2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS) | 978-1-6654-6824-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/IWQoS54832.2022.9812912

Bandwidth Prediction for 5G Cellular Networks

Yuxiang Lin[1,2], Yi Gao[1,3], Wei Dong[1,3]
[1]Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Zhejiang University
[2]Alibaba Group [3]Alibaba-Zhejiang University Joint Institute of Frontier Technologies
Emails: {linyx, gaoyi, dongw}@zju.edu.cn

***Abstract*—Effective bandwidth prediction in the fifth-generation (5G) cellular networks is essential for bandwidth-consuming applications, such as virtual reality and holographic video streaming. However, accurate bandwidth prediction in 5G networks remains a challenging task due to the short-distance coverage and frequent handover properties of 5G base stations. In this paper, we propose HYPER, a HYbrid bandwidth PrEdiction appRoach using commercial smartphones. Hyper uses an AutoRegressive Moving Average (ARMA) time series predictive model for intra-cell bandwidth prediction and a Random Forest (RF) regression model for cross-cell bandwidth prediction. Our ARMA model takes prior bandwidth usage as its input, while the RF model further uses related network and physical features to predict future bandwidth. We conduct a measurement study in commercial 5G networks to analyze the relationship between these features and bandwidth. Moreover, we also propose a handover window adaptation algorithm to automatically adjust the handover window size and determine which model to use during handover. We use commercial 5G smartphones for data collection and conduct extensive experiments in diverse urban environments. Experimental results based on one TB of cellular data show that HYPER can reduce the bandwidth prediction error by more than 13% compared to state-of-the-art bandwidth prediction approaches.**

I. Introduction

The fifth-generation (5G) has attracted extensive attention in the communication industry and academia in recent years. Compared to fourth-generation (4G) LTE, a major improvement of 5G New Radio (NR) is its significantly higher bandwidth. In theory, 5G can deliver up to 20 Gbps peak downlink bitrates and more than 100 Mbps average data rates [22]. 5G, therefore, enables a wide range of applications with high bandwidth requirements such as virtual reality and cloud gaming. For these applications, being aware of the available bandwidth in advance can help achieve adaptive configuration (e.g., rendering granularity, image resolution, etc.) to improve the user experience.

Existing short-term bandwidth prediction approaches in cellular networks can be divided into two categories. One is based on time series predictive models [19], [24], [36], another is based on Machine Learning (ML) regression models [14], [15], [30], [33], [35]. The former takes a time series of past throughput as the inputs to predict cellular link bandwidth while the latter aims to further use related features, including upper-layer information (e.g., Round Trip Time (RTT), loss rate) and lower-layer information (e.g., signal strength, Signal-to-Noise Ratio (SNR), link

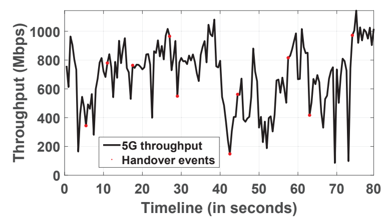


Fig. 1: **5G throughput and handover in a driving scenario.**

quality), to train a regression model. However, we find that both approaches cannot be directly applied in commercial 5G cellular networks. ML regression models suffer from low-fidelity PHY/MAC layer information obtained using Commercial-Off-The-Shelf (COTS) 5G smartphones, leading to relatively worse prediction accuracy. Time series predictive models usually work well for relatively stable networks. However, link bandwidth in cellular networks is highly dynamic, especially for 5G networks. We have conducted a microscopic measurements study in a driving scenario. Figure 1 shows that the short-range coverage of 5G base stations will introduce frequent handover. A handover event may cause significant throughput variations, making the time series model ineffective for 5G bandwidth prediction.

In this paper, we propose HYPER, a hybrid bandwidth prediction approach for 5G cellular networks that combines time series predictive models and ML regression models. HYPER aims to achieve accurate bandwidth prediction for 5G cellular networks with frequent handover events. Specifically, we first apply a light-weight AutoRegressive Moving Average (ARMA) time series predictive model to estimate the intra-cell bandwidth time series with significant temporal correlations, taking prior bandwidth usage of the current cell as the input. To further eliminate the prediction drift caused by significant throughput changes due to frequent handover, we use a Random Forest (RF) regression model to refine the bandwidth prediction results during base station handover.

However, accurate bandwidth prediction is still challenging from two aspects: (1) What features extracted from commercial devices can help improve bandwidth prediction accuracy? Besides the commonly used network features, 5G link bandwidth may be affected by a multitude of physical factors, such as moving speed and handover events [18]. (2) How to determine when to use the ARMA model and when to use the RF model during handover? We aim to use the ARMA model for bandwidth prediction as soon as possible

978-1-6654-6824-4/22/$31.00 © 2022 IEEE

after a handover event due to its good property of capturing the temporal correlations of time series. However, We find that the bandwidth will be affected during handover and even a long period after the handover event. There is a tradeoff in determining the size of handover windows[1] whose data are used for learning the ARMA parameters and simultaneously fed into the RF model for bandwidth prediction. A small window size may include insufficient training data for the ARMA parameters, while a larger one will lead to excessive use of the RF model. Both cases will lead to a reduction in the prediction accuracy. To address the above two challenges, we first conduct an extensive measurement study to extract available upper-layer (network layer), lower-layer (PHY/MAC layer), and physical information from COTS smartphones and identify which features are correlated with cellular link bandwidth. We then propose a handover window adaptation algorithm to automatically set the handover window size based on the stationarity of bandwidth time series.

We implement HYPER in COTS 5G smartphones without any change to the operating system or the need of rooting the phones. We mainly collect data in 5G cellular networks of China Mobile Communications Group (CMCC), a major 5G carrier in China, and extract related information through Application Programming Interfaces (APIs) provided by Android. Our experiments consume more than one TB of cellular data in diverse urban areas with different movement states. Experimental results show that HYPER can achieve a median downlink bandwidth prediction error of 10% in typical driving scenarios in urban areas, improving more than 13% compared to existing bandwidth prediction approaches. For the uplink, HYPER can achieve a median uplink bandwidth prediction error of 5%. We also integrate HYPER into a video streaming application and a congestion control application to show how HYPER can benefit existing bandwidth-related applications. More importantly, overhead analysis results show that HYPER will not introduce much computation overhead during online bandwidth prediction.

In summary, we make the following key contributions:

(1) We have conducted an extensive measurement study in commercial 5G cellular networks to investigate the commercially available information that is related to link bandwidth. We identify three upper-layer information, three lower-layer information, and two physical information that are significantly correlated with 5G cellular link bandwidth.

(2) We propose HYPER, a hybrid bandwidth prediction approach for 5G cellular networks. HYPER predicts the intra-cell bandwidth using an ARMA model and the cross-cell bandwidth using an RF model. We propose a handover window adaptation algorithm based on time series' stationarity to adaptively determine the handover window size and improve the bandwidth prediction performance during handover.

(3) We evaluate HYPER using COTS smartphones in commercial 5G networks. Results show that HYPER achieves a better bandwidth prediction accuracy than existing approaches in all the scenarios we have explored. The low system overhead of HYPER indicates that HYPER is lightweight for commercial 5G networks.

[1] For each cell, its handover window is defined as the period whose bandwidth will be affected by the handover event.

II. Related Work

A. Time Series Predictive Model-based Prediction

In cellular networks, time-stamped throughput data can be stored as time series and frequently show temporal correlations [4]. Historical throughput time series can be used for bandwidth prediction and network planning. Autoregressive models are common time series predictive models that have been extensively used in wireless networks. Papagiannaki et al. [19] use a low-order AutoRegressive Integrated Moving Average (ARIMA) model to capture the short memory process of traffic loads, while Sadek et al. [24] use a Gegenbauer ARMA model to specify long memory processes. There are also some studies using similar models to predict other related metrics in cellular networks. For example, in [11] and [27], autoregressive models are applied to predict future SNR values and handover events, respectively.

However, these autoregressive models can only work well in relatively stable networks without abrupt throughput changes. In other words, ARMA models work well inside base stations but have a poor prediction performance during handover. Thus, these approaches cannot be directly applied in 5G cellular networks due to their frequent handover.

B. ML Model-based Prediction

Using ML models for bandwidth prediction has received significant attention in recent years [12], [14], [17], [21], [30], [33]. Previous studies investigate different sets of features that are correlated with cellular link bandwidth for prediction purposes. Various ML models using different layers' features have been proved effective for bandwidth prediction in LTE networks. Proteus [30] use Regression Trees (RTs) to predict the short-term network performance while LinkForecast [33] uses an RF model. Liu et al. [14] proposes three offline prediction methods (i.e., Naive Bayes (NB), Logistic Regression (LR), and Artificial Neural Networks (ANN)) to conduct link prediction and further propose an online LR model-based link estimator TALENT [15]. Recently, PERCEIVE [13] exploits a Long Short Term Memory (LSTM) model for uplink bandwidth prediction.

However, using these ML models with COTS smartphones will also lead to poor prediction performance. We observe that lower-layer features used in these works (e.g., Reference Signal Received Power (RSRP), Reference Signal Received Quality (RSRQ), and Signal-to-Interference Noise Ratio (SINR)) are usually of low fidelity when collected from COTS smartphones. To make it worse, some commonly used lower-layer information is even not available in smartphones. For example, although we can access Channel Quality Indicator (CQI) through an Android API, its reported value is always a maximum value of *int* type, which means CQI is still a developing function.

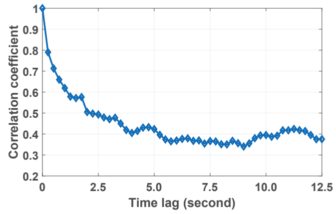
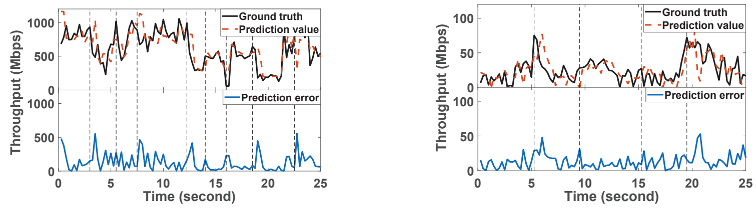


Fig. 2: **Correlation coefficients with different time lags.**



(a) 5G networks

(b) 4G networks

Fig. 3: **An example of bandwidth prediction results using ARMA models in (a) 5G and (b) 4G networks. Vertical dotted lines denote handover events.**

Summary: In HYPER, we prefer to use time series predictive models for bandwidth prediction due to the higher prediction accuracy. However, while time series predictive models cannot deal with handover events, we need to use ML models to assist in predicting. HYPER can adaptively allocate these two types of prediction models for bandwidth prediction within appropriate periods. We use ARMA and RF models in HYPER since (1) ARMA is proved to be as effective as deep learning approaches for predicting time series [4] and (2) RF achieves a trade-off between prediction accuracy and overhead, which is important for online prediction in 5G networks.

III. Background & Measurement Study

A. 5G Infrastructure

5G uses multiple frequency bands including mmWave bands and microwave bands [25]. The mmWave bands at frequencies of 24 to 53 GHz provide a considerable amount of bandwidth. However, mmWave's short wavelength makes its signals vulnerable to attenuation and blockages [26], leading to significant data rate variation and shorter deployment distances between base stations. Microwave 5G radios operate at mid-band frequencies of sub-6 GHz whose radio signal largely remains omnidirectional and offers a decent data rate. In our experiments, mid-band 5G radios are used by the CMCC carrier to form the basis of initial 5G services. However, to be compatible with higher frequency bands in the future, the distance between 5G base stations ($\approx$200 meters) deployed by CMCC is still much shorter than that of existing LTE networks ($\geq$500 meters) [1], [29], leading to frequent handover events.

B. Measurement Study

1) Data Collection Methodology: **5G UE.** We use two types of COTS 5G-capable smartphones, HUAWEI MATE30 5G (Hisilicon Kirin 990) and Xiaomi Mi10 5G (Qualcomm Snapdragon TM865), for 5G cellular data collection. Comparing their performance at the same locations, we have observed that the measurement results of these phones are consistent. We thus use MATE30 for all experiments. We confirm that despite 5G's high throughput, the device-side processing is not a bottleneck for MATE30, which is a high-end smartphone equipped with an eight-core CPU, 8 GB memory, Kirin 990 5G System-on-Chip (SoC) that fully integrates a Balong 5000 5G modem. MATE30 supports both 4G and 5G, allowing us to compare the throughput performance of these networks. In our experiments, we adopt the 5G StandAlone (SA) mode that is supported by CMCC across the city.

5G Monitoring Application. Android 10 [9] is an advanced Android OS and adds platform support for 5G NR related information. We are not aware of any commercial Android application that reports 5G NR-related information. Hence, we have developed a resource monitoring application based on Android 10 APIs for COTS 5G smartphones to obtain related information. We collect the following three categories of information at a sampling rate of 2 Hz to support our measurements: (1) Physical information includes system time, GPS, speed, and the connected cell ID. (2) Upper-layer information includes 5G service status, network interface, and addresses, upload and download throughput[2], packet loss rate[3], RTT, and the variation of RTT. (3) Lower-layer information includes RSRP, RSRQ, and SINR.

Server and Experimental Scenario. In our experiments, we select the HUAWEI Cloud server for speed testing because it provides a maximum bandwidth of 2 Gbps, which is much higher than the highest 5G speed we obtained. We upload a 2 GB file to a server instance on HUAWEI Cloud. We drive in the urban area at a speed of $\sim$50 km/h and repeat the file download 15 times without interruption. In total, we recorded $\sim$700 data samples for the measurement study.

2) Observations: We have analyzed the correlations between link bandwidth and a set of information.

Past Throughput. Although link bandwidth in 5G networks is highly dynamic, past throughput is still the most important information for predicting future bandwidth. Figure 2 shows the correlation coefficients of link bandwidth time series with different time lags l, where $l \in [0, 12.5]$ seconds. We can observe that the temporal correlation of link bandwidth time series is significant even at a few seconds. Therefore, we use the ARMA model in HYPER since it is effective to learn the function of previously observed values and random noise when the time series shows strong temporal correlations [4].

[2]While the state-of-the-art Android OS does not support directly measuring throughput, we use the total traffic difference between adjacent time slots to calculate throughput.

[3]We seldom observe, however, losses of *ping* packets in our measurement traces. Almost all measurements in our traces have a packet loss rate of 0. Hence, we do not take the packet loss rate as a feature. As packet loss is a very informative and vital metric to reflect the current situation of the network, we can still easily use it as a feature of HYPER for bad 5G networks.

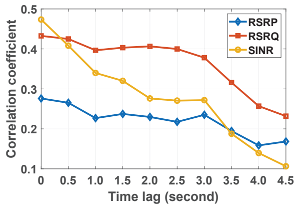


Fig. 4: **Correlation coefficient of RSRP, RSRQ, SINR.**

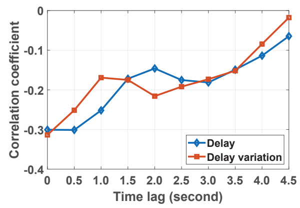


Fig. 5: **Correlation coefficient of RTT, the variance of RTT.**

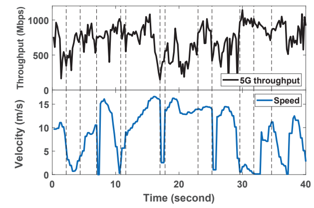


Fig. 6: **5G throughput and the corresponding moving speeds in a driving scenario.**

Handover Events (Cell ID). Similar to LTE networks, a UE can only connect to one base station in 5G cellular networks. In a continuous trace, a handover event happens when the cell ID changes, which means a UE disconnects from one cell and connects to another cell. There will usually be abrupt throughput changes during handover since different base stations may have different bandwidth capacity and resource scheduling algorithms, and serve for different numbers of UEs. Figure 3 plots an example of bandwidth prediction results based on ARMA models in both 5G and 4G networks. The upper figures show the predicted throughput and ground truth throughput. The bottom figures show the corresponding prediction error. We have the following important observations: (1) The ARMA model can achieve good prediction accuracy within a specific cell. Most bandwidth prediction errors can be less than 10%, which shows the effectiveness of the ARMA model for intra-cell bandwidth prediction. (2) The prediction error will significantly increase during handover. This is because the fitting procedure of ARMA assumes a stationary time series [2], while handover events will usually lead to significant throughput changes. (3) Comparing to the LTE networks, handover events happen more frequently in 5G networks. The number of spikes of prediction errors in 5G networks is more than twice that in 4G networks. Moreover, the absolute bandwidth prediction error of 5G networks is significantly larger since the bandwidth of 5G networks is an order of magnitude higher than that of 4G networks. As a result, inaccurately predicted bandwidth will have a greater impact on 5G networks.

RSRP, RSRQ, and SINR. RSRP, RSRQ, and SINR are PHY/MAC layer information that measures the network quality. In COTS smartphones, RSRP, RSRQ, and SINR are of low fidelity and are reported as integers using Android APIs. Let $\{b_i\}$ and $\{p_i\}$ denote the link bandwidth and a PHY/MAC layer information of the i-th second, respectively. Figure 4 shows the correlation coefficients between each PHY/MAC layer information time series $\{p_i\}$ and link bandwidth time series $\{b_{i+l}\}$ with different time lags $l \in [0,5)$ seconds. We can see that the correlation between each PHY/MAC layer information and link bandwidth is significant but will rapidly decrease with the increasing number of time lags. As a result, these data will easily become out-of-date for bandwidth prediction. We have also found that unlike the 4G LTE networks [33], the correlation between RSRP and link bandwidth is relatively less significant than that between the other two PHY/MAC layer information and link bandwidth. Inspecting the trace, we observe that the trends of RSRP and link bandwidth are not consistent at times. This may be because RSRP values will also be affected by other distinct factors of 5G networks such as the orientation of 5G base stations (deployed in the form of panels) [18].

RTT and Variation of RTT. RTT is a typical network layer information and is highly related to link bandwidth. There is often a conflict between throughput and packet RTT [16]. We integrate the *ping* command into our monitoring application to measure the RTT to the server. Figure 5 shows the cross-correlation between link bandwidth and RTT and its variances. We can see that both metrics have a significant negative cross-correlation with link bandwidth. Similarly, the correlation will also rapidly decrease as the time lag increases.

Speed. Due to the density of 5G base stations and the orientation of 5G signals, moving speed will have a great impact on bandwidth. Figure 6 shows an example trace of 5G throughput and speeds in a driving scenario using measurements collected from commercial 5G networks. Dot lines in the figure are the handover events. We can see that: (1) Abrupt speed changes can easily cause significant bandwidth changes. (2) When driving in urban areas, handover events can occur in less than ten seconds. As the average moving speed is around 10m/s, the distance between adjacent 5G base stations is usually less than 200 meters in our experiments. (3) Another interesting observation is that handover events usually occur when speed changes. This may be because the moving speed often changes at intersections in urban areas where there are usually more people and UEs sharing the spectrum, leading to more dynamic signal indicators (e.g., RSRQ, SINR) [29]. Besides, more base stations will be deployed at intersections and respond to handover events when their signal indicators are higher than that of the current serving cell [29]. Therefore, a handover decision will be more likely to be triggered at intersections with speed changes.

Summary. The above analysis shows that the intra-cell bandwidth time series has significant temporal correlations. However, time series predictive models alone cannot work well for 5G bandwidth prediction due to the frequent handover events. On the other hand, we have identified several bandwidth-related information that can be reported by COTS smartphones, including three upper-layer information (past throughput, RTT, the variation of RTT), three lower-layer information (RSRP, RSRQ, SINR), and two physical features (handover, speed). HYPER uses these features to help improve the bandwidth prediction performance during handover.

IV. HYPER Design

A. Intra-cell Model

In HYPER, we use the ARMA model to learn the temporal correlation of intra-cell bandwidth time series. The general ARMA model can be modeled as ARMA(p,q), where p denotes p AutoRegressive (AR) terms and q denotes q Moving Average (MA) terms. Optimizing the performance of the ARMA prediction model requires tuning of the parameters p and q. For each base station, the data used to obtain its training parameters p and q is the bandwidth time series in the handover window. We tune the parameters p and q by a grid search while varying p and q between 0 and 3. We use the Akaike Information Criterion (AIC) for finding p and q [3]. AIC offers a relative estimate of the quality of candidate ARMA models for a given time series. It not only considers the goodness of fit of an ARMA model, but also considers a penalty, which discourages overfitting, to help reduce the complexity of the model. The ARMA parameters p and q are retained for every step prediction within the same base station. For each step, we retrain the ARMA model and predict the bandwidth of the next time slot. This will not introduce much overhead since the ARMA model is lightweight and the throughput data for training is limited within a 5G base station.

B. Cross-cell Model

To compensate for the predicted values during handover, HYPER uses an RF regression model for cross-cell bandwidth prediction due to its capabilities of requiring low memory and computation overhead during online prediction, modeling complex relationships, and avoiding overfitting.

1) Problem Formulation: The regression model takes the related past information as input to predict the bandwidth of the next time slot. In other words, the input to train the regression model is the feature vectors of the past W time slots. Each feature vector contains all the related features mentioned in Section III-B2 except for the cell ID information. Let F_t denote the feature vector at time slot t, an input vector I_t can be expressed as $I_t = [F_{t-1}, F_{t-2}, ..., F_{t-W}]$. The related features have different value ranges and are scaled down to the range of [0,1]. The corresponding train label is the bandwidth B_t at time slot t. Let $M()$ denote the prediction model learned offline. The regression problem can be formulated as $B_t = M(I_t)$.

2) Cross-cell Prediction: In HYPER, we empirically set $W = 1$, which means we use the input vector of the past single time slot to train the RF model. Detailed prediction performance using different numbers of past time slots will be shown in Section V-F. Once the input vectors and the labels are constructed, we randomly select 70% of the total inputs as the training data and use the remaining 30% as the testing data. We perform a grid-based search on the training set to automatically find the optimal hyper-parameters of the RF model to maximize the classification accuracy. Once the RF model is constructed, we use the model to predict the bandwidth during handover.

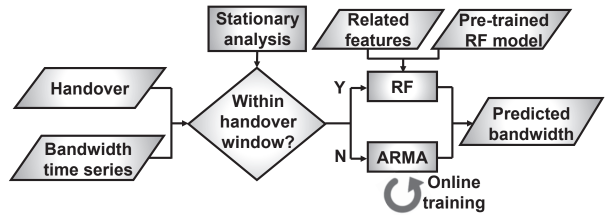


Fig. 7: **Overview of HYPER and its workflow.**

C. Hybrid Prediction Approach

To adaptively combine the above two models, we propose a hybrid prediction approach HYPER to achieve online bandwidth prediction. For a base station, HYPER first uses the pre-trained RF model to predict the bandwidth after the handover event. After determining the handover window size, HYPER will use the data in the handover window to learn the ARMA parameters p and q. Then HYPER will retrain the ARMA model at each time slot to predict the intra-cell bandwidth. Figure 7 shows an overview of HYPER.

It is non-trivial to determine the handover window size. A larger handover window size with more training data will lead to more accurate ARMA models but overuse the RF regression model. To adaptively determine the handover window size for different base stations, we propose a handover window adaptation algorithm based on the stationarity of bandwidth time series. Specifically, after a handover, HYPER will continuously check the stationarity of past bandwidth time series within the new base station. At the same time, HYPER will use the RF model to predict the bandwidth for these time slots. Once the time series becomes stationary, HYPER will determine the handover window size, learn appropriate ARMA parameters of the new base station, and switch to the ARMA model for subsequent bandwidth prediction. It is worth noting that although the total time for handover procedures is small, e.g., less than 200ms [29], the throughput can be affected for several seconds after handover. This is because the communication resources may not be scheduled in time due to the complex cell load [31]. In our experiments, we also observe that the handover window size is usually in seconds.

HYPER uses a lightweight time series stationary analysis approach, the Augmented Dickey-Fuller (ADF) test [6], to quantify the stationarity of a time series. The intuition behind the ADF test is that it can characterize how strongly a time series is stationary [8]. We can get two stationary metrics after running the ADF test on a time series: an ADF value V_{ADF} and a p-value V_p. For the ADF value, it should be a negative number. If the ADF value is less than a confidence value TH_{ADF}, we can assume the time series is stationary. On the other hand, a p-value below a threshold TH_p suggests the time series is stationary. In HYPER, we consider the bandwidth time series is stationary only when both metrics are less than the corresponding thresholds. In our current design, we carefully set TH_{ADF} to a confidence value of 1% and TH_p to 0.5 to ensure that HYPER captures a stationary time series with great confidence.

Algorithm 1 shows the details of our hybrid prediction

Algorithm 1 Hybrid Bandwidth Prediction Algorithm

Input: Time series of input vectors $\mathbf{I} = \{I_1, I_2, ..., I_t, ...\}$; Time series of bandwidth $\mathbf{B} = \{B_0, B_1, ..., B_{t-1}, ...\}$; Time series of connected cell ID $\mathbf{ID} = \{ID_0, ID_1, ..., ID_{t-1}, ...\}$, where ID_t is the connected cell ID at time slot t; Thresholds of ADF value, p-value and bandwidth changes TH_{ADF}, TH_p, TH_e; Minimum handover window size S_{min}; Pre-trained RF model $M()$

Output: Predicted bandwidth series $\hat{B} = \{\hat{B}_1, \hat{B}_2, ..., \hat{B}_t, ...\}$

$startOfTrain \leftarrow 0$, $gotPAndQ \leftarrow False$
for t = 1 to ∞ **do**
  if ID_t != ID_{t-1} or $(\hat{B}_{t-1} - B_{t-1})/B_{t-1} > TH_e$ **then**
    $startOfTrain = t$, $gotPAndQ \leftarrow False$, $isStationary \leftarrow False$
  if $t - startOfTrain < S_{min}$ **then**
    $\hat{B}_t = M(I_t)$
    continue
  if $!isStationary$ **then**
    $(V_{ADF}, V_p) = adfuller(B_{[startOfTrain:(t-1)]})$
  if $V_{ADF} > TH_{ADF}$ or $V_p > TH_p$ **then**
    $\hat{B}_t = M(I_t)$
  else
    $isStationary \leftarrow True$
    if $!gotPAndQ$ **then**
      $(p, q) = armaOrderSelect(B_{[startOfTrain:(t-1)]})$
      $gotPAndQ \leftarrow True$
    $armaModel = ARMATrain(B_{[startOfTrain:(t-1)]}, p, q)$
    $\hat{B}_t = armaModel(t)$

algorithm. For each time slot, HYPER first uses the cell ID of the current serving cell to identify a handover event. Once a handover event is detected, HYPER needs to determine the handover window size of the new cell. It is worth noting that the intra-cell bandwidth may also change rapidly due to the nature of wireless communication and radio resource scheduling [34]. Therefore, we consider that the relative bandwidth prediction error of the last time slot larger than a predefined threshold TH_e will also trigger the handover window adaptation process. In HYPER, we empirically set TH_e to 1. Next, we define a minimum handover window size S_{min} to eliminate unnecessary stationarity tests and ensure the amount of data for training the ARMA model. For each time slot after S_{min}, HYPER calculates the two stationary metrics V_{ADF} and V_p using the $adfuller$ function available in the Statsmodels package [20]. When the time series becomes stationary, HYPER sets the handover window size to the number of past time slots since the handover event. In practice, we observe that even in driving scenarios with fluctuating throughput, most bandwidth time series can be identified as stationary within three seconds after handover. As a result, we set S_{min} to 5 in our experiments. Detailed prediction performance of different S_{min} will be shown in Section V-F. Then HYPER will use the bandwidth time series in the handover window to calculate the ARMA parameters p and q for the current cell. Finally, HYPER trains the ARMA model using these two parameters to estimate future bandwidth. Inside the handover windows, HYPER uses the pre-trained RF model to predict the bandwidth.

In HYPER, the RF model will not introduce much computation overhead. However, the ARMA parameter estimation process is relatively computation-intensive. On the other hand, we have also found that the ARMA parameters usually remain stable in a short time series. Considering both accuracy and complexity, HYPER only estimates p and q once when the handover window size is determined for the new cell. Detailed prediction overhead will be shown in Section V-G.

TABLE I: **Details about the 5G dataset.**

Data Samples	23893 samples (per half-sec throughput with features)
Mobility & Duration	Driving (9 Days), Walking (5 Days), Stationary (2 Days)
Data Amount	1130 GBs of 5G downloaded data
Cells & Handover	510 cells & 670 handover events

V. EVALUATION

A. Implementation

We implement the monitoring APP on commercial smartphones. From our measurement study, we have found that most features (e.g., handover, speed, and low-layer features) collected from commercial smartphones will only change every few seconds even in high-speed moving scenarios. Therefore, we limit the sampling rate in our monitoring application to second-level (i.e., 2Hz) and upload the collected information to a server every 500ms to remove redundant data and reduce computational overhead on the phone. We use a desktop with Intel i7-8700 CPU and 16GB of memory as the server to deploy HYPER. The server is responsible for processing the received information and returning the predicted bandwidth value to the UE. Then the UE can use the predicted bandwidth to make adaptive configurations for its applications. We provide a simple video streaming application and a congestion control application in Section V-B to show how HYPER can help improve the user experience.

The 5G information of our work is mainly collected in urban driving scenarios during the daytime and night of 9 days starting from June 4, 2020. We also collect data in two other mobility scenarios: walking and stationary. For the walking scenario, the data were collected on a university campus ($0.7km \times 1.2km$). For the stationary scenario, we collected data in a meeting room ($8m \times 5m$) on the campus. All the three experiment scenarios and the data collection and analysis devices are depicted in Figure 8. Not surprisingly, we observe more handover events in driving scenarios than in walking and stationary scenarios. In summary, the collected datasets cover a wide range of scenarios, including different times of the day, locations, and movement speeds. The total amount of downloaded data is over one TB. The total numbers of unique cell IDs and handover events are 510 and 670, respectively. Table I shows the full dataset statistics.

B. Prediction Results & Comparison Study

1) Comparison with ML Regression Model-based Approaches: We compare the prediction accuracy of HYPER with prediction approaches that use **RF** [21], [33], **NB**, **LR**, **ANN** [14], [15], [23], and **RT** [30] ML regression models

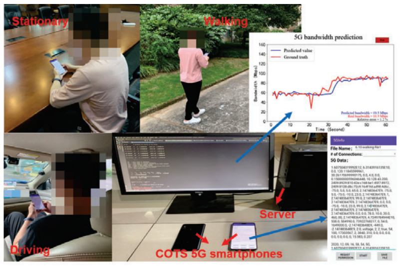
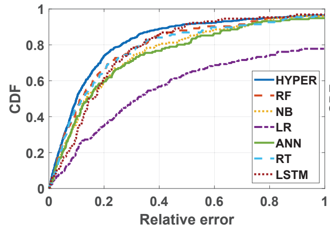
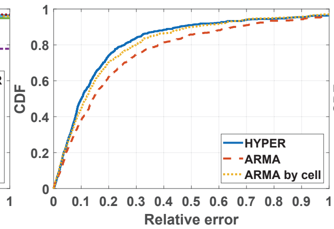


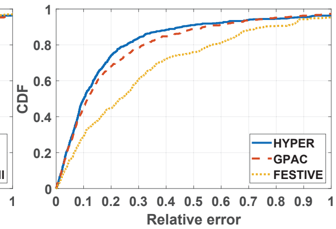
Fig. 8: **HYPER experimental scenarios, applications, and devices.**



(a) Comparison with ML model-based approaches



(b) Comparison with ARMA model-based approaches



(c) Comparison with HTTP-based adaptive streaming protocols

Fig. 9: **Bandwidth prediction performance comparison between HYPER and other approaches.**

and state-of-the-art deep learning-based approaches [12], [13], [17] that uses **LSTM** for bandwidth prediction. It has been proved that the training window size will not significantly affect the prediction accuracy of these models [14], [33]. Considering that using a larger training window size will lead to a higher computation overhead, we set the window size to 1 in these approaches. We carefully perform a grid-based search to find the optimal hyper-parameters when constructing the ML models. In particular, we implement the LSTM neural network following PERCEIVE [13] and train the LSTM using the past 10s input of feature vectors.

Figure 9(a) shows that the median relative prediction error of HYPER is 10.0%, while the median relative prediction errors of ML regression models are between 11.5% and 30.2%. Compared to ML regression model-based approaches, HYPER improves the prediction accuracy by 13.0%-66.9%. These ML model-based approaches do not work well due to the coarse-grained features exposed from the Android OS. We also observe that the LSTM model does not perform well in our experiments. This is because the model needs millisecond-level information to achieve good prediction accuracy [13]. However, as mentioned earlier, most features collected from commercial smartphones will only change every few seconds and are second-level information. The LSTM model will easily include stale features due to the low sampling rate and frequent handover in commercial smartphones, leading to poor prediction accuracy. Besides, LSTM-based approaches require significantly more time for model training/updating and thus are hard to achieve online prediction for the diverse and changeable 5G networks. In contrast, HYPER shows fast and accurate bandwidth prediction results as it carefully considers the handover events and effectively learns the temporal correlations inside base stations.

2) Comparison with ARMA Model-based Approaches: We compare HYPER with the standard ARMA model [2]. The ARMA parameters p and q are selected by varying the parameters for p and q between 0 and 3. For each time slot, we retrain the ARMA model using the throughput of the past 5s. After significant throughput changes where the ARMA model cannot work, we will simply use the last throughput as the predicted value. We further implement an ARMA-by-cell model that only uses the past throughput of the current cell to train the ARMA models. In this model, we also use the last throughput as the predicted value during handover.

Figure 9(b) shows the prediction results of these two ARMA model-based approaches. As seen, the ARMA and ARMA-by-cell approaches achieve median relative prediction errors of 14.3% and 11.7%, respectively. Results show that using the past throughput in the same base station to predict the future bandwidth can achieve higher accuracy. Taking a closer look at Figure 9(a) and (b), the ARMA-by-cell model performs slightly better than the RF model, showing the effectiveness of the ARMA model in predicting 5G link bandwidth.

3) Comparison with HTTP-based Adaptive Streaming Protocols: Many popular HTTP-based adaptive streaming protocols will use historical bandwidth records to predict future bandwidth for video streaming [7], [10], [28]. We make a comparison study with two existing protocols. One is the GPAC's player [7] that uses the last seen bandwidth as the prediction value for the next video segment. Another is FESTIVE [10], which uses smoothed harmonic mean of historical throughput measurements over a time window as the prediction. As recommended in [10], we use the harmonic mean over the last 20 samples to predict the future bandwidth.

Figure 9(c) shows the comparison results of HYPER and the two adaptive streaming protocols. Results show that compared to GPAC and FESTIVE, HYPER can improve the median relative prediction errors by 13.0% and 57.4%, respectively. In particular, the FESTIVE algorithm cannot perform well because it results in severe bandwidth under-utilization with the large bandwidth fluctuation of 5G networks.

C. Uplink Performance

We collect a small uplink dataset with 1,000 samples to show the effectiveness of HYPER in 5G uplink. During the experiment, we continuously upload file data to the server and conduct walking tests on the campus with normal walking speeds. We use 80% samples to train the RF model and use the remaining samples to evaluate the uplink bandwidth prediction performance of HYPER. Figure 10 (left) shows the ground truth and predicted uplink bandwidth. We can also observe that there are significant uplink throughput changes during handover. Figure 10 (right) shows the corresponding uplink bandwidth prediction error. As seen, HYPER can achieve a median relative prediction error of 5.1%, which is significantly

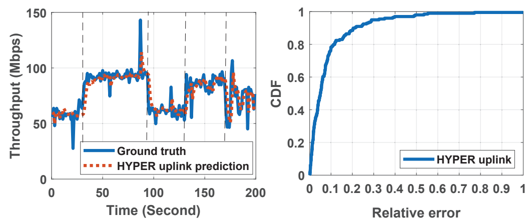


Fig. 10: **Uplink bandwidth prediction performance of HYPER. Dotted lines are the handover events.**

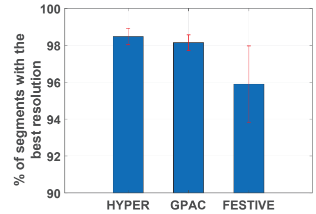


Fig. 11: **Video streaming performance comparison.**

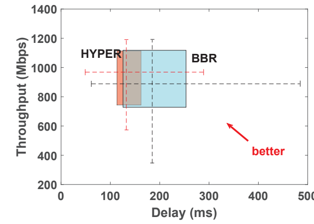


Fig. 12: **Congestion control performance comparison.**

better than that (i.e., 10%) of 5G downlink. This is because the bandwidth variation of the uplink is much smaller than that of the downlink, which makes the prediction results more accurate.

D. Case Study 1: Video Streaming

We compare the prediction performance of HYPER, GPAC, and FESTIVE in a video streaming application. A UHD video with a 120 FPS frame rate is split into 0.5s segments and stored on the server. Each segment is encoded into four copies with different resolutions (1080P, 4K, 8K, 16K). For the four resolutions, we conservatively define the corresponding bandwidth ranges as [0,80], [80,320], [320,640], and [640,∞] in Mbps, to prevent video stalling due to remarkable bandwidth fluctuations. The client specifies the resolution to download each segment based on the predicted bandwidth. Figure 11 shows the percentages of downloaded segments with the best resolution when using different approaches for video streaming. Comparison results indicate that HYPER can reduce the video stalling time by up to 2.6% for state-of-the-art adaptive video streaming applications using 5G networks.

E. Case Study 2: Congestion Control

Another important follow-up application of bandwidth prediction is congestion control. With an accurately predicted bandwidth, senders can match their sending rate to the available bandwidth capacity precisely and rapidly. We propose an end-to-end HYPER-based congestion control algorithm that adaptively sets the senders' rates to the predicted bandwidth of HYPER. We have preliminarily evaluated our HYPER-based congestion control algorithm using traces collected from commercial cellular networks. We compare our algorithm with a leading congestion control algorithm BBR [5].

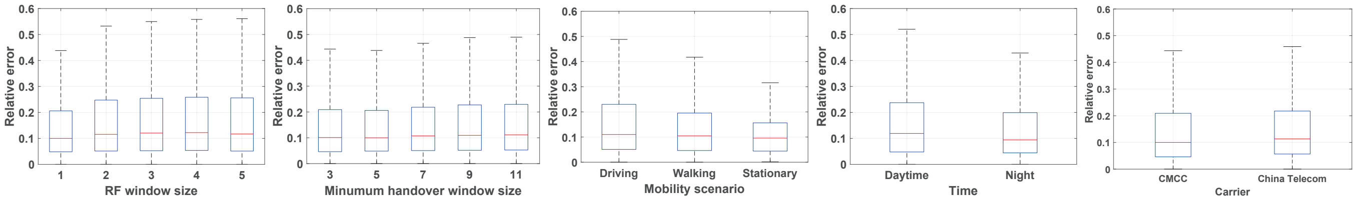
Figure 12 shows the minimal, 25^{th}, median, 75^{th}, and maximal percentile throughputs and delays of our HYPER-based congestion control algorithm and BBR. We can observe that HYPER helps achieve lower delay while ensuring comparable throughput with BBR. This is because, with accurate predicted bandwidth information, our algorithm will introduce less packet congestion, thus reducing the average packet transmission delay. We can also observe that comparing to BBR, our algorithm achieves low variance in both delay and throughput by accurately estimating the bandwidth capacity.

F. Impact Factors

1) Impact of RF Window Size: The information in different numbers of past windows used to train the RF model will affect the prediction performance during handover. To evaluate the impact of RF window size, we use historical information in the past 1-5 time slots (i.e., 0.5s-2.5s) to train the RF model. Figure 13(a) shows the prediction error of different RF window sizes. Results show that adding more historical data does not improve the prediction accuracy. On the contrary, the accuracy may be degraded. The reason might be that historical related features can quickly become out-of-date in rapidly changing cellular networks. Using features further away in the past will degrade the prediction accuracy. Moreover, using a larger RF window size will lead to a higher computation overhead. Therefore, we set the RF window size to 1 in HYPER.

2) Impact of Minimum Handover Window Size: We evaluate the performance of different minimum handover window sizes S_{min} in Algorithm 1. An appropriate S_{min} can effectively reduce the prediction overhead and maintain a good accuracy of the ARMA model. We set the minimum handover window size to 3, 5, 7, 9, and 11 in our experiments. Figure 13(b) shows that the median relative prediction errors will increase when the window size is larger than 7. This is because most of the final handover window size is within three seconds in our experiments. Large minimum window size will overuse the RF model in HYPER, thereby reducing the prediction accuracy. Considering that using a smaller S_{min} will introduce more computation overhead for stationary tests, we set $k = 5$ (i.e., 2.5s) in our experiments.

3) Impact of Mobility: Higher mobility scenarios will introduce more rapid link bandwidth changes and more frequent handover events. To investigate the prediction performance in different mobility scenarios, we use the data collected in three different mobility scenarios (i.e., stationary, walking, and driving). Figure 13(c) shows the bandwidth prediction errors in these scenarios. Results show that the overall prediction result under lower mobility scenarios is more accurate than that under driving scenarios. Interestingly, we can see that the median prediction errors in stationary scenarios are only slightly better than other dynamic scenarios. This is because the 5G link bandwidth is naturally highly dynamic due to varying numbers of UEs in the base station. We also observe that handover events occur even in the stationary scenario. This is because the signal indicator of the current serving cell may decrease due to many practical reasons, for example, some new UEs are accessing the cell. When the signal indicator of the current serving cell is lower than that of a neighboring cell, the handover procedure will be triggered.



(a) (b) (c) (d) (e)

Fig. 13: **Bandwidth prediction error of HYPER with different impact factors. Impact of (a) RF window size, (b) minimum handover window size, (c) mobility scenario, (d) time, and (e) carrier.**

TABLE II: **Prediction delay of HYPER.** *Time* **is the prediction delay of each phase in milliseconds.** *Prop* **refers to the proportion of each phase in the total trace.**

Scenarios \ Phases		P1	P2	P3	P4	AVE
Driving	*Time*	7.035	23.021	1043.937	15.281	62.103
	Prop	27.6%	9.7%	4.7%	58.0%	
Walking	*Time*	7.654	9.525	921.842	10.179	37.093
	Prop	16.4%	3.3%	3.0%	77.3%	
Stationary	*Time*	6.969	8.173	722.629	1.321	9.674
	Prop	5.6%	1.5%	1.1%	91.8%	

4) Impact of Time: To evaluate HYPER's performance at different times, we use two datasets collected in similar driving scenarios during the daytime and night. Figure 13(d) shows that the median bandwidth prediction errors of daytime and night are 11.9% and 9.5%, respectively. Results show that HYPER performs better at night. This is because there are fewer cellular users at night and the bandwidth will be less affected by other UEs in the same base station.

5) Impact of Carrier: We have also collected a small dataset (≈1000 samples) in nearby 5G cellular networks deployed by China Telecom to investigate whether HYPER can be used for another carrier network. Figure 13(e) shows that HYPER can achieve comparable bandwidth prediction accuracy in different carrier networks. This is because the 5G networks deployed by CMCC and China Telecom in the city have similar NR network architecture and base station density.

G. HYPER Overhead

As shown in Algorithm 1, the online prediction process can be divided into four phases: (1) Using the RF model within the minimum handover window during handover; (2) Running the ADF test and using the RF model outside the minimum handover window during handover where the time series is not stationary; (3) Estimating the ARMA parameters, training the ARMA model, and predicting using the ARMA model at the time when the time series becomes stationary; (4) Training the ARMA model, and predicting using the ARMA model for intra-cell time series. We insert timestamps in the code to estimate the prediction delay of each phase in different mobility scenarios. For each scenario, we run HYPER 100 times to calculate the average prediction delay.

Table II shows the proportion and prediction delay of each phase and the average prediction delays in three different mobility scenarios. The proportions of using the ARMA model and the RF model are different in different mobility scenarios. For low mobility scenarios, the ARMA model is preferred while for high mobility scenarios, the RF model will be used more. This is because higher moving speeds will lead to more frequent handover events, which will increase the frequency of using the RF model. Results also show that the prediction delays of most phases are within 30ms except for phase 3 due to the relatively computation-intensive ARMA parameter estimation process. In practical deployment, the delay of phase 3 can be significantly reduced if the ARMA parameters p and q of nearby base stations can be pre-measured. Comparing the average prediction delays of these three scenarios, the driving scenario is the worst because more time-consuming phases during handover (i.e., phase 2 and phase 3) will be included. However, the proportions of these time-consuming phases are less than 10% in all scenarios and will not introduce much prediction delay. The average prediction delays in all scenarios are less than 100ms, which is much smaller than the data collection interval in HYPER (i.e. 500ms) and is sufficient for online prediction.

During each time slot (i.e., 500ms), the collected data size is around 150 bytes and the returned data size is around 8 bytes. In total, the data transmission overhead is around 2.6 Kbps and is negligible for commercial 5G networks. According to [29], the energy cost is around 1.3 mJ/s in HYPER and will not significantly deplete the smartphone's battery. Note that because HYPER uses lightweight prediction models, it can also run entirely on common smartphones at the cost of slightly extra energy consumption.

VI. Implications and Lessons Learned

In this section, we give a summary of the key observations, implications, and lessons learned from our experiments.

Observation 1: Existing bandwidth prediction approaches cannot work well in 5G networks while HYPER achieves good prediction performance in all scenarios we have explored.

Implication 1: Although the prediction accuracy of HYPER is better than state-of-the-art approaches, there is still a gap compared to the actual bandwidth. Considering that cellular network performance may be affected by other commercially unobservable factors, such as radio resource scheduling and other UEs sharing the spectrum, the prediction accuracy can be largely improved with more detailed PHY-layer information exposed from the 5G chipset, such as the number of allocated resource blocks.

Observation 2: Handover can occur in stationary scenarios.

Implication 2: This indicates that the ARMA model may not work even in stationary scenarios. It is a general approach to integrate ARMA and RF models to cooperatively predict the bandwidth for different scenarios.

Observation 3: The environment has a large impact on prediction performance. HYPER performs diversely at different times due to the different numbers of UEs inside each base station and the different traffic of each UE.

Implication 3: When predicting at different times or places, it is suggested to use features collected in similar scenarios to train RF models. To reduce the training cost, we can integrate transfer learning [32] into HYPER for cross-site training.

Observation 4: Stale throughput information of past base stations can affect the prediction accuracy of learning models.

Implication 4: Handover information should be used to train effective learning models for each cell.

VII. Conclusion

In this paper, we first conducted a primary measurement study in commercial 5G networks to learn the temporal correlations of bandwidth time series and identify bandwidth-related features. We then propose HYPER, a hybrid bandwidth prediction approach that combines an ARMA model and an RF regression model to predict intra-cell and cross-cell bandwidth, respectively. We propose a handover window adaptation algorithm in HYPER to enable accurate bandwidth prediction in 5G networks with frequent handover events. Extensive experiments using COTS smartphones in commercial 5G networks show that HYPER outperforms state-of-the-art bandwidth prediction approaches. Results also show that HYPER can work well in various mobility scenarios without introducing much computation overhead.

Acknowledgements

This work is supported by NSFC under grant no. 62072396, Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars under grant no. LR19F020001, the Fundamental Research Funds for the Central Universities, and Alibaba-Zhejiang University Joint Institute of Frontier Technologies. Yi Gao and Wei Dong are the corresponding authors.

References

[1] A. Bousia, E. Kartsakli, L. Alonso, and C. Verikoukis. Dynamic energy efficient distance-aware base station switch on/off scheme for LTE-Advanced. In *Proc. of GLOBECOM*, pages 1532–1537. IEEE, 2012.

[2] G. E. Box, G. M. Jenkins, and G. C. Reinsel. *Time series analysis: forecasting and control*, volume 734. John Wiley & Sons, 2011.

[3] P. J. Brockwell, R. A. Davis, and S. E. Fienberg. *Time series: theory and methods*. Springer Science & Business Media, 1991.

[4] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer. A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques. *IEEE Communications Surveys & Tutorials*, 19(3):1790–1821, 2017.

[5] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson. Bbr: Congestion-based congestion control: Measuring bottleneck bandwidth and round-trip propagation time. *Queue*, 14(5):20–53, 2016.

[6] D. A. Dickey and W. A. Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, 74(366a):427–431, 1979.

[7] ENST. GPAC. http://gpac.wp.mines-telecom.fr/, 2020.

[8] W. A. Fuller. *Introduction to statistical time series*, volume 428. John Wiley & Sons, 2009.

[9] Google. Android 10. https://developer.android.google.com/about/versions/10, 2020.

[10] J. Jiang, V. Sekar, and H. Zhang. Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. In *Proc. of CoNEXT*, pages 97–108, 2012.

[11] Y. Jiang, D. C. Dhanapala, and A. P. Jayasumana. Tracking and prediction of mobility without physical distance measurements in sensor networks. In *Proc. of ICC*, pages 1845–1850. IEEE, 2013.

[12] K. Kousias, A. Pappas, O. Alay, A. Argyriou, and M. Riegler. Long short term memory networks for bandwidth forecasting in mobile broadband networks under mobility. *arXiv preprint arXiv:2011.10563*, 2020.

[13] J. Lee et al. Perceive: deep learning-based cellular uplink prediction using real-time scheduling patterns. In *ACM MobiSys*, 2020.

[14] T. Liu and A. E. Cerpa. Foresee (4c): Wireless link prediction using link features. In *Proc. of IPSN*, pages 294–305. IEEE, 2011.

[15] T. Liu and A. E. Cerpa. TALENT: Temporal adaptive link estimator with no training. In *Proc. of SenSys*, pages 253–266, 2012.

[16] R. Mittal, V. T. Lam, N. Dukkipati, E. Blem, H. Wassel, M. Ghobadi, A. Vahdat, Y. Wang, D. Wetherall, and D. Zats. Timely: Rtt-based congestion control for the datacenter. *ACM SIGCOMM*, 2015.

[17] A. Narayanan et al. Lumos5g: Mapping and predicting commercial mmwave 5g throughput. In *Proc. of IMC*, 2020.

[18] A. Narayanan, E. Ramadan, J. Carpenter, Q. Liu, Y. Liu, F. Qian, and Z.-L. Zhang. A first look at commercial 5g performance on smartphones. In *Proc. of WWW*, 2020.

[19] K. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot. Long-term forecasting of internet backbone traffic: Observations and initial models. In *Proc. of INFOCOM*, volume 2, pages 1178–1188. IEEE, 2003.

[20] J. Perktold et al. Statsmodels: Statistics in python. *Internet: http://www.statsmodels. org/devel/generated/statsmodels. tsa. stattools. adfuller. html [August 12, 2019]*, 2017.

[21] M. Polese, R. Jana, V. Kounev, K. Zhang, S. Deb, and M. Zorzi. Machine learning at the edge: A data-driven architecture with applications to 5g cellular networks. *IEEE TMC*, 2020.

[22] Qualcomm. Everything You Need to Know About 5G. https://www.qualcomm.com/invention/5g/what-is-5g, 2019.

[23] L. Ruan, M. P. I. Dias, and E. Wong. Machine learning-based bandwidth prediction for low-latency h2m applications. *IEEE IoTJ*, 2019.

[24] N. Sadek and A. Khotanzad. Multi-scale high-speed network traffic prediction using k-factor gegenbauer arma model. In *Proc. of ICC*, volume 4, pages 2148–2152. IEEE, 2004.

[25] M. Shafi et al. 5g: A tutorial overview of standards, trials, challenges, deployment, and practice. *IEEE JSAC*, 35(6):1201–1221, 2017.

[26] S. Sur, I. Pefkianakis, X. Zhang, and K.-H. Kim. Wifi-assisted 60 ghz wireless networks. In *Proc. of MobiCom*, pages 28–41, 2017.

[27] W. Wanalertlak, B. Lee, C. Yu, M. Kim, S.-M. Park, and W.-T. Kim. Behavior-based mobility prediction for seamless handoffs in mobile wireless networks. *Wireless networks*, 17(3):645–658, 2011.

[28] X. Xie, X. Zhang, S. Kumar, and L. E. Li. pistream: Physical layer informed adaptive video streaming over lte. In *Proc. of MobiCom*, 2015.

[29] D. Xu et al. Understanding Operational 5G: A First Measurement Study on Its Coverage, Performance and Energy Consumption. *ACM SIGCOMM*, 2020.

[30] Q. Xu, S. Mehrotra, Z. Mao, and J. Li. Proteus: network performance forecast for real-time, interactive mobile applications. In *Proc. of MobiSys*, pages 347–360, 2013.

[31] S. Xu, A. Nikravesh, and Z. M. Mao. Leveraging context-triggered measurements to characterize lte handover performance. In *PAM*, 2019.

[32] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Proc. of NIPS*, 2014.

[33] C. Yue, R. Jin, K. Suh, Y. Qin, B. Wang, and W. Wei. Linkforecast: cellular link bandwidth prediction in lte networks. *IEEE TMC*, 2017.

[34] Y. Zaki, T. Pötsch, J. Chen, L. Subramanian, and C. Görg. Adaptive congestion control for unpredictable cellular networks. In *Proc. of SIGCOMM*, pages 509–522, 2015.

[35] C. Zhang, X. Ouyang, and P. Patras. Zipnet-gan: Inferring fine-grained mobile traffic patterns via a generative adversarial neural network. In *Proc. of CoNEXT*, pages 363–375, 2017.

[36] B. Zhou, D. He, Z. Sun, and W. H. Ng. Network traffic modeling and prediction with arima/garch. In *Proc. of HET-NETs*, pages 1–10, 2005.