On Medium and Long Term Channel Conditions Prediction for Mobile Devices

Xiaoli Wang*, Edward Grinshpun†, David Faucher†, Sameer Sharma†
*Princeton University, Princeton, NJ, USA; xw4@princeton.edu
†Nokia Bell Labs, Murray Hill, NJ; {edward.grinshpun, david.faucher, sameer.sharma}@nokia-bell-labs.com

***Abstract*—In estimating the expected wireless link channel condition for real-time mobile applications, current wireless heat maps cannot provide the required precision. In this paper we introduce a new set of LTE 'channel condition' maps that uniquely enable the radio access network to predict channel conditions of the mobile device with required accuracy. We show how to construct these channel conditions maps via machine learning without extensive driving tests. Finally we demonstrate high accuracy of wireless channel condition prediction for mobile users using our channel condition maps in conjunction with ARMAX and neural networks techniques.**

***Keywords*—channel conditions, prediction, wireless heat maps.**

I. Introduction

Knowing future channel conditions for mobile User Equipment (UE) is one of the two key ingredients (the other is the scheduler-allocated wireless resources at the serving cell) in predicting wireless link throughput. Estimating future channel condition is of great interest in both academia and industry, since it can significantly benefit adaptive mobile applications, especially video. In particular, accurate medium term (100 milliseconds to seconds) predictions can benefit live video chat and TCP throughput optimization, while long term (seconds to tens of seconds) predictions can benefit adaptive streaming video.

Current wireless heat maps, typically represented as color coded geographic maps showing levels of wireless signal quality [1] [2], do not provide adequate precision to enable accurate prediction of wireless link throughput. The main reasons are: (a) these maps do not reflect the high variation in signal quality due to varying levels of interference from neighboring cells, and (b) traditional wireless channel metrics like Signal to Interference and Noise Ratio (SINR) or Relative Signal Strength Indicator (RSSI) are too detached from throughput computation.

In this paper, we use the channel quality metric introduced in [3] to represent 'bits per physical resource block without retransmissions', which we call here 'TBS slope'(TBS: transmission block size). Based on this, we introduce a new kind of LTE per cell/sector channel condition maps, where each map consists of a set of geographic pixels and is parameterized by a vector of discretized neighbor cell loads, which we found to be a suitable representation of interference from neighbor cells. The map contains the following 3-tuple for each pixel: average TBS slope, prediction error margin, and the probability of UE being served by this cell/sector.

Our results show a wide spread of TBS slope for the same UE location under different neighbor cell loads. This leads to the conclusion that any long term UE based channel condition predictions (including the popular crowdsourced channel condition maps) are inadequate due to the inherent lack of the knowledge of expected interference. Therefore, we propose a set of channel condition maps that can vary over time, and present a methodology to construct these channel condition maps by the LTE Radio Access Network (RAN) via machine learning, without extensive driving tests. Finally, we analyze the impact of UE mobility and demonstrate high accuracy (90% or better) of TBS slope prediction when combining these channel condition maps with ARMAX (autoregressive-moving-average model with exogenous inputs) and neural networks techniques.

The rest of the paper is organized as follows. Section II gives a brief overview of related work. Section III describes our methodology for constructing channel condition maps and prediction techniques. Section IV describes our wireless network simulation setup and configuration. Section V presents our prediction results, and Section VI concludes the paper.

II. Related Work

Throughput prediction and channel condition estimation have been extensively studied in academia, with predictions from both network side and client side. However, existing approaches have limitations on either prediction accuracy or user movement.

A. Network-side Prediction

Channel condition prediction based on stochastic models has been well studied [4]–[6], but most of these predictions focus on short-term predictions (in the level of ms) of fading channels using time series analysis, like autocorrelation [7] and clustering in time domain [8]. For long term predictions, even though some models still work in theory [4], [6], in practice, the prediction has high errors (more than 20%) and fails to be used by real-time applications. SINR and RSSI are the metrics that are used for prediction in most existing research [9], [10], however, we find them too detached from the computation of averaged link throughput.

B. Client-side Prediction

An emerging approach for low-cost throughput prediction is crowdsourced throughput inference (e.g. OpenSignal app [2]), which builds a signal quality map by collecting samples, such as RSRP (Reference Signal Receive Power), RSRQ (Reference Signal Received Quality), from distributed mobile devices [11]. However, such signal quality map is only location-based, which cannot reflect the throughput variation over time (e.g. due to varying interference from neighbor cells). For time-dependent throughput estimation, people either propose active measurements of throughput via delay, packet loss [12] and the ratio of payload and transmission time [13], or collect passive measurements together with cell load inference [14]. However, all these works assume the channel only changes slightly over time, where mostly stationary users are considered, so the throughput in the next few seconds can be predicted based on current throughput. [15] considers moving users, and analyzes traffic patterns and predicts future throughput on a fixed route in the time domain on hourly basis. The hourly prediction can be useful for general analysis of traffic patterns, but the prediction granularity is too coarse for real-time video applications.

III. Our Solution and Methodology

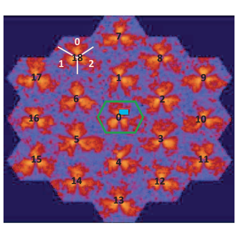
In this section, we present our channel condition maps, and propose an architecture as well as a prediction method for LTE Radio Access Network (RAN) to construct such maps via machine learning.

A. Channel Conditions Maps

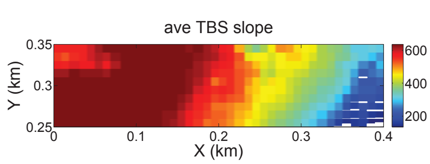
Geographic map is divided into uniform rectangular areas, which we call pixels. Details of the pixel size selection is outside the scope of this paper. Each pixel falls in a coverage area of one or more serving cell/sectors (Fig. 1a. The white 0, 1, 2 indicate orientation of three sectors in a cell.)

For each pixel X there is a list of 5-tuple $< S, P_{S,X}, L_S, M_{S,X}, E_{S,X} >$, where S is a serving cell/sector, $P_{S,X}$ is probability of a UE in the pixel X to be served by S, L_S is a vector load of 'significant' neighboring cells (see definition and details in Section V-B), $M_{S,X}$ is average value of channel metric for X when served by S, and $E_{S,X}$ is standard deviation of the channel metric within X.

To calculate M we use the metric introduced in [3], which we will refer to as "TBS slope" - the average number of useful (excluding retransmissions) bits per second per LTE physical resource block (PRB), expressed as $\overline{\alpha_k} = \sum_j \alpha_{kj}(\frac{n_{kj}}{n_k})$. Here, n_{kj} is the number of useful PRBs allocated to bearer k at MCS(modulation and coding scheme) j; $n_k = \sum_j n_{kj}$ is the total number of PRBs allocated to bearer k; α_{kj} is the number of bits per PRB for the bearer k at MCS j. The LTE scheduler of the serving cell computes TBS slope based upon PRB distribution per MCS over short bin time intervals (e.g. 1 sec or 100 msec).



(a) 19 cell SINR heat map



(b) TBS slope heat map for a zoomed-in blue 10pixel x 40pixel rectangle with pixel size 10m x 10m, computed for neighbor cell load 50%.

Fig. 1: SINR and TBS slope heat map.

As a measure of cell/sector load, we use average percent of aggregate PRBs allocated per second. For each serving cell S, the loads of 'significant' (see Section V-B) neighbors form an ordered vector load L_S. The vector loads L_S are discretized, based upon preconfigured ranges of load for each set of included neighbor cell/sectors, into K intervals denoted as a finite set $\{L_S^K\}$.

For each load $L \in \{L_S^K\}$ and each serving cell/sector S we define a cell channel conditions map as collection of pixels X with the associated 3-tuple $< M_{L,S,X}, P_{L,S,X}, E_{L,S,X} >$. The procedure of constructing the channel condition maps is outlined in Alg. 1.

Algorithm 1 Channel Conditions Map construction

1: For each cell sector S, select N neighboring cell sectors that have major impact, denoted as S_i $(i \in N)$
2: Discretize the load of each cell S_i by equally dividing the load into K bins ($\frac{1}{K}$ granularity), as $L_{S_i}^{k_i}$ $(k_i \in K)$.
3: Construct a unique map for each vector load $L_S = \{L_{S_1}^{k_1}, L_{S_2}^{k_2}, ..., L_{S_N}^{k_N}\}$, where each vector $\{k_1, k_2, ..., k_N\}$ denotes the discretized individual cell sector loads in the range $[\frac{1}{K}, \frac{2}{K}, ..., 1]$. There is $N \times K$ maps for each cell sector.
4: Discretize each map into pixels and store the three historical pixel metrics $< M_{L,S,X}, P_{L,S,X}, E_{L,S,X} >$.

B. Architecture: Maps Assembly via Machine Learning

Fig. 2 presents a high level view of the architecture. Each cell/sector periodically (e.g. every 1 sec) bulk-reports the following time synchronized information to the data analytics engine: average aggregated cell load, average TBS slope per UE/bearer and per location pixel. The data analytics engine uses this information to construct the channel condition maps. The maps are further used by prediction function as described in Section III-C. The required UE location accuracy is dictated by the pixel size, and may be reported by the UE or alternatively computed by the RAN using well known triangulation techniques (e.g. [16]).

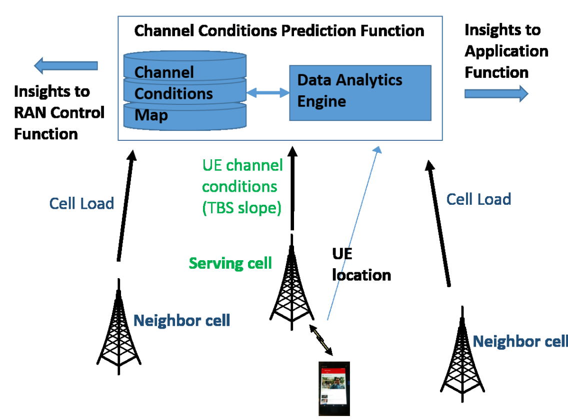


Fig. 2: Map assembly.

The data analytics engine performs the following two functions using the information above: (1) learn the channel condition maps, compute the average channel condition metrics for the pixels in each cell sector and update the Channel Condition Map database; (2) predict the channel condition metrics for served UEs based on the Channel Condition Map metrics (specifically TBS slope) in the selected location pixels and the historical metrics values. In this paper, we focus on TBS slope prediction, assuming that we know the predicted UE location, for example, via GPS report from mobile phones (accuracy of 7.8m [17]), or using the UE location prediction techniques (e.g. [16], [18]). Such channel condition prediction can benefit both RAN control and application functions. RAN control function can use channel condition maps to identify coverage gaps and enable a variety of Self Organizing Networks (SON) techniques, such as facilitating the SDN controller to dynamically readjust the programmable antenna arrays and/or balance the maximal load of cells. Bandwidth demanding mobile applications (e.g. video) can use TBS slope prediction to proactively and gracefully adapt to varying wireless conditions, resulting in significant improvement in the quality of experience for end users.

C. Prediction Methodology

Our prediction is based on a historic database of channel condition maps constructed as outlined in Section III-A. Our database stores the average TBS slope and the error margin, so these values can directly be used in predicting channel conditions for stationary UEs when the user location and neighboring cell loads are known. For stationary UEs (Section V-A) we also apply time series analysis ARMA to capture the variation of channel conditions in the temporal domain. As to moving UEs, our analysis shows a memory effect of channel conditions (details in Section V-C) that depends on UE moving directions, so the stored metric in the database for stationary UEs cannot be directly applied to moving UEs. To tackle this problem, we propose an ARMAX model for medium term and a neural network based function fitting method for long term prediction. Both methods are based on the channel condition maps constructed with stationary UEs' data, and we train parameters that can further add the moving UE memory effect pattern to the model.

1) Stationary user – ARMA: Using UE location and neighboring cell loads, we select the corresponding channel condition map and find the average TBS slope value matched to the UE pixel location. Using ARMA (Eqn. 1), we do prediction for the next $1s$, next $5s$ and next $10s$. For the next $1s$ (medium term) prediction, we directly use our historic TBS slopes averaged over $1s$ intervals and feed that into the ARMA model. For prediction for the next $5s$ and $10s$ (long term), we use moving average of historical values ($X_{t\_MA} = \frac{1}{n}\sum_{k=0}^{n-1} X_{t-k}$, MA of n steps) with steps of $5s$ and $10s$, respectively, in order to smooth the data by filtering out the noise.

$$X_t = \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + \beta_1 \varepsilon_{t-1} + \cdots + \beta_q \varepsilon_{t-q}. \quad (1)$$

2) Moving user (medium term prediction) – ARMAX: We first predict user location for the next second, and then obtain the corresponding cell load information from the eNB. Given the user moving direction and speed, and our estimation of future locations and neighboring cell load, we want to predict the TBS slope for the next 1s. We propose an ARMAX based model to capture the change of TBS slope between two adjacent seconds. We treat the average TBS slope for stationary users at a specific location pixel as the exogenous input, i.e. 'X' in ARMAX. Our ARMAX model is expressed as Eqn. (2), where B is the 'X' term.

$$X_t = \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + \beta_1 \varepsilon_{t-1} + \cdots + \beta_q \varepsilon_{t-q} + \gamma B. \quad (2)$$

Our prediction is for the next 1s, with training for the previous 15s for the same moving user. As the user direction/speed changes while the user is moving, the model parameters α, β, γ will be updated as the model will be retrained using the latest historical 15s.

3) Moving user (long term prediction) – function fitting: As ARMAX is a linear prediction method, it cannot fully capture the channel condition pattern in a long prediction window, if the pattern is nonlinear. Therefore, we propose a more complex model – a neural network (NN) based function fitting method [19] to extract TBS slope changing pattern while users are moving, and then use this model to predict TBS slope for future users going through the same area.

Instead of directly fitting a function for the TBS slope with moving users, we use the difference between TBS slope, i.e. $delta = X_m - X_s$, where X_m and X_s are average TBS slopes for moving and stationary users, respectively. While fitting the function, we use the user location pixel as the input, and the corresponding $delta$ value as the output. The training values are based on users moving horizontally at the speed of 30km/h. When a user comes with a different speed, we can still use the same model by mapping this user's location to the input. In terms of the NN model, we use 10 hidden layers, and the performance is evaluated using mean square error (MSE).

IV. Simulation Setup

We simulate a 19-cell LTE system (Fig. 1a), each with three sectors, using Nokia sample level LTE simulator with cell 0 being in the center, tier 1 consisting of six cells surrounding cell 0, and tier 2 consisting of additional twelve cells surrounding tier 1 (Fig. 3). To simulate the channel conditions that mobile devices experience, we vary the load

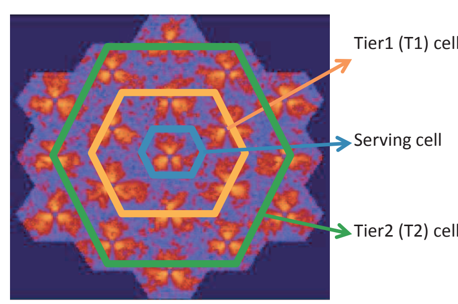


Fig. 3: Categorize the 19 cells into three tiers.

of cells, user locations and user speed. Then we record the SINR, MCS, TBS, PRBs and time stamp of each user every TTI (transmission time interval, 1ms), which is then used to calculate the metrics.

Center cell 0 sector 0 is our target cell/sector where users move inside the blue rectangle (Fig. 1a). We vary load of the remaining 18 cells in granularity of 0.25 and 0.1 (25% and 10% total PRB utilization) to characterize the impact of different neighboring cell loads on users in cell 0. For each simulation, we choose a neighboring cell load variation and a user direction/speed, and the simulation is run for 2800s, where the first 2000s serve as the training data to construct channel condition maps, and the remaining 800s serve as the testing data. For stationary users, each simulation is for one pixel, where 25 users are evenly distributed in each pixel and the average TBS slope of these users are treated as the TBS slope for that pixel. In simulations for moving users, for each neighboring cell load variation, we place 25 users in one pixel, and they move together with the same moving speed and direction. The average of users passing across the same pixel is taken as the average TBS slope for moving users in that pixel. This value is treated as the ground truth of the TBS slope for moving users, and the predicted values from our prediction approach is compared with this value for error analysis. The salient simulator setting and configurable parameters are presented in Table. I and Table. II.

We specify UE path, default mobile speed and other cell load to vary the UE location, speed and load of neighboring cells, respectively. A path consists of a sequence of points through which the UE will traverse. Its speed determines how long it will take to traverse from point to point. The other cell/sector load parameter generates interference for the UEs of interest. It is specified on a per cell/sector basis and consists of the ratio of occupied PRBs to bandwidth (e.g., a 10MHz system bandwidth with a load of 1.0 would result in generated interference as if all 50 PRBs per TTI were in use).

V. Results and Analysis

In this section, we analyze the effect of neighboring cell loads and evaluate the performance of our channel condition prediction approach for both stationary and moving UEs. We use mean absolute percentage error (MAPE) (Eqn. (3)) to evaluate the prediction accuracy. Here, P_k is the predicted value, A_k is the actual value, and n is the total number of testing samples.

$$\frac{1}{n}\sum_{k=1}^{n}\frac{|P_k - A_k|}{A_k} \times 100 \tag{3}$$

A. Time Domain Prediction (stationary UE)

We compare prediction errors with (using ARMA) and without (using TBS slope mean for each pixel) accounting for the temporal channel condition variation for stationary UEs with different granularity of measurement (1s, 5s and 10s). The results with 75 users (Table III with all neighboring cell loads set to 0.5 (50% PRB utilization)) show only a slight improvement when using ARMA. Since using ARMA incurs high computation cost and requires more training data, in real systems, we suggest simply using prediction with mean.

With both approaches the prediction error decreases as a longer interval is taken, since smoothing the data over longer time interval averages out the temporal variation of the channel conditions, which is similar to white noise.

B. Impact of Neighboring Cell Load

We categorize the cells into three tiers based on their distance to the serving cell, as shown in Fig. 3, where T1 and T2 are two tiers of neighboring cells. We vary the load of T1 cells in granularity of 0.1, the load of T2 cells in granularity of 0.25, and then measure the average TBS slope and corresponding prediction MAPE error of a stationary user within the serving cell. From Fig. 4, we observe that when the load of T2 cells are fixed, the average TBS slope of the target user increases as the load of T1 cells decreases. The same trend applies to the case with fixed T1 cell load but varying T2 cell load. In addition, we can infer from Fig. 4 that T1 cells have larger impact than T2 cells, as users in the serving cell experience steeper drop of the average TBS slope when T1 cell load increases. This is reasonable because the interference from neighboring cells is negative correlated to the distance between the serving cell and the neighboring cell.

To reduce the complexity in building the map database, we want to minimize the number of neighboring cells that need to

Carrier Frequency (DL)	700 MHz
Carrier	FDD, 10 MHz bandwidth (50 PRBs per TTI)
Cell Distance	2 km between cell centers
UE Receive mode	2 Rx Antennas
CQI reporting	periodic every 20 m-sec

TABLE I: The salient simulator settings

Rayleigh Channel Model	ITU.EPA Pedestrian A, pedestrian B, or Vehicular A
Default Mobile Speed	configurable as X km/hr
Other Cell/Sector Load	ratio of occupied PRBS to bandwidth
UE Path	determine location of UE throughout the simulation
Simulation Duration	specified in seconds

TABLE II: Configurable simulator settings

Prediction interval	1s	5s	10s
MAPE(ARMA prediction)	5.85%	2.69%	1.96%
MAPE(mean prediction)	6.12%	3.31%	2.44%

TABLE III: Temporal Prediction Error

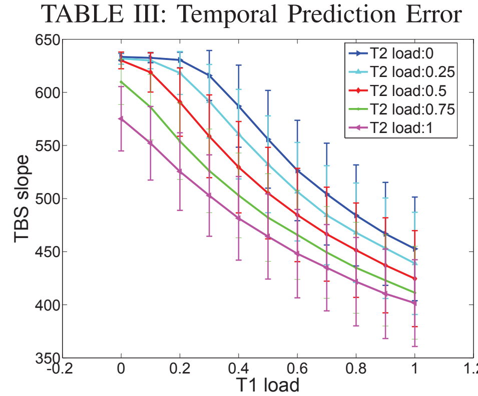
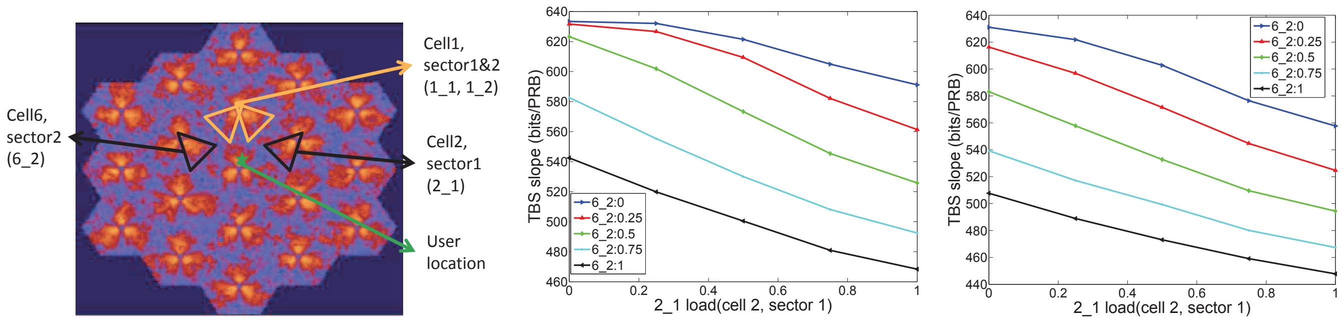


Fig. 4: TBS slope with varying T1, T2 load



(a) Selected sector (b) Sectors 1_1, 1_2 no load (c) Sectors 1_1, 1_2 full load

Fig. 5: Impact of cell/sector load variations inside T1 cells

be considered. Here, we analyze the impact of different cells in T1 by choosing four cells in T1, as defined in Fig. 5a. The two cells (6_2, 2_1) in black are the two cells facing directly to the serving cell, but the other two cells (1_1, 1_2) in yellow are not. Based on the comparison of zero load (Fig. 5b) and full load (Fig. 5c) of the two yellow cells, the average TBS slope only slightly decreases as the load increases from zero to one, thus these cells are not 'significant' for the map database. In contrast, for the two cells in black, the average TBS slope decreases dramatically as the cell load increases, so these two cells can be treated as the main contributor of interference.

C. Impact of User Direction/Speed (Moving UE)

In our simulation, users move horizontally and diagonally at the speed of 30km/h, 60km/h and 90km/h. The traces of moving users are shown in Fig. 6. These traces cover areas with varying channel conditions, based on which, we analyze the change of channel conditions with users moving through pixels.

For the scenario of users moving from good channel condition to bad channel conditions (Fig. 7), moving users experience higher TBS slope than stationary users. This is because the better channel condition from those pixels on the left contribute to the average TBS slope as users moving from left to the center, which can be explained as a memory effect for moving users.

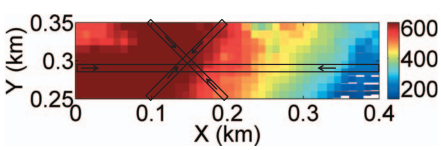
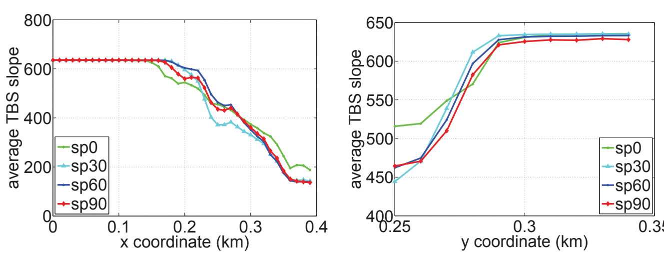


Fig. 6: The traces of users moving horizontally and diagonally in the 10×40 pixel area (TBS slope heat map).



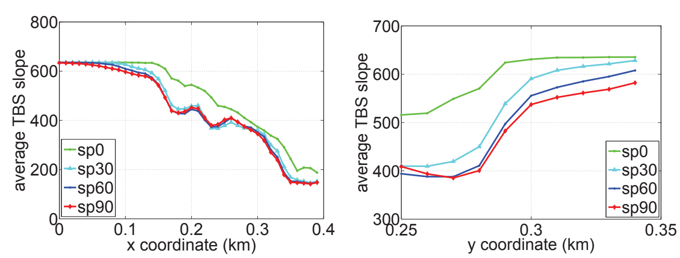
(a) left to right (b) top left to bottom right

Fig. 7: User moving from good to poor channel condition.

Conversely, when users move in the reverse direction of these traces (Fig.8), moving users will experience lower TBS slope than stationary users. This further confirms that the memory effect we observe in the scenario of moving from good to poor channel conditions also applies to the scenario of moving from poor to good channel conditions. Furthermore, moving users speed does not appear to have a significant impact on TBS slope (8% or less difference for 30km/h vs. 90km/h, and in most cases 4% or less).

D. Medium Term Prediction of Channel Conditions (Moving UE))

The ARMAX model is trained for the first 15s of a user moving horizontally, and the model is applied to the remaining data of the moving user in predicting the TBS slope for the next 1s with historical TBS slope information for the previous 5s. When the user speed changes, a new model will be trained for 15s. The prediction error based on ARMAX (75 test users) is presented in Table IV and Fig. 9a. Our ARMAX results are compared with pixel mean, which simply calculates the average TBS slope for stationary users at a pixel location (i.e. 'X' term in our ARMAX model), as the case for the stationary users. Based on the results, we observe that the ARMAX-based approach significantly reduces the error (to less than 10%) compared to pixel mean, due to the memory effect of moving users discussed in the previous section.

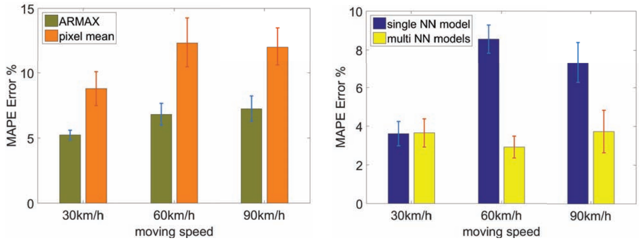


(a) right to left (b) bottom right to top left

Fig. 8: User moving from poor to good channel condition.

Moving Speed	30km/h	60km/h	90km/h
ARMAX	5.23%	6.81%	7.24%
Pixel Mean	8.79%	12.37%	12.06%

TABLE IV: Moving User Medium Term Prediction Error



(a) medium term (b) long term

Fig. 9: Prediction error.

Moving Speed	30km/h	60km/h	90km/h
MAPE(single NN model)	3.62%	8.56%	7.34%
MAPE(multi NN models)	3.65%	2.92%	3.73%

TABLE V: Moving User Long Term Prediction Error

E. Long Term Prediction of Channel Conditions (Moving UE))

We use a neural network model with 10 layers of hidden network, which is trained in the space domain, with training data of one user (single NN model) moving horizontally from left to right at 30km/h. The input is the user location pixel, and the output is the TBS slope difference between moving users and stationary users. Then the training model is used to predict other users moving at 30km/h, as well as users moving at 60km/h and 90km/h, when they pass through the same area. The prediction is for the TBS slope in the entire time interval when the user passing through the same area. In our test pixel locations in Fig. 6, it is 47s for 30km/h, 24s for 60km/h and 16s for 90km/h. We tested over 75 users, and the average error is presented by the blue bar in Fig. 9b and Table V. For other users moving at 30km/h (the same speed as training data), the error is around 3.6%. For users moving at other speeds (60km/h and 90km/h), the error increases to 8%. When training is done for each user speed (multi NN model), the error is reduced to around 3% (yellow bar in Fig. 9b).

Comparing the results from ARMAX model and the neural network model, we find that neural network model is more accurate, and it can also be used for the medium term prediction. This is because the neural network model better captures the nonlinear relation among pixels on the moving user trace, while ARMAX only capture the linear relation between two adjacent pixels for the past trace and there is a chance that the pattern between two adjacent pixels will be different for future locations. However, the higher accuracy of neural network is at the cost of more complexity, as the neural network model needs to be trained on the same trace before the model can be applied to a user moving along this trace. In contrast, ARMAX model can be trained online when a user moves along a new trace, based on the channel condition of this user in the past 15s. Also, the computation cost of neural network training is higher than ARMAX training. Therefore, based on the accuracy requirement, an application will decide which model to be used in the prediction.

VI. Conclusions

In this paper we introduce a new set of LTE 'channel condition' maps that uniquely enable the radio access network to predict UE channel conditions with required accuracy. We present an architecture and methodology of assembling these channel conditions maps via machine learning. Finally we demonstrate high accuracy of wireless channel condition prediction for mobile users using our channel condition maps in conjunction with ARMAX and neural networks techniques, with resulting prediction accuracy of 90% or better.

References

[1] G. Judd, X. Wang, and P. Steenkiste, "Efficient channel-aware rate adaptation in dynamic environments," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pp. 118–131, ACM, 2008.

[2] "Open signal coverage map."

[3] Z. Sayeed, E. Grinshpun, D. Faucher, and S. Sharma, "Long-term application-level wireless link quality prediction," *Proceedings of 36-th IEEE Sarnoff Symposium*, pp. 1–6, 2015.

[4] N. Bui, F. Michelinakis, and J. Widmer, "A model for throughput prediction for mobile users," in *European Wireless 2014; 20th European Wireless Conference; Proceedings of*, pp. 1–6, VDE, 2014.

[5] A. Duel-Hallen, "Fading channel prediction for mobile radio adaptive transmission systems," *Proceedings of the IEEE*, vol. 95, no. 12, pp. 2299–2313, 2007.

[6] M. Malmirchegini and Y. Mostofi, "On the spatial predictability of communication channels," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 3, pp. 964–978, 2012.

[7] L. Thiele, M. Olbrich, M. Kurras, and B. Matthiesen, "Channel aging effects in comp transmission: gains from linear channel prediction," in *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, pp. 1924–1928, IEEE, 2011.

[8] V.-H. Pham, X. Wang, J. Nadeau, and J.-Y. Chouinard, "Cluster-based time-domain channel prediction for dynamic wireless communications," in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, pp. 59–63, IEEE, 2010.

[9] A. Duel-Hallen, H. Hallen, and T.-S. Yang, "Long range prediction and reduced feedback for mobile radio adaptive ofdm systems," *Wireless Communications, IEEE Transactions on*, vol. 5, no. 10, pp. 2723–2733, 2006.

[10] M. Ni, X. Xu, and R. Mathar, "A channel feedback model with robust SINR prediction for LTE systems," in *Antennas and Propagation (EuCAP), 2013 7th European Conference on*, pp. 1866–1870, IEEE, 2013.

[11] K. Evensen, A. Petlund, H. Riiser, P. Vigmostad, D. Kaspar, C. Griwodz, and P. Halvorsen, "Mobile video streaming using location-based network prediction and transparent handover," in *Proceedings of the 21st international workshop on Network and operating systems support for digital audio and video*, pp. 21–26, ACM, 2011.

[12] Q. Xu, S. Mehrotra, Z. Mao, and J. Li, "PROTEUS: network performance forecast for real-time, interactive mobile applications," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pp. 347–360, ACM, 2013.

[13] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck, "An in-depth study of LTE: Effect of network protocol and application behavior on performance," in *ACM SIGCOMM Computer Communication Review*, vol. 43, pp. 363–374, ACM, 2013.

[14] A. Chakraborty, V. Navda, V. N. Padmanabhan, and R. Ramjee, "Coordinating cellular background transfers using loadsense," in *Proceedings of the 19th annual international conference on Mobile computing & networking*, pp. 63–74, ACM, 2013.

[15] H. Konishi, K. Kanai, and J. Katto, "Improvement of throughput prediction accuracy for video streaming in mobile environment," in *Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference on*, pp. 749–750, IEEE, 2014.

[16] V. Osa, J. Matamales, J. F. Monserrat, and J. López, "Localization in wireless networks: The potential of triangulation techniques," *Wireless personal communications*, vol. 68, no. 4, pp. 1525–1538, 2013.

[17] "US govenment information about GPS accuracy."

[18] R. M. Vaghefi and R. M. Buehrer, "Improving positioning in LTE through collaboration," in *Positioning, Navigation and Communication (WPNC), 2014 11th Workshop on*, pp. 1–6, IEEE, 2014.

[19] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.