

Transfer Learning for Channel Quality Prediction

Claudia Parera^{*†}, Alessandro E. C. Redondi^{*}, Matteo Cesana^{*}, Qi Liao[†] and Iliaria Malanchini[†]

^{*}DEIB, Politecnico di Milano, Milan, Italy

E-Mail: {claudia.parera, alessandroenrico.redondi, matteo.cesana}@polimi.it

[†]Nokia Bell Labs, Stuttgart, Germany

E-Mail: {qi.liao, ilaria.malanchini}@nokia-bell-labs.com

Abstract—The ability to predict the quality of a wireless channel is essential for enabling anticipatory networking tasks. Traditional channel quality prediction problems encompass predicting future conditions based on past measurements of the same channel. In this paper we study the channel quality prediction problem across different wireless channels. To this extent, we consider a reference scenario including multiple 4G cells, each of which operates on multiple concurrent frequency carriers. We propose a framework based on transfer learning to predict the channel quality of a given frequency carrier when no or minimal information is available on the very same frequency carrier for model training. For the transfer learning task we use convolutional neural networks and long short-term memory networks. We compare their performance against statistical methods on a dataset collected from a commercial 4G mobile radio network. The performance evaluation carried out on the reference dataset demonstrates the validity of the proposed transfer learning approach, achieving a root mean squared error of 0.3 on average.

I. INTRODUCTION

Anticipating the channel quality with high accuracy and low overhead can boost the proactive optimization of mobile radio networks. As examples, channel quality prediction (CQP) has been proposed for efficient resource allocation of video streaming traffic [1], interference alignment performance enhancement [2], and massive multiple-input multiple-output (MIMO) requested feedback reduction [3].

The common approach to CQP leverages past information of the channel to predict its future quality. Conversely, in this work, we address the CQP problem across different wireless channels. To this end, we focus on 4G Long Term Evolution (LTE) cells across different frequency carriers (i.e. 1.8 GHz and 2.1 GHz). The key performance indicator for channel quality used in this work is the channel quality indicator (CQI), which is an index that reflects the channel status provided by the user equipments (UEs) to their respective cells, ranging from 0 to 15 in LTE.

We introduce a transfer learning framework to predict the CQI across different channels. Two main scenarios are considered: (i) when knowledge is transferred across different frequency carriers of the same 4G cell and (ii) when knowledge is transferred across channels of different cells operating at the same frequency carrier.

The proposed framework can be conveniently used by a network operator to make educated decisions in different relevant situations:

- Network optimization/management: Only a sub-Gig frequency carrier is active at one cell, and the network operator needs to decide whether to activate higher layers by anticipating their expected quality.
- Radio resource/energy management: The carriers at higher frequencies at a cell perform duty cycling for energy management purposes and the network operator needs to decide when to switch them on/off.

The performance of the proposed transfer learning approach is validated against classical machine learning and statistical methods. The dataset used for the experimental part of this study consists of CQI values collected over a time period of one month from a commercial mobile radio network deployed in a medium-sized city in Northern Italy.

The main contributions of this paper are as follows:

- We provide deep learning architectures that significantly outperform statistical methods, showing the high non-linearity of the CQI.
- We introduce transfer learning to carry out predictions across different cells and frequency carriers. Our transfer learning methods significantly outperform traditional machine learning methods when the amount of samples available from the target cells is limited.

The rest of this paper is organized as follows: Section II reviews previous work in the area of CQP with an emphasis on the ones using deep learning. Section III describes the dataset as well as the preprocessing steps followed. Section IV focuses on the proposed deep transfer learning approach. Experiments and discussion of the obtained results are reported in Section V. Finally, Section VI summarizes the main contributions of this paper and describes future research directions.

II. RELATED WORK

CQP techniques have been widely investigated in multi-fold network environments; either to take advantage of future link improvements or to counter bad conditions before they impact the system [4].

In the field of cognitive radio networks, traditional approaches for CQP consist of collecting wireless channel statistics and carrying out predictions by applying conventional methods, such as Wiener filters, cubic spline extrapolation and short-term average [5]. Other studies exploit the nonlinear characteristics of the channel. For instance, in [6] the spectrum sensing process is modeled as a non-stationary Hidden Markov

Model. In [7], spatial and temporal correlation are taken into account to model the CQP as a multivariate Gaussian Process.

More recent approaches include the application of deep learning by modeling the CQP problem as a supervised machine learning problem. For instance, [8] uses Taguchi optimization, and long short-term memory networks (LSTMs) for spectrum prediction, specifically for channel quality as well as channel occupancy. A similar approach has been proposed in [9] for 5G, where convolutional neural networks (CNNs) and LSTMs are used for making predictions. For a comprehensive overview on channel quality prediction the interested reader may refer to the survey in [4].

In general, most of the aforementioned works leverage past channel quality to anticipate future channel quality. Conversely, we target the case where limited or no data is available on the channel we wish to predict. We show that taking data from a given channel can be used to make predictions about a different channel.

III. PROBLEM STATEMENT AND BACKGROUND

In this work we propose machine learning approaches to predict wireless channel quality in 4G LTE networks.

The reference scenario includes multiple LTE cells hosting two frequency carriers, operating at 1.8 GHz and 2.1 GHz respectively. The problem at hand can be defined as predicting the CQI of the downlink wireless channel operating at a given frequency, with no or limited available data from the same channel. Specifically, we focus on two sub-problems:

- *Intra-cell* CQI prediction, where CQI values of a given frequency carrier are forecast mainly from CQI values of a different frequency carrier of the same cell. As an example, the CQI of the carrier at 1.8 GHz is leveraged to predict the CQI of the carrier at 2.1 GHz.
- *Inter-cell* CQI prediction, where CQI values of a given frequency carrier are forecast from CQI values of a different cell, working at either the same or a different frequency carrier.

In both of the aforementioned problems, we consider the cases where no or limited information on the carrier to be predicted is available for model training.

To address these problems, we leverage a dataset obtained from a commercial 4G network deployed in a medium-sized city in Northern Italy. It includes channel quality information for 5 multi-carrier LTE cells. In detail, the dataset contains 10 time series for 5 different cells working at two different frequencies (i.e., 1.8 and 2.1 GHz, respectively). Each time series element reports the hourly average of the CQI. The total amount of data is equivalent to 583 CQI measurements for each series. The data was recorded between January 8, 2017 and February 1, 2017, for a total of 24 days and 7 hours.

The reference dataset was preprocessed according to the following steps:

- Missing value and outlier detection: No missing values and outliers were found.
- Stationary assessment: Most of the methods for time series forecasting work under the assumption that the

time series is stationary. By using Dickey-Fuller [10] and KPSS [11] statistics tests, we found that in the majority of cases the data was already stationary. However, to avoid non-stationary cases, a first order difference transformation is carried out to the whole dataset.

A. Notation

Let $(x_i^{cf})_{i=0}^{T-1} = \{x_0^{cf}, x_1^{cf}, \dots, x_{T-1}^{cf}\}$ be the sequence of CQI values obtained for cell $c \in \mathcal{C}$ in a frequency carrier $f \in \mathcal{F}$ during T hours. Without loss of generality $\mathcal{C} = \{A, B\}$ contains arbitrary pairs of cells used as target and source domains for the transfer learning task. Similarly, $\mathcal{F} = \{1, 2\}$ contains the different frequency carriers. Hereafter, we use 1 and 2 to denote 1.8 GHz and 2.1 GHz, respectively.

The CQP problem can be formalized as follows: given no or a limited amount (i.e. $t - l$) of CQI observations from a target frequency carrier f and a target cell c , $(x_i^{cf})_{i=l}^{t-1}$, we aim to forecast future CQI values $(\hat{y}_i^{cf})_{i=t}^{t+N}$, where N denotes the forecasting horizon. We leverage CQI observations from a different carrier $f' \neq f$ of the same cell c or from a set of the same or/and different carriers $\mathcal{F}' \subseteq \mathcal{F}$ of a different cell $c' \neq c$.

IV. PREDICTION APPROACHES

In this section, we describe the prediction approaches adopted in this work, including baseline algorithms used as benchmarks, the deep and transfer learning methods as well as all the considered prediction scenarios.

A. Baselines

1) *Average*: Every CQI sample in $(\hat{y}_i^{cf})_{i=t}^{t+N}$ is predicted as the average value of previous observations at the same time during the same season in $(x_i^{cf})_{i=0}^{t-1}$, given by $\hat{y}_t^{cf} = \frac{1}{H} \sum_{i=1}^H x_{t-iw}$. In our case $w = 24$ since we assume a season of 24 hours. Let T_{tr} denote the number of samples used for training chosen from the previous observations $(x_i^{cf})_{i=0}^{t-1}$, and let $H = T_{tr}/w$ represent the amount of samples to average.

2) *Auto ARIMA*: This is the automatic implementation of Auto-Regressive Integrated Moving Average (ARIMA) introduced by Jenkins in [12]. Seasonal ARIMA models are usually denoted by $ARIMA(p, d, q)(P, D, Q)_m$, where coefficients p, d, q are the order of the autoregressive model, the degree of differencing, and the order of the moving-average model, respectively. m refers to the number of periods in each season and P, D, Q refer to the autoregressive, differencing, and moving average terms of the seasonal part of the model, respectively. By using grid search we optimize the ARIMA coefficients (p, d, q, m, P, D, Q) that fit our data.

B. Deep Learning

The deep learning pipeline consists of the following steps: training, testing and performance evaluation. First, the general time series forecasting problem, is re-framed as a supervised machine learning problem. For this purpose, we use sliding windows of size $w = 24$ by shifting the original time series one step to the right T times. Fig. 1 shows the process in

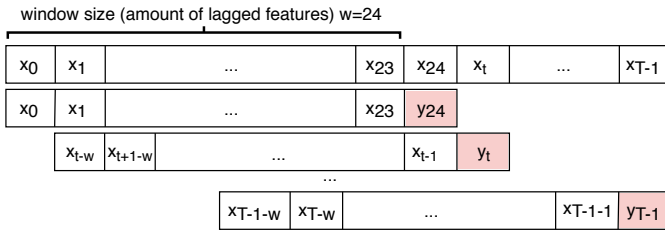


Fig. 1: Sliding windows for time series forecasting

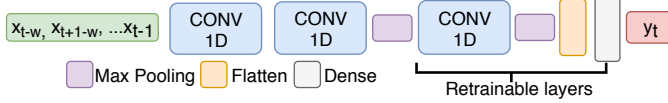


Fig. 2: CNN architecture

details. The resulting supervised machine learning problem is defined as finding the function $g(\mathbf{x}, \boldsymbol{\theta}) = y$, $\mathbf{x} \in \mathbb{R}^w$ and $y \in \mathbb{R}$ that maps w hours of CQI observations to the CQI value on the next hour. Each input vector \mathbf{x} is given by the subsequence $(x_i^{c_f})_{i=t-w}^{t-1}$, y is given by $y_t^{c_f}$ and $\boldsymbol{\theta}$ represent the neural network weights. Then the data is divided into training, cross validation and test sets. Before training, the data is scaled between -1 and 1 by using a min-max scaler, which is fitted to the training set and applied to the cross validation and test sets. It is worth noting that the first order difference and scaling transformations should be reversed before evaluating the performance of our algorithms. During training, a model $g(\mathbf{x}, \boldsymbol{\theta})$ is created by fitting the selected architecture on the training set. The cross validation set is used at a later stage by the network to select a good combination of hyper-parameters. At testing time, $g(\mathbf{x}, \boldsymbol{\theta})$ is applied to data coming from the same or a different frequency carrier. Finally, the data is projected to the original space by reversing the scaling and first order difference transformations.

1) *CNN*: CNNs [13] have significantly improved the state of the art in computer vision. They have shown remarkable performance as feature extractors, when using hierarchical data. This means, the lower layers of the network can extract general patterns from the inputs, while the deeper layers extract the more specific ones. Inspired by this fact, we first evaluate the use of CNNs for a general time series forecasting problem and in a second step we use the same architecture for the transfer learning approach. In both cases, the reference CNN architecture is comprised of 7 layers; the first 5 layers of the network are a combination of one dimensional (1D) convolutional layers followed by a 1D max pooling layer. Finally, a flatten layer as well as a dense layer are stacked for producing the final output. We apply rectified linear unit (ReLU) nonlinear transformations as the activation function. Fig. 2 shows the architecture in detail, as well as the model inputs and outputs. We use 256 filters and a kernel size of 3. For the max pooling layers we use stride $s = 2$ and pool size $\delta = 2$. The dimension of the first convolutional layer corresponds to the dimension of the feature space, which is the window size $w = 24$, in our case.

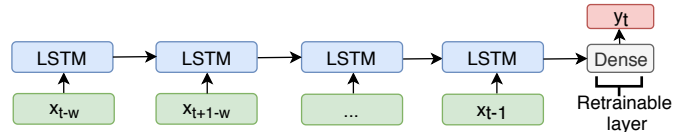


Fig. 3: LSTM architecture

2) *LSTM*: LSTMs [14] have shown great performance in natural language processing. Their success lies in the fact that they are recurrent neural networks containing multiple logical gates and are capable of learning long term dependencies in sequence predictions. In this paper, we use a many-to-one LSTM architecture, which is comprised of 1 LSTM layer followed by 1 dense layer that produces the final output. Fig. 3 shows the unrolled architecture in time, as well as the inputs and outputs. We use subsequences with a length equal to the selected window size w to predict the next sample in the sequence as inputs. By using $w = 24$ as the dimension of the feature space, the model will not capture time correlation patterns longer than 24 hours. For this purpose, we use the stateful LSTM implementation in Keras [15] that returns the current sequence in each training step. The returned sequence is then used in the next iteration allowing us to capture time correlation patterns longer than 24 hours.

C. Transfer Learning

For the transfer learning task we first train a model on the source domain, then we create a new model by taking the previous model, freezing its first layers and adding new layers that will be randomly initialized. Finally, we retrain on the target domain.

For the CNN architecture, we freeze the first two convolutional layers and randomly initialize the last one (Fig. 2). We transfer more general features learned by the first layers of the network in a richer source domain (i.e. longer time series) to a limited target domain (i.e. shorter time series). For fine tuning we keep the same architecture. The LSTM is used as feature extractor since we just retrained the last layer of the network, which is equivalent to train a linear classifier on top of a pretrained LSTM (Fig. 3). We transfer the long term patterns learned in a richer domain to a limited target domain.

D. Scenarios

In this section we describe the prediction scenarios considered in this paper. We introduce two general approaches depending on data availability in the target domain: 1) intra-cell CQI prediction and 2) inter-cell CQI prediction.

Table I summarizes all the considered forecasting scenarios as well as the inputs and outputs in every case. Let $P^{c_{f'} \rightarrow c_f}$ be the scenario where past CQI values of frequency f' and cell c' (hereafter referred to as “data $c_{f'}^c$ ”) are used to predict future CQI values of data c_f . When using transfer learning methods, we use the operator “+” to denote the retraining stage for fine tuning. For instance, let $P^{c_{f'} + \hat{c}_{\hat{f}} \rightarrow c_f}$ denote the scenario where the model pretrained on data $c_{f'}^c$ is retrained using data $\hat{c}_{\hat{f}}$, where $f', \hat{f} \in \mathcal{F}$, and $c', \hat{c} \in \mathcal{C}$. Along similar lines, multiple stages of retraining can be considered, denoted

TABLE I. Considered Scenarios

Problem	Notation	Inputs	Fine tuning	Outputs	Algorithms
Intra-cell CQI Prediction	$P^{A_2 \rightarrow A_2}$	$(x_i^{A_2})_{i=l}^{t-1}$	-	$(\hat{y}_i^{A_2})_{i=t}^{t+N}$	S Average, S Auto ARIMA, S CNN, S LSTM
	$P^{A_1 \rightarrow A_2}$	$(x_i^{A_1})_{i=0}^{t-1}$	-	$(\hat{y}_i^{A_2})_{i=t}^{t+N}$	BS Auto ARIMA, BS CNN, BS LSTM
	$P^{A_1+A_2 \rightarrow A_2}$	$(x_i^{A_1})_{i=0}^{t-1}$	$(x_i^{A_2})_{i=l}^{t-1}$	$(\hat{y}_i^{A_2})_{i=t}^{t+N}$	TL CNN, TL LSTM
Inter-cell CQI Prediction	$P^{B_2 \rightarrow A_2}$	$(x_i^{B_2})_{i=0}^{t-1}$	-	$(\hat{y}_i^{A_2})_{i=t}^{t+N}$	BS Auto ARIMA, BS CNN, BS LSTM
	$P^{B_2+A_2 \rightarrow A_2}$	$(x_i^{B_2})_{i=0}^{t-1}$	$(x_i^{A_2})_{i=l}^{t-1}$	$(\hat{y}_i^{A_2})_{i=t}^{t+N}$	TL CNN, TL LSTM
	$P^{B_1+B_2 \rightarrow A_2}$	$(x_i^{B_1})_{i=0}^{t-1}$	$(x_i^{B_2})_{i=0}^{t-1}$	$(\hat{y}_i^{A_2})_{i=t}^{t+N}$	P TL CNN, P TL LSTM
	$P^{B_1+B_2+A_2 \rightarrow A_2}$	$(x_i^{B_1})_{i=0}^{t-1}$	$(x_i^{B_2})_{i=0}^{t-1}, (x_i^{A_2})_{i=l}^{t-1}$	$(\hat{y}_i^{A_2})_{i=t}^{t+N}$	P TL CNN, P TL LSTM

by multiple “+” operations. Note that the time indices are omitted in the aforementioned notations for brevity. Overall, values on the left side of the arrow are related to the training phase, using observations taken before the time t ; whereas values on the right side are related to the forecasting step, beginning on time t .

Without loss of generality, let us define $c = A$ and $f = 2$ as the target cell and target frequency on which the CQI values are to be predicted, respectively. For different scenarios we select different data $c', \hat{c} \in \mathcal{C} := \{A, B\}$ and $f', \hat{f} \in \mathcal{F} := \{1, 2\}$ for training or fine tuning. Below we describe in detail the scenarios listed in Table I.

1) *Intra-cell CQI Prediction*: The goal is to predict future CQI values of A_2 , given all past CQI observations from the same cell and a different frequency (i.e. data from A_1) but no or a limited amount of data from A_2 .

No data available from A_2 for training: $P^{A_1 \rightarrow A_2}$ corresponds to the case where the higher frequency carrier was never activated. Therefore, there is no data available from cell A working at frequency 2. We forecast $(\hat{y}_i^{A_2})_{i=t}^{t+N}$ by training a model using $(x_i^{A_1})_{i=0}^{t-1}$. We assume that CQI values coming from layers working at different frequencies follow the same distribution.

Limited amount of data available from A_2 for training: Here the higher frequency carrier has been turned on a number of days ago, and a limited amount of data A_2 is available for training. The first solution is to use the limited data from A_2 to predict future CQI values from A_2 . However, if the time correlation is longer than the limited observed period, this model will fail to capture this long time correlation. This is why we extend our approach to use all the past observations from the lower frequency A_1 plus the limited amount of observations from A_2 . Here, we introduce the concept of transfer learning across network layers working at different frequencies. Both approaches are explained below:

- $P^{A_2 \rightarrow A_2}$: We use past CQI observations $(x_i^{A_2})_{i=l}^{t-1}$ from the same cell and frequency carrier to predict $(\hat{y}_i^{A_2})_{i=t}^{t+N}$.
- $P^{A_1+A_2 \rightarrow A_2}$: This is the transfer learning case. We use sufficient data $(x_i^{A_1})_{i=0}^{t-1}$ from frequency 1, and limited data $(x_i^{A_2})_{i=l}^{t-1}$ from frequency 2. We first train $g^{A_1 \rightarrow A_1}$ on $(x_i^{A_1})_{i=0}^{t-1}$. Then, we create $g^{A_1+A_2 \rightarrow A_2}$ and retrain it with the limited data $(x_i^{A_2})_{i=l}^{t-1}$. Finally, we use the

retrained $g^{A_1+A_2 \rightarrow A_2}$ to carry out the prediction.

2) *Inter-cell CQI Prediction*: Here we consider the scenarios where training samples come from a different cell B , under the assumptions that cell B has both frequency carriers active.

No data available from A_2 for training: If the higher frequency carrier of cell A was never activated, an alternative is to rely solely on data collected from another cell B . Here we have two different options:

- $P^{B_2 \rightarrow A_2}$: We learn $g^{B_2 \rightarrow B_2}$ using $(x_i^{B_2})_{i=0}^{t-1}$ and apply the trained model to predict $(\hat{y}_i^{A_2})_{i=t}^{t+N}$. This approach assumes that data coming from cells in different locations working at the same frequency follows the same distribution.
- $P^{B_1+B_2 \rightarrow A_2}$: We aim at applying the complete behavior of a cell B to the target cell A using all data available from B . Using $(x_i^{B_1})_{i=0}^{t-1}$ and $(x_i^{B_2})_{i=0}^{t-1}$ we learn the model $g^{B_1+B_2 \rightarrow B_2}$ and apply it to predict $(\hat{y}_i^{A_2})_{i=t}^{t+N}$.

Limited amount of data available from A_2 for training: In case where $(x_i^{A_2})_{i=l}^{t-1}$ is available from A_2 ; either we use $(x_i^{A_2})_{i=l}^{t-1}$ to predict $(\hat{y}_i^{A_2})_{i=t}^{t+N}$ as before, or we introduce transfer learning; this time across cells. Below we explain the transfer learning approaches across cells:

- $P^{B_2+A_2 \rightarrow A_2}$: We learn $g^{B_2 \rightarrow B_2}$ in cell B working at frequency 2. We create $g^{B_2+A_2 \rightarrow A_2}$ by taking $g^{B_2 \rightarrow B_2}$ and retraining with the limited data from A_2 $(x_i^{A_2})_{i=l}^{t-1}$. Finally, we use the retrained model $g^{B_2+A_2 \rightarrow A_2}$ to predict $(\hat{y}_i^{A_2})_{i=t}^{t+N}$.
- $P^{B_1+B_2+A_2 \rightarrow A_2}$: We aim at transferring the whole behavior from B to A . We learn $g^{B_1+B_2 \rightarrow B_2}$ in B and transfer this model to A_2 by retraining $g^{B_1+B_2 \rightarrow B_2}$ on A_2 to derive $g^{B_1+B_2+A_2 \rightarrow A_2}$.

V. NUMERICAL RESULTS

In this section we describe the setup of the experiments and discuss the performance of intra-cell and inter-cell CQI predictions with the approaches provided in Section IV.

A. Experimental Setup

The reference dataset comprised of 583 CQI samples across time (Section III) is divided in training, cross validation and test sets containing 535, 24 and 24 samples, respectively. The

TABLE II. Hyper-parameters

Hyper-parameters	CNN, TL-CNN	LSTM, TL-LSTM
Batch size	128	1
Number of epochs	300 early stopping	300
Early stopping	patience = 3	-
Total number of layers	7	2
Activation function	ReLU	ReLU
Optimizer	Adam	Adam
Loss	mean squared error	mean squared error
Learning rate	0.001	0.001
Dropout	0.2	0.2

different prediction approaches encompass different parameters, which require fine tuning for further optimization. The Average method is parameter free; whereas Auto ARIMA requires $m = 24$ to be set *a priori* (Section IV-A2). As for CNNs and LSTMs, the use of different window sizes $\{1, 6, 12, 24\}$ was explored. Manual cross validation is carried out to choose a good architecture for this problem. We tried different batch sizes in the set $\{1, 13, 43, 128\}$. Table II shows the final selection of parameters and hyper-parameters after cross validation for each model.

The prediction accuracy is measured by the root mean squared error (RMSE) between real and predicted values in the test set (with size $T_{\text{test}} = N$) defined by Eq. 1:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=t}^{t+N} (\hat{y}_i - y_i)^2}. \quad (1)$$

In the following sections we show the average RMSE across cells when changing the amount of days taken from the target domain for model training or fine tuning.

B. Intra-cell CQI Predictions

Fig. 4 shows the average RMSE when applying intra-cell CQI prediction (Table I). The x-axis shows the amount of CQI samples (measured in days) available from the target frequency that were used for training or fine tuning the models accordingly. The S curves refer to cases where the proposed methods are applied by leveraging only data available from the same frequency ($P^{A_2 \rightarrow A_2}$). The TL curves are related to the transfer learning scenarios, where the model is pretrained on source data coming from A_1 , then fine tuned on target data coming from A_2 ($P^{A_1+A_2 \rightarrow A_2}$). Finally, the points labeled as BS refer to cases where there is no data available from the target frequency for training or fine tuning; thus the only data available is from a lower frequency ($P^{A_1 \rightarrow A_2}$). We can draw the following conclusions:

- On average we achieve a RMSE of 0.3. This is reasonable since CQI values are in the range of 0 to 15 with very low variance.
- The deep learning approaches (i.e. S-CNN, S-LSTM, TL-CNN and TL-LSTM) outperform Auto ARIMA and Average methods in all cases. The conjecture here is that the deep learning methods better capture the CQI time series non-linearity.
- Overall, LSTMs outperform CNNs when we have at least some data from the target frequency for training and fine

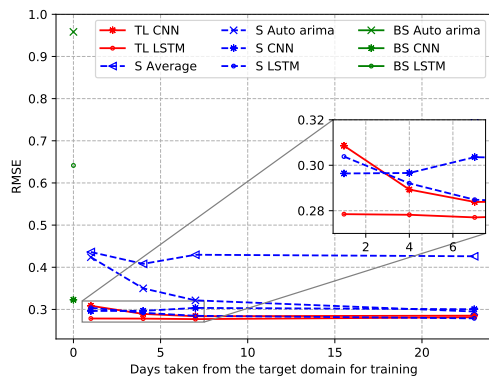


Fig. 4: RMSE, Intra-cell CQI predictions

tuning. The reason is that LSTMs are better at capturing long time correlations than CNNs. However, when no data is available from the target domain, CNN performs better.

- The transfer learning methods outperform traditional machine learning methods when the amount of data taken from the target frequency for training or fine tuning decreases specially when using LSTMs.
- In general the transfer learning algorithms (TL) significantly outperform the cases where we do not use data of the target domain (BS). Therefore, using a limited amount of data for fine tuning helps to improve the performance.

C. Inter-cell CQI Predictions

In this section, we analyze numerically the performance of inter-cell CQI prediction approaches (Table I), where we use as source domain a different cell B at the same frequency ($P^{B_2 \rightarrow A_2}$ and $P^{B_2+A_2 \rightarrow A_2}$), or the source domain includes information on all the frequencies ($P^{B_1+B_2 \rightarrow A_2}$ and $P^{B_1+B_2+A_2 \rightarrow A_2}$). Fig. 5 shows the average RMSE over the same 4 cells used shown before. It is worth noting that the fifth cell of the dataset is not shown in the results and just used as source domain for the inter-cell case. We can observe the following:

- The performance features a trend similar to the one of the intra-cell CQI prediction case with RMSE close to 0.3. Similarly, the deep learning approaches outperform the baseline methods and the LSTMs perform better than CNNs, when there is limited data from the target frequency to be used for training or fine tuning. Otherwise, when no data is available from the target frequency, CNN methods perform better.
- The transfer learning algorithms (TL-CNN, TL-LSTM, P-TL-CNN and P-TL-LSTM) outperform the traditional machine learning approaches, specially when the amount of data available from the target frequency is scarce.
- The model transfer approaches (P-TL-CNN and P-TL-LSTM) outperform the transfer learning models (TL-CNN, TL-LSTM), specifically when the amount of data from the target frequency available for training is small. Therefore, transferring the whole behavior from one cell to another can help to improve the prediction performance

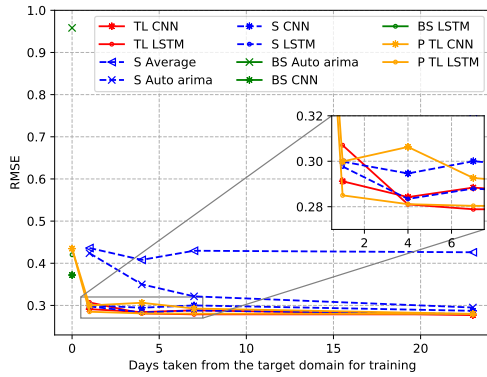


Fig. 5: RMSE, Inter-cell CQI predictions

when the amount of data from the target domain is limited.

D. Discussion

The above mentioned numerical results suggest that, among all the proposed algorithms, the LSTM performs the best. This is expected since the LSTM is able of capturing longer time correlation than the CNN. Overall, the proposed LSTM based methods achieve a low RMSE under 0.3. This is partially given by the low variance of the CQI that is averaged hourly and across users in each cell. Fig. 6 summarizes all the methods that use an LSTM. The solid lines show the intra-cell prediction approaches; whereas the dashed lines show the inter-cell prediction approaches.

- When the amount of data available from the target frequency is limited, the best approach is using intra-cell transfer. Otherwise using data from the same frequency leads to better results. Overall, the intra-cell approaches outperform the inter-cell ones.
- If intra-cell transfer cannot be performed due to the unavailability of the data in source cell, the inter-cell approach can still be used since the difference between P-TL-LSTM and intra TL-LSTM is small.

VI. CONCLUSIONS

We proposed a transfer learning framework to solve the CQP problem in the challenging cases where no or limited information is available on the channel we wish to predict. The proposed framework was tested on a dataset from a commercial LTE network, showcasing how transfer learning can be carried out across pairs of cells working at different frequencies, or at the same frequency in different locations. The obtained results show that the proposed deep transfer learning methods are particularly effective when the amount of data available from the target channel to be predicted is limited.

This is a preliminary study towards applying transfer learning to CQI forecasting. Immediate follow ups will include the application of this approach to other network key performance indicators as well as the introduction of a “sophisticated” selection criteria to identify the cells which should be used as a source domain (wireless channels and cells) for the transfer

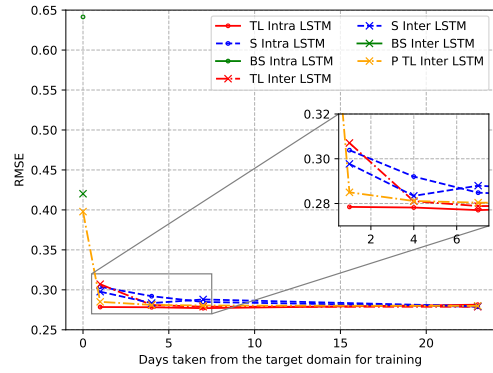


Fig. 6: RMSE, Frequency vs. location

learning task. Future work will also include optimization and fine tuning of the deep learning models, and the evaluation of different deep neural network architectures.

ACKNOWLEDGEMENT

This work is funded by the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 643002.

REFERENCES

- [1] D. Tsilimantos, A. Nogales-Gómez, and S. Valentin, “Anticipatory radio resource management for mobile video streaming with linear programming,” in *IEEE ICC*, 2016, pp. 1–6.
- [2] A. Dong, H. Zhang, and D. Yuan, “Achievable rate improvement through channel prediction for interference alignment,” in *IEEE APCC*, 2013, pp. 293–298.
- [3] R. Nagashima, T. Ohtsuki, W. Jiang, Y. Takatori, and T. Nakagawa, “Channel prediction for massive MIMO with channel compression based on principal component analysis,” in *IEEE PIMRC*, 2016, pp. 1–6.
- [4] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, “A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1790–1821, 2017.
- [5] X. Xu, M. Ni, and R. Mathar, “Improving QoS by predictive channel quality feedback for LTE,” in *IEEE SoftCOM*, 2013, pp. 1–5.
- [6] X. Xing, T. Jing, Y. Huo, H. Li, and X. Cheng, “Channel quality prediction based on bayesian inference in cognitive radio networks,” in *IEEE INFOCOM*, 2013, pp. 1465–1473.
- [7] Q. Liao, S. Valentin, and S. Stańczak, “Channel gain prediction in wireless networks based on spatial-temporal correlation,” in *IEEE SPAWC*, 2015, pp. 400–404.
- [8] L. Yu, J. Chen, G. Ding, Y. Tu, J. Yang, and J. Sun, “Spectrum prediction based on Taguchi method in deep learning with long short-term memory,” *IEEE Access*, vol. 6, pp. 45 923–45 933, 2018.
- [9] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, “Channel state information prediction for 5G wireless communications: A deep learning approach,” *IEEE Transactions on Network Science and Engineering*, 2018.
- [10] D. A. Dickey and W. A. Fuller, “Distribution of the estimators for autoregressive time series with a unit root,” *Journal of the American statistical association*, vol. 74, no. 366a, pp. 427–431, 1979.
- [11] D. Kwiatkowski, P. C. Phillips, P. Schmidt, and Y. Shin, “Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?” *Journal of econometrics*, vol. 54, no. 1-3, pp. 159–178, 1992.
- [12] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [14] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [15] F. Chollet *et al.*, “Keras,” <https://github.com/fchollet/keras>, 2015.