

Channel stability prediction to optimize signaling overhead in 5G networks using machine learning

Sihem Bakri*, Maha Bouaziz*, Pantelis A. Frangoudis[‡], and Adlen Ksentini*

*EURECOM, Sophia Antipolis, France

[‡]Distributed Systems Group, TU Wien, Vienna, Austria

Email: *{name.surname}@eurecom.fr, [‡]pantelis.frangoudis@dsg.tuwien.ac.at

Abstract—Channel quality feedback is crucial for the operation of 4G and 5G radio networks, as it allows to control User Equipment (UE) connectivity, transmission scheduling, and the modulation and rate of the data transmitted over the wireless link. However, when such feedback is frequent and the number of UEs in a cell is large, the channel may be overloaded by signaling messages, resulting in lower throughput and data loss. Optimizing this signaling process thus represents a key challenge. In this paper, we focus on Channel Quality Indicator (CQI) reports that are periodically sent from a UE to the base station, and propose mechanisms to optimize the reporting process with the aim of reducing signaling overhead and avoiding the associated channel overloads, particularly when channel conditions are stable. To this end, we apply machine learning mechanisms to predict channel stability, which can be used to decide if the CQI of a UE is necessary to be reported, and in turn to control the reporting frequency. We study two machine learning models for this purpose, namely Support Vector Machines (SVM) and Neural Networks (NN). Simulation results show that both provide a high prediction accuracy, with NN consistently outperforming SVM in our settings, especially as CQI reporting frequency reduces.

Index Terms—5G, signaling overhead, CQI optimization, machine learning, SVM, NN.

I. INTRODUCTION

5G mobile networks [1] are designed to provide new and enhanced services that make life easier in several areas introduced in the Internet of Things (IoT), such as smart cities, health care, agriculture, transportation, and manufacturing. To accommodate these services, 5G has to meet critical requirements in terms of low latency, high reliability, high bandwidth, and the support for massive numbers of connected devices.

Providing a reliable communication technology represents a key challenge for 5G systems in both Core Network [2] and Radio Access Network (RAN) levels. In order to achieve reliability at the RAN level, a base station (eNodeB or gNodeB, in 4G and 5G terminology, respectively) should allocate a sufficient amount of radio resources per UE, and appropriately select the modulation and coding scheme (MCS) in order to meet the requirements of each considered application. The amount and configuration of these resources, i.e., physical Resource Blocks (pRB), are directly related with the channel conditions at the UE end. For this reason, the base station should ideally know in real time the quality of the channel of each device, which allows it to properly schedule the necessary number of physical resources (NpRB) for transmission [3]. In 4G and

5G networks, this number depends on the *Channel Quality Indicator (CQI)* value, which is periodically reported by UEs to the base station, and conveys their current communication channel quality [4]. Nevertheless, the periodic transmission of CQI information incurs signaling overhead; this may overload links and negatively impact RAN performance. Therefore, it is important to optimize this signaling process, in order to be able to improve on Quality of Service (QoS).

This work is put in the context of our 5G RAN slicing design presented in [5], including a slice orchestrator responsible for cross-slice resource sharing [6]. Our architecture requires accurate channel quality information per UE at the gNodeB and at the slice orchestrator level, in order to be able to estimate and dynamically adjust the radio resource allocation to satisfy the heterogeneous requirements of co-existing network slices. This information is reported by UEs via standard procedures, and propagates to the slice orchestrator via a southbound protocol by base stations. The challenge that we face and the particular motivation for this paper is to reduce this reporting overhead.

The key element responsible for fluctuations in CQI values are the changes in the radio environment which may be due to user mobility, multi-path effects, and other phenomena. We introduce the term *channel mobility* to denote time-varying changes in the radio environment of a UE: On the one hand, the channel is considered static if its conditions are mostly stable, when typically the UE is static or low-mobility for a period of time. Thus, the reported CQI values remain constant or show minimal variation, which does not impact radio resource allocation. On the other hand, the channel is considered mobile when it varies significantly due to factors such as UE mobility and other effects. In this case, the CQI values exhibit significant fluctuations. Consequently, it is crucial that the base station is informed about the changed channel quality information, in order to determine the appropriate amount of resources to be allocated, and update the NpRB values for the different UEs.

Our work is in the direction of reducing the signaling overhead by optimizing the reporting of CQI information via limiting the amount of unnecessary transmitted messages, at the same time ensuring that the e/gNodeB has an accurate view of the channel conditions at the UE end. Ideally, the UEs should notify the base station only when their channel conditions have actually changed.

Detecting whether the channel is static or mobile over time, though, is challenging, and this is the main issue we address in this paper. To this end, we apply machine learning (ML) techniques in order to be able to detect channel variations. Our approach involves collecting data from the system in order to study, analyze and extract the information needed to make a decision. In fact, having a lot of different types of data about per-UE channel quality, complemented with other information such as user mobility patterns, fine-grained geographical locations, etc. would assist in getting a more accurate view from the output of the ML algorithm, to properly identify the stability of the channel. However, this may not be feasible for technical and privacy reasons.

In this paper, we make the following contributions: We propose a ML-driven methodology to predict channel mobility and accordingly adapt the CQI reporting frequency, aiming to reduce signaling overhead while maintaining an accurate view of channel conditions per UE to appropriately allocate radio resources. We study, analyze and predict the channel's state using two different machine learning algorithms to evaluate their suitability and select the more accurate one for our purposes. Our first design goal is to avoid the collection of many data metrics, thus focusing only on the CQI parameter; the features we have selected for training and classification can be calculated solely by the statistical processing of the collected CQI values. Our second design goal is to avoid the collection of large volumes of CQI data; to this end, we evaluate different frequencies to collect these data and their impact on the accuracy of identifying channel mobility.

The remainder of this paper is organized as follows: Section II discusses some of the relevant works in the literature which focus on reducing the CQI signaling overhead. Our proposed ML-based method is described in detail in Section III. Section IV presents performance evaluation results, where we quantitatively compare candidate ML algorithms for our purposes. Finally, Section V concludes the paper and reports on our plans for future work.

II. RELATED WORK

Several research works in the literature have been elaborated in order to control the transmission of CQI information, allowing the optimization of the signaling overhead. Two families of such control techniques have been devised.

A. Techniques based on frequency

Techniques of this kind are based on compression models. Indeed, the idea to reduce the signaling overhead consists in sending a compressed CQI value of a series of pRBs, instead of sending a CQI value for each one. In this context, three categories were proposed as follows [7]: i) Broadband compression, where a single CQI value transmitted refers to all pRBs of the bandwidth, ii) sub-band compression, where the bandwidth is divided into multiple sub-bands with the same size, and the UE selects only one CQI value to be transmitted to the base station, and iii) full band compression, where the base station estimates the total bandwidth quality,

using mathematical transformations such as the discrete cosine transform and the Haar wavelet transform.

Sivridis and He [8] presented a non-predictive signaling reduction scheme, where users with high signal-to-interference-plus-noise ratio (SINR) transmit only broadband information, while users with low SINR are allowed to return on-demand instant CQI information at high rates. Therefore, a technique was proposed to determine the threshold that separates users required to use full-band feedback from users required to use compression in the wide-band frequency domain.

The work of Kang and Kim [9] is based on the sub-band compression method, allowing to analyze and select the best M-feedback for orthogonal frequency division multiple access (OFDMA) systems. In addition, a combined optimization was applied to minimize feedback overhead costs based on the number of reported resource blocks per user and the signal to quantization noise ratio bits.

Abdulhasan et al. [10] presented a compression scheme for CQIs in a 3GPP-LTE and LTE-A system, where CQI values are communicated to eNodeBs based on a defined threshold. A trade-off was presented to select the appropriate threshold, since a high threshold is recommended for high speed conditions, whereas a low threshold is recommended to ensure reliable transmission mainly in an overloaded network.

B. Techniques based on timing

Chiumento et al. [11] proposed an estimation method of the channel quality based on Gaussian Process regression at the base station. This is achieved thanks to an adaptive and online CQI prediction scheme allowing estimation of the channel quality variations, and the behavior monitoring of each user.

In [12], the authors dealt with the CQI aging problem, which could be defined by the mismatch between the CQI used for the channel adaptation and the current state of the channel. This problem is caused due to processing delays or because of infrequent CQI reporting. To overcome this problem, authors proposed a comparison study of various signal-to-noise ratio prediction algorithms, such as Kalman filters, among others.

In [13], an algorithm is presented for dynamic CQI resource allocation using ARQ information, Doppler mobility monitoring, and the MAC layer service classifier. The idea consists in the combination of information from the physical and MAC layers, providing an additional information about the channel quality and its effect on MAC frames in terms of delay, packet error rate, etc. This approach allows tuning the periodicity of the feedback window in order to address QoS, robustness, and feedback overhead tradeoffs.

The proposed methods to optimize signaling overhead provided relatively interesting results. However, almost all of them consisted in the prediction of channel quality based on complex optimization algorithms, as well as required several input parameters that are often not feasible to acquire, nor accurate in some conditions. Our approach is inspired from these schemes, but it is based on a simpler intelligent mechanism, which only requires as input CQI information for accurate channel mobility prediction.

III. CHANNEL STABILITY PREDICTION USING MACHINE LEARNING

This section focuses on the description of the proposed concept to predict channel stability based on ML. This mechanism can then be used to optimize the transmission of CQI data messages and thus reduce the associated signaling overhead.

A. Overview and objectives

CQI messages are sent periodically from UEs to the e/gNodeB, in order to provide it information about the channel quality allowing it to appropriately allocate resources. When channel quality is relatively stable, the CQI values do not vary a lot. Therefore, an increased CQI reporting frequency does not contribute to the view the base station has on the actual radio conditions of a UE link, and does not affect the quality of the radio resource allocation. We thus take advantage of channel stability to avoid transmitting unnecessary CQI reports and alleviate the associated overhead.

Our approach consists in monitoring the channel state for a period T . If channel mobility is identified by the predictor, a new CQI value is required to adjust resource allocation. Otherwise, there is no need to receive new CQI values; the e/gNodeB allocates radio resources considering the last received CQI value as accurate and stable, and the CQI reporting frequency can be reduced. The different steps of this concept are illustrated in Fig. 1.

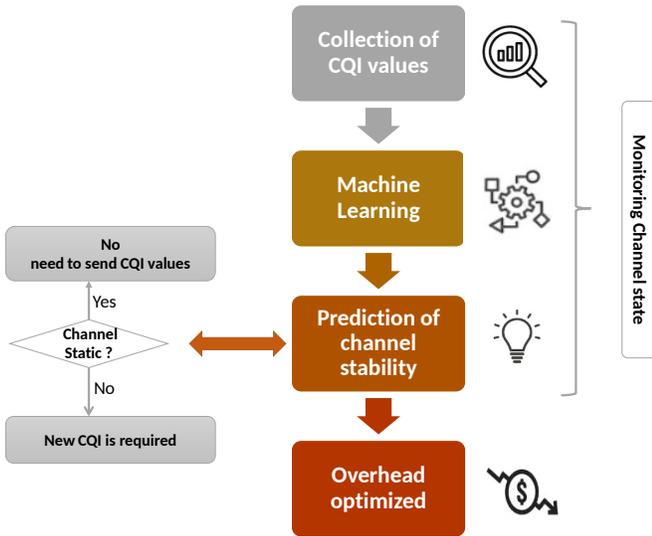


Fig. 1: Our concept and methodology to reduce CQI monitoring overhead.

The proposed monitoring phase is based on a machine learning algorithm, which helps predict the channel state. In this paper, we have tested some ML algorithms, then we have selected Support Vector Machines and Neural Networks as they offer better accuracy. The next section focuses on describing how these two ML schemes are applied in our network settings in order to predict channel mobility based on different frequencies of collected data.

B. Steps to predict the channel's state

In this section, we present in detail the steps involved in our ML-based methodology, and the metrics we use to evaluate the performance of the candidate algorithms for channel mobility prediction. Indeed, there are different machine learning approaches [14]. The most commonly used are: i) Supervised learning, which consists in training the algorithm using a set of data consisting of an input and a desired output. Then, a function that maps an input to an output is inferred based on the training data. This function allows the mapping of new unseen data instances to output values, which may correspond to distinct classes; ii) Unsupervised learning, which consists in learning in a self-organized way allowing to find an unknown sample from a set of data without being based on an existing label; and iii) Reinforcement learning, which allows to decision making by interacting with an environment, formulated as a Markov decision process. It has similarities with supervised learning, but without the need for labelled input/output pairs.

In this paper, we apply supervised learning techniques and define two classes (static and mobile) based on the CQI parameter. We evaluate two supervised learning algorithms in order to predict the channel state and assign it to the appropriate class:

- Neural Network (NN) [15]: This mechanism is modelled and inspired from the human brain, aiming to create an artificial neural network. The concept consists in learning the machine by incorporating new data. The machine typically consists of different layers of interconnected neurons, each one of which interprets the input data through a kind of machine perception and sends an output to a connected neuron, until the last layer provides the output of the system.
- Support Vector Machine (SVM) [16]: This approach consists in learning from a set of multi-dimensional data vectors labelled by their category (class), and creating a model used to classify new data by finding the hyperplane that separates the training data by the optimal (maximum) margin. SVM is a binary *linear* classifier.

Fig. 2 presents the different steps involved in the channel state prediction process.

1) *Feature vector creation and labeling phase*: This phase involves the collection of data and their processing in order to extract specific features and create feature vectors (also called characteristic vectors) that will be used for training a classifier. Raw data are collected in the form of vectors for different channels during a period T . A feature vector is then created for each data vector (i.e, for each channel).

In fact, different types of data representing the channel state may be used, such as SNIR, CQI, and others [17]. We select the CQI parameter for the proposed predictive system, as this parameter provides sufficient information on the channel state and is used by the MAC scheduler to allocate resources and decide on parameters such as the MCS. We extract a feature vector from each CQI data vector after a preprocessing step,

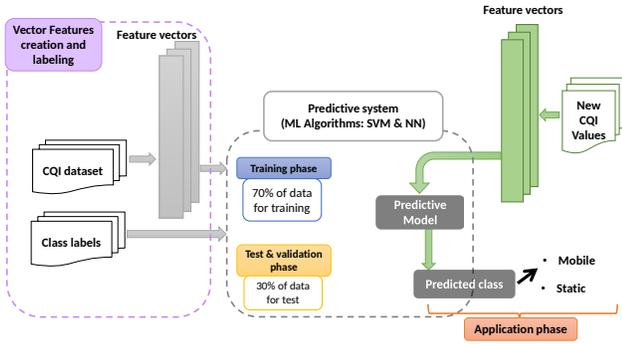


Fig. 2: ML-driven channel stability prediction.

in order to have the relevant data for the predictive system to identify the channel state (mobile or static).

Preprocessing is carried out on the data vector $CQI_T = [cqi_1, cqi_2, \dots, cqi_n]$ of n CQI values collected during a period T in order to extract the characteristic vector $F = [F_1 \ F_2 \ F_3]$. The extracted features are the following.

- F_1 : The difference between the maximum and minimum values of collected CQIs in the data vector CQI_T .

$$F_1 = cqi_{max} - cqi_{min} \quad (1)$$

The channel may be static if F_1 is small or zero, which might mean that the UE is static and the environment is stable (there are no significant effects that cause a drastic change in the CQI value). This feature can provide an idea of the channel state, but it is not sufficient to make a decision.

- F_2 : Variance.

$$F_2 = \frac{1}{n} \sum_{n=1}^n (cqi_i - \overline{CQI_T})^2 \quad (2)$$

This feature measures the dispersion of CQI values relatively to the average $\overline{CQI_T}$, which characterizes the level at which the CQI can have a value more or less far from its expectation.

- F_3 : The vertical change of the CQI curve slope, representing the CQI change in different samples in period T .

$$F_3 = |CQI(t_{i+\Delta}) - CQI(t_i)|, \quad (3)$$

where $CQI(t_i)$ and $CQI(t_{i+\Delta})$ are the CQIs collected at t_i and $t_{i+\Delta}$ respectively, where these two times are inside the sample ($\Delta=5$ in our case). Multiple F_3 values are extracted for each sample. Thus, the size of F depends on the number of F_3 .

After the creation of vector F , a known label (static or mobile) is assigned to it, in order to be used for the training phase.

2) *Creation of a machine learning-based predictive system:* To create the predictive system, a machine learning algorithm operates in two phases as follows.

a) *Training phase:* 70% of the feature vectors with their labels (representing the real classes) are used to train the classifier. During this training phase, the ML algorithm creates a function that maps inputs (feature vectors) to outputs (labels), used then to classify new vectors. In this stage, the SVM algorithm learns a linear function, while the NN algorithm also supports non linear functions.

b) *Test and validation phase:* This phase uses the rest of the feature vectors (30%). It consists in checking the predicted classes of these vectors against their assigned labels. The validation of the predictive system is based on a confusion matrix [18], which consists of the number correctly and incorrectly classified samples per class. Performance is evaluated in terms of the following metrics:

- Accuracy, i.e., the ratio of the number of correctly predicted vectors to the total number of vectors.

$$\text{Accuracy} = \frac{\# \text{ correctly predicted}}{\# \text{ feature vectors}} \quad (4)$$

- F1-score, which is defined by the weighted average of *precision* and *recall*, where, precision is the ratio of the number of correctly predicted mobile class instances to the total number of predicted mobile class ones (i.e., false and correct), and recall, also called sensitivity, is the ratio of the number of correctly predicted instances of the mobile class to the number of all true mobile class ones.

$$\text{F1.score} = \frac{2 * (\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} \quad (5)$$

3) *Application phase:* This step consists in classifying a new CQI data set over different frequencies of collecting data (i.e., sample size variation). To evaluate this phase, we use the *True Positive Rate (TPR)* and the *True Negative Rate (TNR)* metrics, where the positive class refers to the mobile class and the negative one refers to the static class. TPR and TNR are defined as follows:

$$\text{TPR} = \frac{\# \text{ correctly classified as mobile}}{\# \text{ mobile}} \quad (6)$$

$$\text{TNR} = \frac{\# \text{ correctly classified as static}}{\# \text{ static}} \quad (7)$$

IV. PERFORMANCE EVALUATION

This section focuses on the performance evaluation of the channel mobility predictive system provided by the two ML algorithms (NN and SVM). We first create our dataset by generating CQI values for different channel mobility states using the ns3 simulator. Then, we use MATLAB [19], [20] to train and test ML algorithms based on the provided data set, as well as to evaluate new CQI data sets with different CQI collection frequencies. For the NN case, we trained a neural network with a single hidden layer using the Levenberg-Marquardt algorithm. We experimented with different layer

sizes and found that using 10 neurons in the hidden layer provided the best accuracy among the options that we tested. Performance is evaluated first for the test and validation phase, and then for the application phase for both ML algorithms.

1) *Test and validation phase evaluation:* In order to create a data set with realistic CQI values corresponding to different degrees of user mobility, we simulated an LTE cell using ns3, where UEs move with different constant velocities. We thus generated approximately 15,500 vectors of CQI values with different channel mobility states and extracted a feature vector for each CQI vector as described in Section III-B1, which we labelled either as static or mobile, depending on the level of UE mobility. Note that a feature vector is calculated on a sequence of CQI values collected during a time period $T = 400$ ms. For both ML algorithms considered, we use 70% of our data for training and the remaining 30% for test and validation. Table I presents the results of the validation phase in terms of accuracy and F1-score for the two candidate ML mechanisms.

TABLE I: Accuracy and F1-score of NN and SVM algorithm

	NN	SVM
Accuracy	96.43%	92.86%
F1-score	96.29%	92.30%

As shown in this table, both algorithms are able to learn and predict the channel state with high performance, as they provide an accuracy and F1-score of more than 90%. We notice, though, that the NN scheme outperforms SVM in terms of accuracy and F1-score by approximately 4%.

This can be explained by the fact that SVM is based on the margin maximization of the linear hyperplane separator between the two classes (static and mobile) [16]. That is why it is not able to well classify vectors close to this separation. Contrariwise, NN is based on a non-linear function to separate between classes allowing it to better handle such cases.

2) *Application phase evaluation:* After the test and validation of the predictive system, we use ns3 simulations in a similar way to create a new CQI dataset for the application phase, in order to evaluate the efficiency of our classifiers and evaluate their behavior for different CQI collection frequencies. The generated feature vectors are created from raw data that correspond to two types of UE mobility (mobile and stable) and are labelled as such. Four groups of test data were generated, each for a different CQI reporting frequency, namely every 2, 10, 50 and 100 ms. Performance is evaluated in terms of TPR for the mobile class and TNR for the static class. The obtained results are as follows:

- The static channel state obtains TNR performance between 99% and 100% for any CQI collection frequency and for both NN and SVM algorithms. In fact, for this class, all CQI values are relatively close to each other. This is why the selected samples with the different frequencies provide a small variation in the features of the vector F , allowing to identify the channel as static.

Therefore, it is evident that both algorithms succeed in correctly predicting channel mobility in the static case.

- The prediction of the mobile channel state is harder, as CQI values are highly varied. Therefore, the predictive system should detect the variation of the CQI values with different data collection frequencies, as well as, appropriately select samples (with the varied CQI values) on which it is based to select the appropriate class. The obtained results of TPR performance for the different reporting periods are illustrated in Fig. 3.

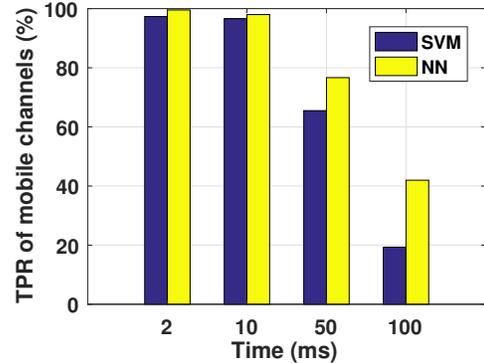


Fig. 3: TPR of mobile channel for different CQI reporting frequencies. The x -axis represents the period between two consecutive CQI reports by a UE.

As shown in this figure, for small periods (2ms and 10ms), both algorithms achieve a high TPR performance (more than 95%). It can also be noticed that there is a small increase with the NN algorithm. However, for low CQI collection frequencies, TPR is significantly reduced for both NN and SVM algorithms. For 50ms and 100ms, TPR is respectively around 76.67% and 42% with NN, compared to 65.5% and 19% respectively when using SVM. These results show that the predictive system provided by NN has the ability to detect channel mobility more effectively and is more robust when CQI is reported with a lower frequency (large periods). Although when the reporting period is 100ms the TPR is relatively low for NN, it still achieves a 2× increase in performance compared to the TPR of the SVM algorithm. The NN algorithm outperforms SVM, thanks to the non-linear function used to separate between classes.

3) *Prediction quality evaluation:* This part focuses on evaluating the quality of the prediction of the NN and SVM-based approaches relatively to different CQI collection frequencies. To quantify it, we are relying on appropriate prediction quality metrics for each ML scheme. In particular, the metric we use for SVM is the prediction likelihood (also called prediction probability), which is an expression of the certainty that a feature vector is correctly classified. It is calculated by taking into account how far the score returned by the SVM classifier is from the threshold value that classifies a test vector. In a similar sense, the metric we use for the NN classifier is the

Mean Squared Error (MSE), which conveys the uncertainty about the correctness of the classification.

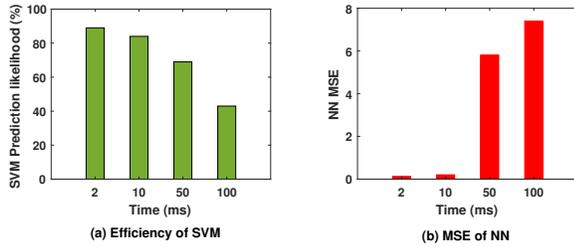


Fig. 4: Prediction score for different CQI reporting frequencies. The x -axis represents the period between two consecutive CQI reports by a UE.

As illustrated in Fig. 4, when the CQI collection frequency decreases, the likelihood to correctly predict the channel state by the SVM algorithm reduces and the mean squared error of the NN algorithm increases. These results are due to the fact that when the CQI reporting frequency is smaller, there are fewer raw CQI samples during the time window $T = 400$ ms out of which a feature vector is created. This lost information has often the effect that the variability of CQI values in a window decreases. There are cases when the real value of the CQI between consecutive samples in a window fluctuates but this is not captured in the data samples, making the CQI appear to remain mostly constant during the collection period and, in turn, causing the algorithm to mis-classify the channel as static. For these reasons, the prediction error rate impacts the TPR as presented in Fig. 3, where the TPR drops as the collection data frequency decreases.

V. CONCLUSION

We focused on ways to reduce the signaling overhead caused by the periodic transmission of channel quality feedback in the form of CQI reports in 4G and 5G mobile networks. Our approach consists in avoiding to transmit unnecessary CQI messages by taking into account the stability of channel conditions, i.e., reducing the amount of CQI reports when the value of the latter does not change significantly over time, as a result of a stable channel. To this end, we addressed the challenge of predicting the channel's stability, proposing machine learning-based mechanisms that only require CQI information as input. Our mechanisms thus operate in a standards-compliant way and require no cross-layer or other external information, such as user locations or mobility patterns. We compared two ML schemes for this purpose, namely Support Vector Machines (SVM) and Neural Networks (NN), evaluating and analyzing their prediction accuracy. We further addressed the tradeoff between prediction accuracy and data collection frequency, and experimentally showed neural networks to consistently outperform SVMs in all our settings.

In this paper, we mainly focused on evaluating the prediction accuracy of the candidate ML schemes. The next step is to launch a deeper study on the impact of our proposed methodology and mechanisms, integrating them in the 5G network

slice management architecture that we have proposed in our prior work. Our immediate goal is to evaluate the signaling cost improvements that can be achieved, and the impact of our proposed mechanisms on the allocated resources and the attained performance in terms of latency and throughput for heterogeneous 5G network slices.

ACKNOWLEDGEMENT

This work was partially supported by the European Union's Horizon 2020 Research and Innovation Program under the 5G!Drones project (Grant No. 857031).

REFERENCES

- [1] M. Agiwal, et al. "Next Generation 5G Wireless Networks: A Comprehensive Survey," *IEEE Communications Surveys and Tutorials*, vol. 18(1), pp. 1617-1655, Feb. 2016.
- [2] N. Salhab et al. "NFV Orchestration Platform for 5G over On-the-fly Provisioned Infrastructure", IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2019.
- [3] F. Capozzi et al. "Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey," *IEEE Communications Surveys and Tutorials*, vol. 15(2), pp. 678-700, Jul. 2012.
- [4] Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception (3GPP TS 36.101 version 15.3.0 Release 15).
- [5] A. Ksentini et al. "Providing Low Latency Guarantees for Slicing-Ready 5G Systems via Two-Level MAC Scheduling," *IEEE Network*, vol. 32(6), pp. 116-123, Dec. 2018.
- [6] S. Bakri et al. "Dynamic slicing of RAN resources for heterogeneous coexisting 5G services," in Proc. IEEE GLOBECOM, 2019.
- [7] M. Cordina et al., "A support vector machine based sub-band CQI feedback compression scheme for 3GPP LTE systems," in Proc. ISWCS, 2017.
- [8] L. Sivridis and J. He, "A Strategy to Reduce the Signaling Requirements of CQI Feedback Schemes," *Wireless Personal Communications*, vol. 70(1), pp. 85-98, May 2013.
- [9] M. Kang and K.S. Kim, "Performance Analysis and Optimization of Best-M Feedback for OFDMA Systems," *IEEE Communications Letters*, vol. 16(10), Oct. 2012.
- [10] M.Q. Abdulhasan et al. "An Adaptive Threshold Feedback Compression Scheme Based on Channel Quality Indicator (CQI) in Long Term Evolution (LTE) System," *Wireless Personal Communications*, vol. 82(4), pp. 2323-2349, Jun. 2015.
- [11] A. Chiumento et al. "Adaptive CSI and feedback estimation in LTE and beyond: a Gaussian process regression approach," *EURASIP Journal on Wireless Communications and Networking*, Jun. 2015.
- [12] R. Abi Akl et al. "Compensating for CQI aging by channel prediction: The LTE downlink," in Proc. IEEE GLOBECOM, 2012.
- [13] M. Abdul Awal et al. "Dynamic CQI Resource Allocation for OFDMA Systems," in Proc. IEEE WCNC, 2011.
- [14] O. Simeone, "A Very Brief Introduction to Machine Learning With Applications to Communication Systems," *IEEE TCCN*, vol. 4(4), pp. 648-664, Nov. 2018.
- [15] I.M. El Emery, "On the Application of Artificial Neural Networks in Analyzing and Classifying the Human Chromosomes," *JCS*, vol. 2(1), Jan. 2006.
- [16] V. Kecman, "Support Vector Machines – An Introduction," in *Support Vector Machines: Theory and Applications. Studies in Fuzziness and Soft Computing*, vol. 177, L. Wang, Ed. Berlin, Heidelberg: Springer, 2005, pp. 1-47.
- [17] Illyne Saffar et al, "Machine Learning with partially labeled Data for Indoor Outdoor Detection", IEEE Annual Consumer Communications & Networking Conference (CCNC) Las Vegas USA, January 2019.
- [18] D.M.W. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," *Journal of Machine Learning Technologies*, vol. 2(1), pp. 37-63, Jan. 2011.
- [19] MATLAB Statistics and Machine Learning Toolbox: Support Vector Machine Classification, MathWorks 2019.
- [20] Deep Learning Toolbox (Formerly Neural Network Toolbox), MathWorks 2019.