CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Harnessing collaboration to improve the accuracy of throughput prediction in cellular networks
Authors	Raca, Darijo;Zahran, Ahmed;Sreenan, Cormac J.;Tiwari, Abhishek;Gupta, Riten
Publication date	2026-02-19
Original Citation	Raca, D, Zahran, A, Sreenan, C J, Tiwari, A & Gupta, R 2026, Harnessing collaboration to improve the accuracy of throughput prediction in cellular networks. in K Erenli, C Guetl, Y Jararweh & J Jansen (eds), 2025 3rd International Conference on Foundation and Large Language Models, FLLM 2025. 2025 3rd International Conference on Foundation and Large Language Models, FLLM 2025, Institute of Electrical and Electronics Engineers Inc., pp. 1284-1289, 2025 3rd International Conference on Foundation and Large Language Models, FLLM 2025, Vienna, Austria, 25/11/25. https://doi.org/10.1109/FLLM67465.2025.11391126
Type of publication	Conference item
Link to publisher's version	https://www.scopus.com/pages/publications/105035919796 - 10.1109/FLLM67465.2025.11391126
Rights	© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Download date	2026-11-03 03:43:06
Item downloaded from	https://hdl.handle.net/10468/18790



University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Harnessing Collaboration to Improve the Accuracy of Throughput Prediction in Cellular Networks

Darijo Raca
Faculty of Electrical Engineering
University of Sarajevo
Bosnia and Herzegovina
Email: draca@etf.unsa.ba

Ahmed Zahran, Cormac J. Sreenan
School of Computer Science & IT
University College Cork
Cork, Ireland
Email: firstname.lastname@ucc.ie

Abhishek Tiwari, Riten Gupta
Meta Platforms Inc.
Menlo Park, CA, USA
Email: atiwari,ritengupta@meta.com

***Abstract*—Throughput prediction in cellular networks has garnered considerable interest in recent years due to its demonstrated positive impact on quality of experience. Existing proposals operate by having each user device make its own predictions, in a standalone manner, on the basis of its local measurements. Our hypothesis is that pooling of device measurements in a collaborative way can yield more accurate predictions, by allowing a broader set of observations from within a cell to be combined. To this end, we identify shortcomings in existing datasets, and then present our collaborative approach, along with an extensive evaluation. When compared to operating standalone, the results show a reduction in prediction error of up to 66% for users that have been inactive, and up to 17% for active users.**

***Index Terms*—Throughput prediction, cellular networks, collaborative, measurement, machine learning**

I. Introduction

Mobile devices using cellular networks operate in a highly dynamic environment in which the quality of the wireless communication link varies continuously. This is due to a range of factors, including, inter alia, those related to the physical layer behaviour of the wireless link; the operation of the base station, which schedules access to the medium and configuration of antennas; the demands of other users sharing the cell; and, fading effects due to user mobility. While best-effort applications can work reasonably well in adapting to such variability, other applications struggle in providing a suitable user quality of experience (QoE), notably for resource-demanding video streaming, networked gaming and augmented reality. This observation has motivated several studies seeking to offer accurate and timely inference for throughput prediction in cellular networks. These include papers proposing novel techniques such as [1]–[4], papers related to video QoE [5]–[7], and experiences of implementing Throughput Prediction (TP) in real-world trials [8].

TP represents a time series problem in which a historical window (the *history*) of metrics is used to predict the future throughput for a target *horizon*. The length of the history and horizon windows depends on a number of factors, including the application needs and the dependency between history metrics and future throughput. For example, the throughput horizon for video-on-demand streaming would be of the order of 4-10 seconds depending on encoding parameters, since video adaptation algorithms make their download decisions for video segments that correspond to a fixed playback duration. A shorter horizon, e.g. 0.5-2 seconds, would be considered for time-sensitive applications, in which throughput impacts short-term decisions, such as encoding in real-time conferencing or sensing/actuation in industrial applications. In *univariate* models, the historic metrics are typically the recorded throughput samples. Advanced prediction approaches, such as machine learning, can also integrate other metrics, such as from the wireless physical layer (PHY) and other context information such as mobility/speed, leading to the development and use of *multi-variate* models.

Machine learning models for TP leverage both traditional and deep learning algorithms, e.g. [1], [9], [10]. These train a model based on measurement datasets collected from mobile network simulations or so-called ”drive tests”, where laptops or smartphones are used over a period of time to record various metrics whilst actively downloading. It is common in these tests to use large file downloads, but recently the shortcomings of subsequently using these models for TP in scenarios with more dynamic traffic, such as video, has been identified [8]. Once the model is trained, its TP accuracy is evaluated for each mobile device, using its local measurements.

The aforementioned solutions use what we term as *standalone* inference/prediction engines. These provide predictions at each end-user device, relying only on that device's metrics to predict its future throughput. Such an approach takes advantage of being able to collect metrics locally and utilise them with knowledge of the intended application. We propose an innovative *collaborative* approach, in which data from different users is combined to predict future throughput. By taking advantage of greater amounts of data and from a wider variety of devices and contexts we expect to be able to train a more robust model and more accurate predictions. We are motivated by the observation that it is common for many users to be active concurrently within a cell, and typically engaged in video streaming using one of a small set of popular applications, thus broadening the base of measurements for

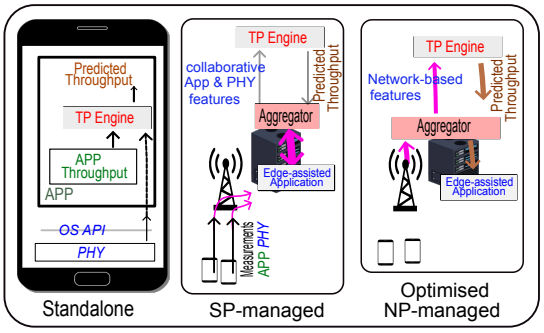


Fig. 1. Illustration for standalone, SP managed, and NP managed collaborative prediction. For collaborative prediction, only edge-assisted applications is shown.

similar activities. We also note that collaboration benefits users who move from an inactive state to an active one, as they would otherwise be operating without a relevant recent history of samples. Distinct from our focus on collaborative inference, the use of federated learning across multiple devices has been proposed for efficient and privacy-preserving model update [11].

Our research is also motivated by the shift towards edge-assisted applications, where many application functions will be offloaded to the edge. This shift implies that a service provider would have the capability to implement offloaded functions towards which users can supply decision-relevant data. The contributions of this work include:

- Collaborative TP as a novel approach, using data from different users to improve the prediction accuracy in various scenarios with respect to user activity and prediction horizon.
- An innovative prediction engine design to eliminate wireless communication overhead in the case of TP that is orchestrated by a Network Provider (NP).
- Sharing our insights on collecting system-level datasets that capture metrics from multiple users sharing the same cell.

Our results show that collaborative prediction reduces the prediction error up to 66% for users that have been inactive, and up to 17% for active users.

Section II presents our collaborative solution, including optimisations. Section III explains dataset considerations followed by the performance evaluation Section IV.

II. Collaborative Throughput Prediction

In this section we firstly present the design of our collaborative TP model, followed by a discussion of some optimisations.

A. Collaborative Model Design

In device-based TP, the prediction relies solely on the features that can be sourced on the mobile device. These features include historic application throughput and PHY metrics through the device APIs. For example, Android has telephony manager APIs that provide access to PHY-level information, such as RSRP, RSRQ, CQI, cell information, and signal strength from neighbour cell(s). The application can gather PHY and throughput metrics and feed them to the prediction engine to estimate future throughput, as illustrated in Figure 1.

In collaborative TP, metrics from other users sharing the same base station are collectively considered to predict the future throughput of the user. In this case, collaborative user data are considered additional model features to capture the context of cell resources and their utilization. The number of these users varies over time. Hence, the integration of the collaborative user information brings two design questions. First, how to develop a model that accommodates user dynamics. Second, how the collaborating user data will be collected in real networks.

To accommodate the variability in the number of users while ensuring a unified interface for collaborative metrics, the gathered information from collaborating users are aggregated using statistical metrics. In our models, we consider minimum, maximum, mean, quartiles (25%, 50%, 75%) and standard deviation of collected application and PHY features. Also, we consider the number of collaborating users as an additional feature that signifies the importance of the reported statistics. Note that as the number of users increases, the considered statistical features better represent the operating context.

The mechanism of gathering collaborative user information depends on the implementation scenario. We envision two implementation scenarios, including Service Provider (SP)- or NP-managed, depending on the entity orchestrating the collaborative TP. An overview of each scenario is shown below.

1) SP Managed Collaborative Prediction: In this scenario, the service provider would deploy an *Aggregator* to collect data from its active users in each cell. We envisage this to be effective for popular service providers with high penetration rates amongst users, thus engaging with many/most of the devices in a cell. Edge-assisted applications will have application-specific offloaded functions deployed at the edge, naturally representing endpoints for relevant metrics collected on the user device, as illustrated in Figure 1. However, the SP aggregator can also receive metrics' data directly from user devices. Each mobile device shares relevant prediction metrics periodically, e.g., every 1 second. In practice the overhead for this is very low (in terms of additional bytes to be transmitted) and it would be expected that the user device would be in frequent communication to its service provider in any case, so that such metrics might even be piggybacked on other communication. (We will return to this point in Section IV-C.) The device data could be shared with the Aggregator through a distributed event streaming platform, e.g., Kafka. The Aggregator would then implement an Application Programming Interface (API) that can be queried by an application function to get the predicted throughput when needed. To predict a user's throughput, the Aggregator would use its collaborative

TP model to predict the throughput based on the collected data across all devices in the cell.

2) NP Managed Collaborative Prediction: In this scenario, the network provider instead provides its own Aggregator that can obtain PHY metrics from the base station[1]. These metrics are collected from user control channel reports that are periodically communicated for scheduling and handover decisions, thus not requiring additional explicit communication by the device. However, the NP would lack access to application-level historic throughput history, unless it collects this through an API to the SP, as illustrated in Figure 1. Such an exchange could take place over the wired backbone with insignificant overhead. However, this design implies that NP should interact with the corresponding SP of every active user, raising scalability and interoperability issues. Thus, we propose a design optimisation that eliminates this requirement.

B. Collaborative TP Optimisations

NP-based prediction optimisation. We posit that NPs can provide accurate predictions by leveraging network-level throughput information alone, instead of also needing application-level throughput, thus negating any dependence on SPs. Figure 2 shows a scatter plot for PDCP throughput collected at the cellular base station and the application throughput recorded at the user device. The data for this plot was collected using the simulation setup described later in the paper. It illustrates a high correlation between the network and application throughput measures. Quite significantly, this optimisation eliminates the need for the NP Aggregator to interface with service providers or indeed seek to obtain application-level information from devices themselves. In this scenario, the NP would only maintain a single interface that enables end devices or an authenticated SP agent to query the predicted throughput.

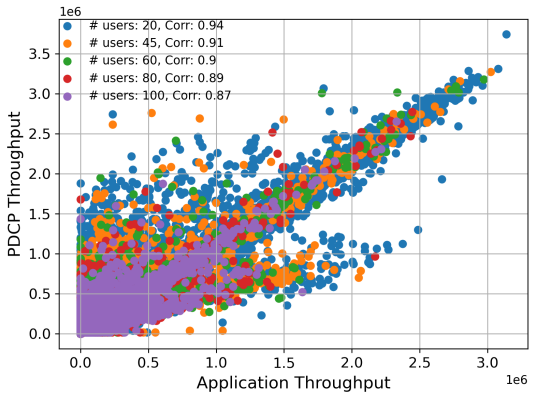


Fig. 2. Throughput and PDCP show significant correlation

Optimising inference computation. When the Aggregator receives a request from a user device requesting a TP request, it would be expected to then calculate collaborative statistical metrics separately for that target user. This process induces a computation overhead proportional to the number of active users and the frequency of TP requests from those users. To reduce such computation overhead, we propose calculating collaborative metrics' statistics based on the entire active user population, including the querying user. This change means that successive requests can use the same statistics in making a prediction (note that collaborative metrics capture the base station operating context independently of the requesting user). From a temporal perspective, another optimisation to mitigate overheads would be to reduce the frequency with which the collaborative statistical metrics are computed. For example, in a scenario involving mostly static and/or a small number of users, it might be acceptable to tolerate a less accurate TP in return for reduced computation costs. The design of such a tradeoff-guided mechanism lies beyond the scope of the current paper.

III. Dataset Collection

In this section we firstly explain why existing datasets are unsuitable for our purpose, followed by an explanation of the methodology that we used for data collection and processing.

A. Available Datasets

A recent survey paper bemoans the lack of publicly available datasets that contain detailed performance measurements of cellular networks [12]. Network providers routinely conduct drivetests but for commercial reasons do not usually make these available to others. The methodology for data collection usually involves the use of mobile devices actively downloading content, typically large files, in a range of different scenarios with and without mobility. The majority contain data from users devices; a few include data from the network side, but usually not from operational cellular networks. Datasets are typically suitable for standalone prediction engines, but lack information on other users sharing the same base station. This makes them unsuitable for our collaborative purpose, where we need to measure the simultaneous observations of many users within the same cell. To address this deficiency, we leveraged network simulation, specifically ns3, to gather data from multiple users connected to the same base station.

B. Data Collection Setup

Motivated by the lack of datasets containing information about all users sharing resources in the cell, we opt for controlled simulation setup in ns-3. We simulate LTE mobile networks with one-cell and varying numbers of mobile users sharing the resources.

Using ns-3, we create a one-cell, one-sector layout configuration, with the number of users ranging from 20 to 100, depending on the scenario[2]. Users are randomly scattered

[1]We assume that the base station exposes such data to other NP entities.

[2]Each scenario is independent experiment where the number of users are fixed.

across the cluster. For the mobility model, we use a random waypoint model with the minimum and maximum velocities set to 15 m/s and 20 m/s, respectively. We set the path loss model to a hybrid buildings propagation model, with a trace-driven fading model generated using a MATLAB script for users moving at an average velocity of 17 m/s. The serving cell antenna model is configured as an omnidirectional antenna model, with a transmission power of 46 dBm and a carrier bandwidth of 10 MHz (50 RBs). All users employ an on-off application over TCP for the application traffic model. The off and on time values are drawn from a uniform distribution, with minimum and maximum values set to 15 and 25 seconds, respectively. Experiment duration is set to 1000 seconds.

We collect raw PHY and throughput data using the built-in collection mechanism (i.e., data sources) in ns-3. While the data is collected with millisecond granularity, we average all values over a 1-second sampling window. This approach aligns with the data collection experiments in real mobile networks [13]. Using an Android phone and leveraging the standard Android library (*telephony* class) for PHY data collection enforces a minimum one-second sampling period. While sub-second sampling is possible in real networks, it requires a rooted phone and specialized software [14].

C. Raw Data Processing

After data collection, we process the raw data by averaging metrics over one-second intervals for each user independently. The calculation of collaborative statistics depends on the percentage of users participating in the collaboration. We compute statistics for scenarios with 10%, 30%, 50%, 70%, 90%, and 100% of participating users. To illustrate, in an experiment with the 100 users in the cell, a 10% scenario means that only 10 users are included in the collaborative statistics calculation. Collaborative statistics include the minimum, maximum, mean, quartiles (25%, 50%, 75%), and standard deviation of the collected application and PHY metrics from participating users. These statistics are computed for each one-second interval. Specifically, we iterate through each second for every user and calculate collaborative statistics for that interval. Additionally, only active participating users are randomly selected during each second. To ensure statistical significance, this process is repeated 10 times per user. Finally, each user's output file contains both their individual statistics and the collaborative statistics for each run.

IV. Performance Evaluation

Section IV-A presents an overview for our evaluation. We present the impact of collaboration on the prediction accuracy in Section IV-B followed by the performance of optimised models in Section IV-C.

A. Evaluation Setup

We investigate the impact of collaboration through comparing the prediction accuracy in different scenarios as follows:

- ***Standalone*** TP represents prediction models limited to target user data.
- ***SPx*** represents SP managed collaboration managed with x % of cell users.
- ***NP*** represents NP managed collaboration, where prediction is based on statistical features from all cell users.

For these scenarios, we consider different contexts capturing user activity and prediction horizons.

We use the Random Forest (RF) algorithm [15], an ensemble learning method based on bagging, for throughput prediction. Our focus is on predicting the *average* throughput over given horizon. For the input data, we use the full *history* of each PHY, throughput and collaborative metrics.

Our key metric is the Absolute Residual Error ratio (ARE), defined as the absolute value of the residual error divided by the actual throughput. The residual error is the difference between the actual and predicted throughput values.

B. Impact of Collaboration

1) Inactive user case: Predicting throughput in case of inactivity is relevant to many practical scenarios, such as starting or resuming a streaming session or starting a file update process. The inability to accurately capture the network throughput impacts many performance metrics, such as start up delay or video quality selection. TP models for inactive users solely rely on PHY metrics for non-collaborative models. In collaborative models, PHY and collaborative user application metrics are also used as input to the model. Figure 3 shows the prediction accuracy for inactive users in different collaborative scenarios with both history and horizon set to 5 seconds.

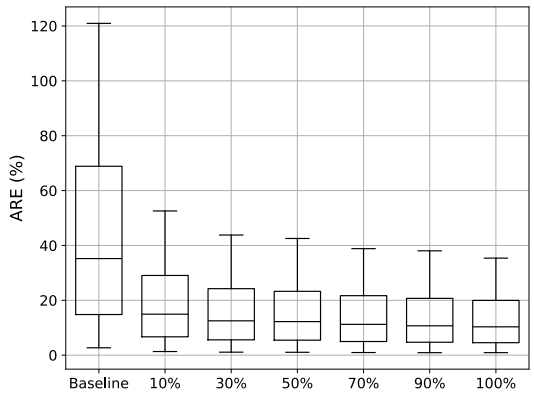


Fig. 3. Prediction accuracy for different collaboration scenarios for inactive users (H5H5)

The five-second horizon is useful for media streaming with short segment durations. The figure shows that collaboration significantly reduces the prediction error. Collaboration reduces the prediction error 75 percentile from 76 in case of standalone prediction to 30% and 20% in case of SP10 and NP, respectively. Hence, collaboration between users can reduce TP error by up to 66%.

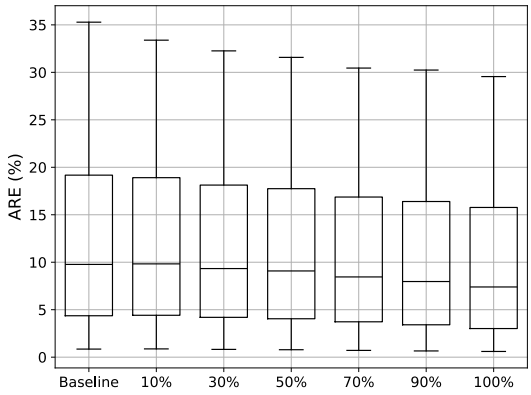


Fig. 4. Prediction accuracy for different collaboration scenarios for active users (H5H5)

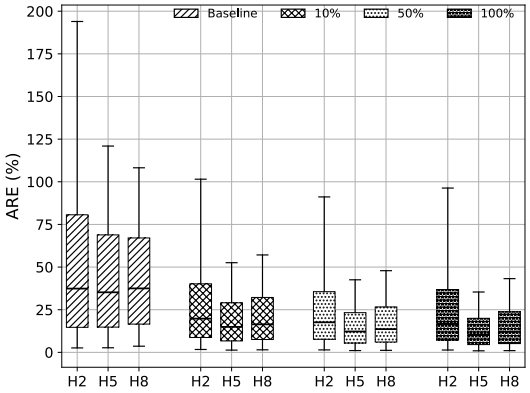


Fig. 5. Prediction accuracy for different collaboration scenarios for inactive users (history is same for all scenarios and is set to 5s).

2) Active user scenario: Adaptive applications integrate network status in various decisions to deliver the best user experience. Figure 4 compares the prediction error for active users in various collaborative scenarios with both history and horizon set to 5 seconds. While standalone prediction features a significant improvement to the inactive case, user collaboration still facilitates error reduction as the number of collaborating users increases in all error statistics. The prediction error 75 percentile drops from 19.2% in standalone TP to 16% in case of NP case. This reduction corresponds to a 17% reduction in the prediction error. Hence, collaboration improves TP model perception of cell context leading to improved prediction accuracy.

3) Impact on different Horizons: Figures 5 and 6 plot the absolute relative residual error for the inactive and active scenarios, respectively, for various prediction horizons (2, 5, 8 seconds). Both figures shows that collaboration improves the prediction accuracy for all horizons. The figures also show increasing the horizon makes the prediction more challenging for active case for both collaborative and non-collaborative prediction scnarios. However, collaborative and non-collaborative models exhibit different trends in the inactive case.

C. Optimised Collaboration Performance

Figures 7 and 8 compare the prediction accuracy when TP models consider collaborative user metrics only versus using metrics from all cell users to capture the network context for inactive and active target user, respectively. Both figures show identical error statistics in all scenarios of user activity and collaboration population. This result confirms our proposed approach for optimising the aggregator calculation as presented in Section II-B

Figure 9 compares the prediction accuracy of the models that consider PDCP throughput as a feature to the original models that rely on user throughput in the case of active and inactive users. In both cases, the accuracy statistics exhibit identical performance. Hence, the NP can leverage its own

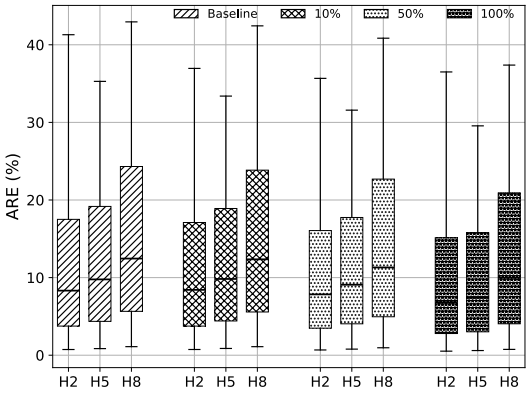


Fig. 6. Prediction accuracy for different collaboration scenarios for active users (history is same for all scenarios and is set to 5s).

scheduling information to predict future user throughput without sourcing the application throughput from all SPs or end-devices.

In the SP solution, each user device is expected to periodically report its summarised metrics to the Aggregator. In our implementation this comprises 5 numeric values for a total of 20 bytes. Even if this were to be reported every 1 second, it is clear that the additional overhead is negligible, and likely to be dwarfed by other service-specific updates from the application on the device to the service provider.

V. Conclusion

The topic of throughput prediction in cellular networks has garnered considerable interest in recent years, with its promise to improve the performance and user experience of a variety of applications, including video streaming, gaming and augmented reality. Recent proposals employ machine learning, which allow sophisticated multi-variate models to be trained, yielding good results even in such dynamic settings. Models are trained using data gathered from either network

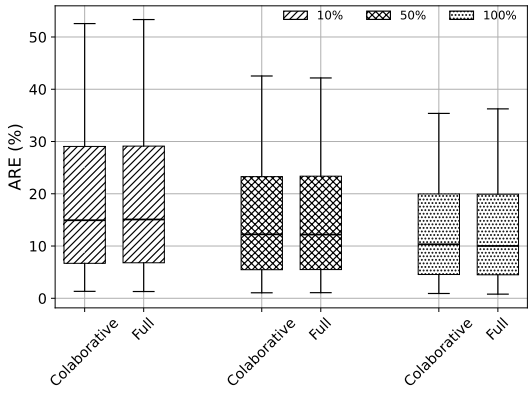


Fig. 7. Prediction accuracy for default and SP-managed scenarios for inactive users (history and horizon are set to 5s).

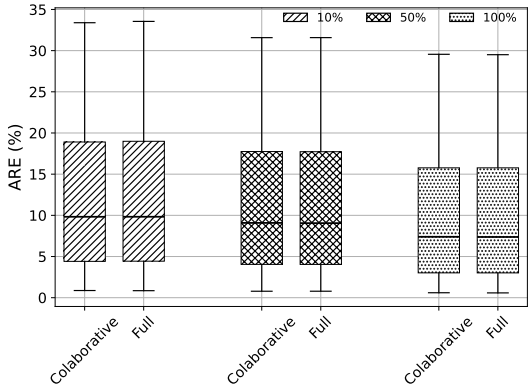


Fig. 8. Comparing collaboration-based and cell-based features for active users (history and horizon are set to 5s).

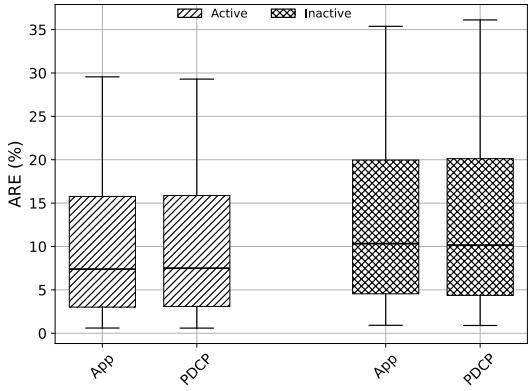


Fig. 9. Prediction accuracy for collaboration scenarios with and without PDCP throughput as a feature (H5H5)

simulations or so-called drivetests, in which metrics from end-users devices are collected. Existing proposals operate having each user device make its own predictions on the basis of its local measurements. Our hypothesis is that pooling of device measurements in a collaborative manner yields more accurate predictions, by allowing a broader set of observations to be combined. We presented our collaborative approach, and a set of results, which demonstrate a significant reduction for prediction error when compared to operating standalone. Future work includes detailed design of a system to collect and process data from users, and its optimisation in terms of the tradeoff between overheads and performance accuracy.

ACKNOWLEDGMENT

This publication has emanated from research supported in part with the financial support of Taighde Éireann - Research Ireland under Grant number 13/RC/2077_P2. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

REFERENCES

[1] D. Raca, A. H. Zahran, C. J. Sreenan, R. K. Sinha, E. Halepovic, R. Jana, and V. Gopalakrishnan, "On leveraging machine and deep learning for throughput prediction in cellular networks: Design, performance, and challenges," *IEEE Communications Magazine*, vol. 58, no. 3, 2020.

[2] D. Minovski, N. Ögren, K. Mitra, and C. Åhlund, "Throughput prediction using machine learning in LTE and 5G networks," *IEEE Transactions on Mobile Computing*, vol. 22, no. 3, pp. 1825–1840, 2023.

[3] I. Batool, M. M. Fouda, and Z. M. Fadlullah, "Deep learning-based throughput prediction in 5G cellular networks," in *2024 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 2024, pp. 1–6.

[4] W. Ye, X. Hu, S. Sleder, A. Zhang, U. K. Dayalan, A. Hassan, R. A. K. Fezeu, A. Jajoo, M. Lee, E. Ramadan, F. Qian, and Z.-L. Zhang, "Dissecting carrier aggregation in 5g networks: Measurement, qoe implications and prediction," in *Proceedings of the ACM SIGCOMM 2024 Conference*, ser. ACM SIGCOMM '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 340–357.

[5] A. Ahmad, A. B. Mansoor, A. A. Barakabitze, A. Hines, L. Atzori, and R. Walshe, "Supervised-learning-based QoE prediction of video streaming in future networks: A tutorial with comparative study," *IEEE Communications Magazine*, vol. 59, no. 11, pp. 88–94, 2021.

[6] G. Lv, Q. Wu, W. Wang, Z. Li, and G. Xie, "Lumos: towards better video streaming QoE through accurate throughput prediction," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022, pp. 650–659.

[7] Z. Jia, X. Min, and G. Zhai, "End-to-end prediction of streaming video quality of experience: Dataset and approach," in *2024 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 2024, pp. 1–5.

[8] D. Raca, A. H. Zahran, C. J. Sreenan, R. K. Sinha, E. Halepovic, and V. Gopalakrishnan, "Device-based cellular throughput prediction for video streaming: Lessons from a real-world evaluation," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 318–334, 2024.

[9] M. Boban, C. Jiao, and M. Gharba, "Measurement-based evaluation of uplink throughput prediction," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, 2022, pp. 1–6.

[10] L. Mei, J. Gou, Y. Cai, H. Cao, and Y. Liu, "Realtime mobile bandwidth and handoff predictions in 4g/5g networks," *Computer Networks*, vol. 204, p. 108736, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621005879

[11] H. N. Aung and H. Ohsaki, "FL-PERF: Predicting TCP throughput with federated learning," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2023, pp. 4332–4337.

[12] M. Amini, R. Stanica, and C. Rosenberg, "Where are the (cellular) data?" *ACM Comput. Surv.*, vol. 56, no. 2, Sep. 2023. [Online]. Available: https://doi.org/10.1145/3610402

[13] D. Raca, J. J. Quinlan, A. H. Zahran, and C. J. Sreenan, "Beyond throughput: A 4G LTE dataset with channel and context metrics," in *9th ACM Multimedia Systems Conference*, ser. MMSys '18, 2018, pp. 460–465.

[14] Y. Li, C. Peng, Z. Yuan, H. Deng, J. Li, and T. Wang, "Mobileinsight: Analyzing cellular network information on smartphones," *GetMobile: Mobile Comp. and Comm.*, vol. 21, no. 1, p. 39–42, may 2017. [Online]. Available: https://doi.org/10.1145/3103535.3103548

[15] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, 2001.