as the training dataset. Specifically, the collected logs were smoothed using the exponentially weighted moving average (EWMA) expressed in Eq.(4):

$$y_t = \sum_{i=0}^{t} \alpha(1-\alpha)^i x_{t-i}, \tag{4}$$

where y_t is the EWMA data corresponding to time t, x_{t-i} is the data point at time $t-i$, α is smoothing factor and is described using the smoothing window w as in Eq.(5):

$$\alpha = \frac{2}{w+1}. \tag{5}$$

EWMA gives the highest weighting to the most recent data, and the weight decreases exponentially as the data becomes older. This makes it possible to track channel fluctuations appropriately based on historical data.

For creating the estimation models, we use random forest (RF). This is because RF has advantages such as robustness to high-dimensional input data and low computational cost and so it is suitable for real-time measurements [9]. Random forests are a type of supervised learning that combines two methods; bagging and decision trees. Supervised learning is a method of learning with correct answers given to the training data. The dataset used for training requires explanatory and objective variables. The explanatory variables are the data that explain the objective variable and the objective variable is the correct data that wants to be predicted. By using explanatory variables that are associated with the correct data, it is possible to create a model that can derive the correct data using only the explanatory variables.

Bagging randomly reconstructs and extracts data from the training data and uses that data to create a weak learner. Then, each of these results is collected and a final decision is made based on the average value. RF employs a decision tree as the weak learner. It is a system in which two-choice questions are connected in a hierarchical structure and the correct answer is finally obtained by answering the questions one by one.

IV. Experimental Evaluation and Results

This section describes measurement experiments carried out to evaluate the accuracy of the prediction of potential throughput using the proposed method. We also confirm the estimation accuracy of the saturation and virtual throughput estimation models actually created from the data collected in the experiments and the accuracy of predicting potential throughput using these models.

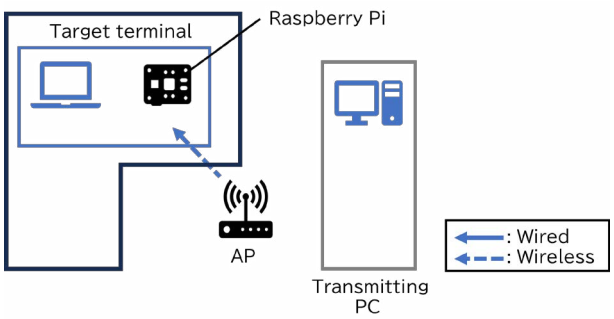


Fig. 3: Experimental setup (obtaining training data for saturated and virtual throughput estimation model).

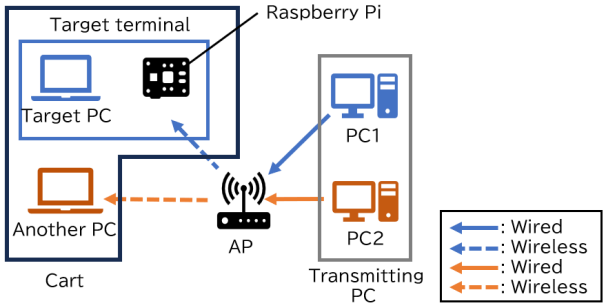


Fig. 4: Experimental setup (obtaining evaluation data for potential throughput prediction model).

A. Measurement Overview

First, we collected training data for the saturated and virtual throughput estimation model using the experimental setup shown in Fig. 3. In the experiment, UDP DL traffic was sent from the transmitting PC to the target PC for 30 seconds at X Mbps using iperf3. X was set from 10 to 700 Mbps in 10 Mbps increments. The above measurements were repeated while changing the distance from the AP.

Every second RSSI, COR, MCS and throughput were obtained from these measurements and used as training data for the virtual throughput estimation model. As the traffic increased, there was a point at which the throughput hardly increased any further, and the RSSI and throughput at this point were used as the training data for the saturated throughput estimation model.

RSSI, COR and MCS are gotten by typing the Raspberry Pi OS of "iw" command on Raspberry Pi. "iw" is a command line utility for configuring a wireless LAN network. If the interface name of the WLAN dongle is "wlan1", RSSI and MCS can be directly taken by typing "iw dev wlan1 link". On the other hand, COR is calculated from "channel active time" and "channel busy time", which can be obtained by "iw wlan1 survey dump". "channel active time" is the time that has elapsed since the terminal connected to the channel, and "channel busy time" is the length of time that the channel was busy. Therefore, COR can be calculated by obtaining "channel active time" and "channel busy time" every second, calculating the increment of each, and using Eq.(1).

The model was created using a random forest with 80% of the training data and the remaining 20% of the test data.

The root mean squared error (RMSE) expressed

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2} \tag{6}$$

was used to evaluate the estimated model. Note that $\hat{y}_i$ refers to the estimated value and y_i refers to the correct value.

Next, we acquired evaluation data for the potential throughput prediction model using the experimental setup shown in Fig. 4. The target PC, Raspberry Pi and another PC were connected wirelessly to the AP, and the traffic transmitting PCs were wired to the AP. First, a constant UDP DL traffic was applied from the traffic transmitting PC1 to another PC using iperf3, and the RSSI, COR and MCS were measured by the Raspberry Pi for 10 seconds. The average value of these 10 seconds was used as input for the potential throughput prediction model. Next, while keeping the traffic on the other PC constant, increasing the traffic sent from the traffic transmitting PC2 to the target PC by increments of 10 Mbps and measuring the throughput. The throughput was measured for 30 seconds for each traffic, and the value with the largest median was used as the correct value of potential throughput.

TABLE II: Experimental Parameters

AP/Wi-Fi	Name	Archer C80
	Number of antennas	4
	Standard	IEEE 802.11ac 5 GHz
	Center frequency	5240 MHz
	Channel width	80 MHz
Transmitting PCs	Name	MINISFORUM TL50
	CPU	Intel® Core™ i5-1135G7
	RAM	16 GB
	LAN port	2.5 Gbit
	OS	Windows 10 Pro
	Software	iperf3 3.16
Receiving PC Another PC	Name	dynabook S73/H
	CPU	Intel® Core™ i5-1135G7
	RAM	16 GB
	USB port	USB 3.0
	OS	Windows 11
	Software	iperf3 3.16
Raspberry Pi	Name	Raspberry Pi4 ModelB 4GB
	OS	Raspberry Pi OS
WLAN USB dongle	Name	Netgear AXE3000
	Number of antennas	2
	Interface	USB 3.0
LAN cable	Maximum transmission speed	1 Gbit

B. Experimental Specifications

Table II shows the experimental specifications. In these experiments, the 5 GHz band 802.11ac was used, with fixed channel 48 and a channel width 80 MHz. The devices connected wirelessly to the AP were equipped with a WLAN USB dongle and use it to communicate with the AP.

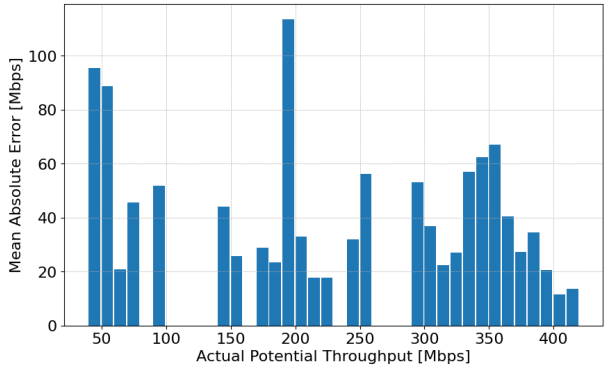


Fig. 5: Distribution of the error of the potential throughput to the correct value.

C. Results and Evaluation of Prediction Accuracy

The RMSE for the saturated throughput estimation model was 18 Mbps. Approximately 91% of the test data was within an error rate of ±30 Mbps. About 83% of the test data was within the error rate of 5% or less.

The RMSE of the virtual throughput estimation model was 24 Mbps. Approximately 85% of the test data was within the error rate of ±30 Mbps. About 86% of the test data was within an error rate of 15% or less. The importance of each feature was 0.28 for RSSI, 0.70 for COR and 0.02 for MCS, which means that COR is the most important and MCS is almost useless for virtual throughput estimation.

The distribution of prediction errors for potential throughput against correct values is shown in Fig. 5. This shows that in most cases there is a fixed error of 30 to 40 Mbps, regardless of the value of the actual potential throughput. This seems to be within the proper range based on the RMSE of the saturation and virtual throughput estimation model.

The distribution of the absolute errors of the potential throughput for RSSI and COR are shown in Fig. 6. For the 70 evaluation data, the average absolute error of the potential throughput was about 45 Mbps. In particular, focusing on the part of RSSI greater than −60 dBm, there were 40 of the 54 evaluated data within an error rate of 20% or less. The average absolute error of the potential throughput was about 42 Mbps, and the average error rate was about 16%. On the other hand, at points where the RSSI was smaller than −60 dBm, the average absolute error was about 60 Mbps and the average error rate was about 94%. In such areas with low RSSI, the throughput values are inherently small, so the effects of the fixed errors discussed above are pronounced in the error rate. However, even considering this factor, the prediction accuracy is significantly reduced in these low-RSSI regions.

V. Conclusions and Future Work

In this paper, we proposed a method for predicting the potential throughput, i.e., the throughput that can be newly achieved by the terminal without applying new traffic, by using communication logs. The proposed method obtains RSSI, COR, and MCS as communication logs. Then, the saturated throughput, i.e., the throughput that a terminal can achieve in the absence of other terminals, is estimated from the

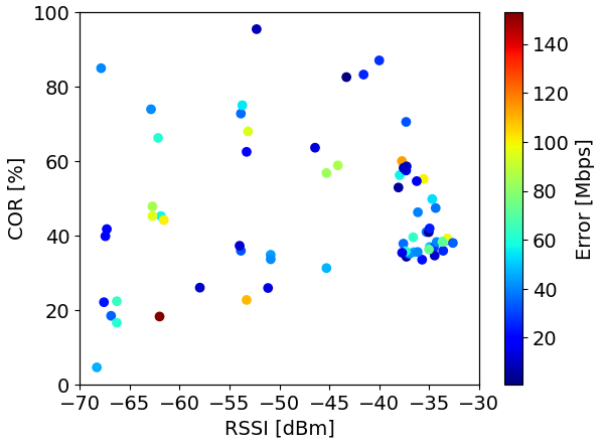


Fig. 6: Distribution of absolute error for RSSI and COR.

RSSI, and the virtual throughput, i.e., the throughput that a terminal cannot achieve due to channel occupation by other terminals, is estimated from the RSSI, COR, and MCS. The potential throughput is calculated by subtracting the virtual throughput from the saturated throughput. From the results of the experimental evaluation, it was confirmed that the potential throughput can be estimated with an error of approximately 45 Mbps on average. It was also found that the potential throughput can be estimated with an error rate of about 16% if the RSSI is greater than −60 dBm in the area.

However, this prediction accuracy can cause problems in some cases when considering the use of this system in actual situations. Therefore, it is necessary to improve the prediction accuracy by extending the data set by making measurements in various environments, changing the machine learning used to create the estimation model, and increasing the number of indicators used as communication logs. In addition, in this experiment, we assumed a target terminal and one other terminal, but since the communication speed changes depending on the number of terminals in WLAN, it is necessary to confirm the prediction accuracy when the number of terminals is large.

References

[1] Google Inc.,“System requirements & supported devices for YouTube,” https://support.google.com/youtube/answer/78358?hl=en&sjid=12398588744344819770-AP (accessed Apr. 24, 2024)

[2] Amazon Web Services, “What’s The Difference Between Throughput And Latency?,” https://aws.amazon.com/jp/compare/the-difference-between-throughput-and-latency/ (accessed Apr. 24, 2024)

[3] E. Ak and B. Canberk, ”Forecasting quality of service for next-generation data-driven WiFi6 campus networks,” IEEE Transactions on Network and Service Management, vol.18, no.4, pp.4744-4755, Dec. 2021.

[4] Y. Tsuchiya, N. Suga, K. Uruma, K. Yano, Y. Suzuki, and M Fujisawa, “WLAN throughput prediction using deep learning with throughput, RSS, and COR,” 2022 International Symposium on Intelligent Signal Processing and Communication Systems, Penang, Malaysia, Nov. 2022.

[5] T. Hasegawa and H. Takeno, “Relations of RSSI and average throughput of IEEE802.11n wireless LAN system,” IPSJ Journal, vol.52, no.9, pp.2829-2840, Sept. 2011.

[6] Z. Shah, S. Rau and A. Baig, “Throughput comparison of IEEE 802.11ac and IEEE 802.11n in an indoor environment with interference,” 2015 International Telecommunication Networks And Applications Conference (ITNAC), Sydney, Australia, pp. 196-201, Nov. 2015.

[7] M. Darwish, M. B. Ali, M. Altaeb, S. O. Sati, and M. S. Elmusrati, "Comparison between high throughput and efficiency of 802.11 wireless standards," 2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Nov. 2022.
[8] D. Raca, A. H. Zahran, C. J. Sreenan, R. K. Sinha, E. Halepovic, and V. Gopalakrishnan, "Device-based cellular throughput prediction for video streaming: lessons from a real-world evaluation," IEEE Transactions on Machine Learning in Communications and Networking, p. 1, Jan. 2024.
[9] N. Stepanov, D. Alekseeva, A. Ometov and E. S. Lohan, "Applying machine learning to LTE traffic prediction: comparison of bagging, random forest, and SVM," 2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Brno, Czech Republic, pp. 119-123, Oct. 2020.