IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 18 October 2023, accepted 7 December 2023, date of publication 18 December 2023, date of current version 3 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3344317

RESEARCH ARTICLE

ALPACA: An Asymmetric Loss Prediction Algorithm for Channel Adaptation Based on a Convolutional-Recurrent Neural Network in URLLC Systems

KIRILL GLINSKIY, ALEKSEY KUREEV, AND EVGENY KHOROV, (Senior Member, IEEE)
Wireless Networks Laboratory, Institute for Information Transmission Problems of the Russian Academy of Sciences (IITP RAS), 127051 Moscow, Russia
Telecommunications Systems Lab, HSE University, 101000 Moscow, Russia

Corresponding author: Evgeny Khorov (khorov@frtk.ru)

The research has been carried out at HSE University and supported by the Russian Science Foundation (Grant No 21-79-10158, https://rscf.ru/en/project/21-79-10158/).

ABSTRACT A key feature of 5G systems is the Ultra-Reliable Low-Latency Communication (URLLC), which can be used for remote surgery, smart grids, industrial control, etc. URLLC requires millisecond-level delays and very high reliability, i.e., less than 10^{-5} packet loss probability. The ability to satisfy these very strict quality of service requirements depends on selecting the Modulation and Coding Schemes (MCS) for data transmissions. On the one hand, the selected MCS shall be robust enough to avoid multiple retransmissions within a small delay budget. On the other hand, the MCS shall be high-rate to reduce channel resource consumption and, thus, shall increase the system capacity for URLLC. The MCS selection problem is extremely challenging to capture the quickly varying wireless channel effects, e.g., in highly mobile scenarios, because the decision shall be made long before the actual transmission occurs. The paper proposes a novel MCS selection algorithm called ALPACA (Asymmetric Loss Prediction Algorithm for Channel Adaptation), which relies on a widely used class of convolutional-recurrent neural networks. However, in contrast to existing approaches, ALPACA explicitly considers the asymmetric error cost for channel prediction by utilizing quantile regression loss. Both real-life channel measurements and 3GPP channel models are used to evaluate the performance of ALPACA. Numerical results demonstrate the increase in the reliability and reduction in resource consumption compared with the existing MCS selection algorithms, which results in 40% growth of the network capacity.

INDEX TERMS Channel prediction, deep learning, MCS selection, quantile regression loss, URLLC.

I. INTRODUCTION

Being a key feature of 5G systems, Ultra-Reliable Low-Latency Communication (URLLC) is needed for such applications as unmanned vehicles, remote surgery, and smart grids, where the cost of a missed deadline or lost transmission can be very high. These applications impose very strict quality of service (QoS) requirements, e.g., delays below ten milliseconds and packet loss ratio below 10^{-5}. Providing low latencies and high reliability strongly depends on selecting the appropriate Modulation and Coding Scheme (MCS). The MCS selection algorithm shall select a sufficiently robust MCS for the reliability goal, while keeping the resource consumption down, making the selection of only the lowest MCS unsuitable. In mobile scenarios, the channel quality changes rapidly, and the channel state information obtained at a time may become obsolete when the scheduled transmission occurs. Therefore, the algorithm that selects an appropriate MCS shall not only adapt to the current channel condition but also predict the channel quality changes in the future. In the literature, many approaches have been proposed to address the channel prediction problem. A group of algorithms introduces clear assumptions of channel properties, proposes a channel model, collects some statistics

The associate editor coordinating the review of this manuscript and approving it for publication was Ivan Wang-Hei Ho.

from tuning it, and then solves some channel prediction optimization problem [1]. Despite their interpretability, these algorithms rely on the introduced assumption about channel behavior, and they may not be suitable if, in some scenarios, the assumptions are not held. Additionally, these simple heuristics and models have limited ability to be generalized and to be used for different user speeds. This problem limits the effectiveness of simpler approaches, forcing authors to supplement these algorithms with additional functions, like periodically modifying the transmit frame structure [2], adding model parameter tracking [3].

That is why algorithms based on neural networks (NN) are often considered as the most promising [4], [5], [6], [7], [8], [9] because they are capable of extracting useful channel behavior patterns *automatically* from training data, i.e., channel measurements. Unfortunately, NN-based approaches can hardly find the exact solution for some optimization problems. However, the objective can be expressed in the form of a loss function. Specifically, in contrast to eMBB, for URLLC, packet losses and missed deadlines are much more dangerous than channel resource waste caused by selecting a bit lower MCS, therefore resulting in different rewards for various types of errors in channel prediction. To the best of our knowledge, this issue has not been considered in existing channel-prediction approaches, which motivates us for this study.

Naturally, the performance of various channel prediction methods significantly depends on the scenario and the testing data: various approaches shall be compared under the same conditions. However, few datasets containing channel measurements exist. Most of them cannot be used for evaluating MCS selection algorithms for URLLC because they do not provide enough high-granularity measurements and/or consider only static or low-mobility scenarios [10], [11], [12].

The key goal of the paper is to design a CQI prediction approach that can compensate changes in the channel quality between the channel measurements and the real data transmission.

The contribution of this paper is two-fold.

First, we introduce an MCS selection algorithm called ALPACA (Asymmetric Loss Prediction Algorithm for Channel Adaptation) that considers the strict QoS requirements of the URLLC systems. ALPACA is based on a deep convolutional-recurrent neural network that predicts the channel quality. In contrast to existing works, ALPACA uses a specific asymmetric loss function in the form of quantile regression loss, which explicitly addresses the problem of different outcomes of under- and overestimating the channel quality, as the former increases the channel resource consumption, while the latter results in losses crucial for the URLLC systems.

Second, to accurately evaluate the performance of the ALPACA, we collect real channel measurements in several 3GPP-like scenarios. Then we use a hybrid simulation based on (i) the ns-3 [13] network simulator that emulates the operation of the 5G protocol stack and (ii) various channel traces. To obtain channel traces, we use the collected real channel measurements obtained from an original testbed. For consistency, we also evaluate the performance of the developed algorithms using the 3GPP 38.901 channel model for similar scenarios [14].

TABLE 1. List of acronyms.

CNN	Convolutional Neural Network
CRS	Cell Reference Signals
CSI	Channel State Information
CQI	Channel Quality Indicator
eNB	evolved NodeB
LSTM	Long-Short Term Memory
MCS	Modulation and Coding Scheme
MAE	Mean Absolute Error
NN	Neural Network
PLR	Packet Loss Ratio
QoS	Quality of Service
RB	Resource Block
RNN	Recurrent Neural Network
SINR	Signal to Interference and Noise Ratio
SDR	Software-Defined Radio
TB	Transport Block
UE	User Equipment
URLLC	Ultra-Reliable Low-Latency Communications

This paper is organized as follows. We examine the state-of-the-art approaches to MCS selection for URLLC and the use of NN-based algorithms for channel prediction in Section II. Then we briefly overview the channel measurement procedure and describe our methodology for dataset gathering and processing in Section III. Section IV describes the developed ALPACA algorithm. Additionally, this section contains a general description of the MCS selection pipeline and data flow. Section V describes the simulation setup and scenario, as well as analyzes the evaluation results of the proposed MCS selection algorithm. Section VI concludes this work, and Section VII analyzes the future research direction.

II. RELATED WORK

The usage of channel prediction for link adaptation receives great attention because of its significant impact on network performance. Multiple approaches have been proposed in the literature, including those based on signal processing [15] or statistical models such as linear auto-regression [2].

Recently, neural networks became extremely promising for link adaptation because they excel at handling complex non-linear tasks with a sufficient amount of data. For example, an NN-based algorithm is used to predict channel conditions in a 5G system [16], [17], [18], [19]. The algorithm is tested in many experimental scenarios, but all the measurements were collected with static or pedestrian-level mobility UEs, limiting the conclusions to semi-static channels. Similar work was carried out in [20], which proposed an NN-based algorithm for an LTE-A network, where the algorithm is trained on experimental data collected from a deployed base station operator, with transfer learning used to improve the performance of the network. A promising architecture of neural network for the channel adaptation task is the

Convolutional Neural Network (CNN), originally designed for image processing tasks, but achieving high accuracy in the channel prediction task in [21]. Another CNN, inspired by the image super-resolution technique was proposed by [22]. In [9], [17], [23], [24], [25], [26], [27], [28], and [29] a channel state information predictor is proposed based on a recurrent neural network and its modifications, such as Long-Short Term Memory (LSTM) [30], [31] network. The simulations and experimental studies confirm that the neural network can pick up and learn the channel evolution patterns in various fading channels, including the Rayleigh channel [24], as well as experimental measurements [32].

Additional modifications to the neural recurrent architecture, such as adding constant learning during deployment were proposed in [25]. Paper [27] proposed an algorithm to estimate Channel State Information (CSI) in a complex MIMO system, named CSINet, which uses an end-to-end approach for CSI representation and reconstruction. In [29], the LSTM network is trained to predict the CSI for the drone-ground data link scenario. Recently, the authors of the paper [33] proposed neural network designs with a so-called attention mechanism, which increases the weight (or importance) of some elements of the sequence and, thus, improves the prediction performance.

However, the algorithms designed in most of the papers do not consider the strict QoS requirements imposed by URLLC systems. For example, one of the arising challenges is the vastly different outcome for under- or overestimating the channel quality. These challenges are sometimes addressed by using some heuristic algorithms for channel prediction, e.g., [1], relying on the moving average and minimum approach to robustly predict the channel quality. Additionally, the parameters need to be adjusted for each different channel environment. Furthermore, the data for evaluating and testing the algorithms need to be considered while designing the channel prediction algorithms. As the NN algorithms are data-driven, the best option for closest-to-life data needs to be experimentally obtained, which is rarely used, and channel modeling data is often used instead [23].

While experimental testbeds to measure the channel properties exist, each of them possesses some drawbacks that limit its functionality for channel prediction in URLLC in mobile scenarios, which is the target of this paper. The simplest approach in the literature uses off-the-shelf devices such as smartphones for LTE channel quality measurement [10]. However, in the testbed, access to the low-level PHY measurements is limited, meaning that the unprocessed channel properties are inaccessible using this solution. Another approach is to construct the measurement setup from scratch using specific hardware and software. For example, in [11], a sophisticated testbed is used to measure the fast-fading channels. This testbed is mechanically complex, allowing high-speed antenna movements, and is not universally applicable, as moving antenna testbed is limited to static channel environments by its design. To record signals from deployed base stations and further process them is proposed in [34]. Unfortunately, this approach also does not allow an accurate estimation of the mobile channel conditions during the rapid changes because the minimum interval between measurements is too long [12].

In the next section, we address the issue with the availability of training data and describe our approach to gathering the necessary dataset.

III. DATASET

A. CHANNEL MEASUREMENT MECHANISM IN CELLULAR NETWORKS

While 5G systems are deployed in many countries, its coverage can be lacking outside very dense areas, while its predecessor, 4G, is still highly prevalent. So, we designed our testbed to be compatible with both 4G and 5G systems using the Software-Defined Radio (SDR) platform.

This design also allows us to be more flexible with our dataset creation. Specifically, the measurement dataset is collected in 4G systems because of their current abundance in the areas of interest, while 5G cells are also capable of being recorded.

The physical layers of 4G and 5G systems have much in common. For example, both use OFDM (Orthogonal Frequency Division Multiplexing). Channel resources are divided into resource blocks (RB), which are later assigned to the specific signals. In both technologies, the total number of RBs depends on the available bandwidth (in 5G, it also depends on the used numerology).

In 4G systems, the UE can estimate the channel quality using the received cell reference signals (CRS) sent by the base station. Then, as the CRS is standardized, the UE determines how the CRS degrades in the channel and therefore performs the channel estimation. After that, the receiver calculates the Signal to Interference and Noise Ratio (SINR). While all these stages are performed at the UE, the base station receives information about the channel quality via a Channel Quality Indicator (CQI), which is calculated by mapping the SINR value to an integer value from 0 to 15, where 0 is the worst signal quality, and 15 is the best one. Then the gNB uses CQI to select the MCS scheme. Therefore, the MCS selection procedure significantly depends on the knowledge about the CQI for the certain RB that can be obtained by using channel prediction algorithms.

B. EXPERIMENTAL TESTBED

Because of the lack of existing channel measurements applicable to our research, we developed a new channel measurement testbed that satisfies the following requirements:

- The testbed has to support a high granularity of channel estimation measurements for rapidly varying channels.
- The testbed has to operate with no emission in the licensed spectrum, i.e., only as a receiver.
- The testbed has to be compatible with off-the-shelf cellular base stations to obtain measurements for real-life scenarios without the need to deploy cutting-edge equipment.

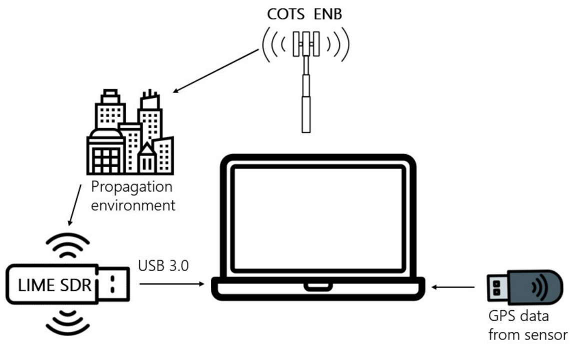


FIGURE 1. Testbed scheme.

The testbed and the overall method used in the dataset-gathering process were originally proposed in [35]. However, in this paper, they are extended to accommodate different frequency bands and bandwidths.

Figure 1 shows the scheme of the experiment. During the experiment, the signals from the commercial off-the-shelf (COTS) cells are received and processed by the SDR connected to the PC emulating the UE receiver implemented in the software framework.

We measure the channel in the following mobile scenarios:

1) The xEVA (experimental Extended Vehicular A) scenario: the testbed is located in a car that moved with a speed of approximately 40 to 60 kmph in a city center with moderate traffic and different environments, including high buildings and a park zone. Such environment shall provide variable delay spread of reflected signals
2) The xEPA (experimental Extended Pedestrian A) scenario: the testbed is carried by a pedestrian with a speed of 3 kmph in a city center.

We record parameters of movements such as speed and position via a GPS tracker attached to the testbed.

In the experiment, we record the channel state as In-phase and Quadrature (IQ) samples and process them using a modified version of the srsLTE software suite [35] in real-time. The output of the data processing pipeline is the CQI and SINR values for each RB. We obtain the sampling frequency of one CQI value per slot. This value allows tracking the channel fading with frequencies up to 1 kHz, according to Nyquist frequency. Note that the sampling frequency is significantly higher than the maximum Doppler frequency for 60 kmph and 2.6 GHz, which is below 200 Hz.

IV. ALPACA

To estimate the future CQI based on previous channel measurements, we designed an NN-based algorithm called ALPACA (Asymmetric Loss Prediction Algorithm for Channel Adaptation). ALPACA treats the channel prediction problem as a multivariate time-series prediction task. The input of the neural network is the previous CQI values for all RB in the form of a time-frequency array, and the output is the predicted CQI values for all RB in the future slot. In the time dimension, the size of the input array is 128 (meaning that with a per-TTI CQI report, the NN knows 128 previous CQI values). In the frequency dimension, the size of the array equals the number of RB groups.

A. NEURAL NETWORK ARCHITECTURE

The ALPACA NN has several layers. First, to predict the CQI, we need to extract the features from the previous measurements, which can be represented as different spatio-temporal patterns on the time-frequency grid. For that, we use convolutional neural network layers. Then, we use the LSTM layer, to obtain information about temporal dependencies. The resulting features are used by the network head, consisting of fully connected layers, to interpret the extracted features and predict the next values of the channel quality indicator accordingly. The neural network design is based on paper [16]. However, the CSI prediction task is significantly different from the CQI prediction one because, in contrast to complex CSI values, the CQI ones are integer. Consequently, we do not need the 2D convolutional layer.

Let us describe the developed neural network in detail, see Fig. 2. The first stage of the neural network is a stack of two 1D convolutional layers designed to extract frequency-specific and local features. The first layer consists of 32 filters with a kernel size of 5, while the second one contains 64 filters with the same kernel size. We use the Rectified Linear Unit (ReLU) activation function on these layers and the Batch Normalization technique [36] to improve the performance of the convolutional network.

Then the outputs of the CNN layers are passed through the LSTM layer. The LSTM layer is added to create more time-global features from the previously extracted local time-frequency patterns of the CNN, therefore increasing the receptive field of the NN, as well as improving its capabilities on the long-term prediction of rare events, which are often crucial for enabling URLLC.

The used LSTM has a hidden state size of 128. The structure of the LSTM cell is presented in Fig. 3. The parameters of the LSTM layer and the data flow can be described as follows:

$$\begin{aligned} f_t &= \sigma_g\left(W_f \times x_t + U_f \times h_{t-1} + b_f\right) \\ i_t &= \sigma_g\left(W_i \times x_t + U_i \times h_{t-1} + b_i\right) \\ o_t &= \sigma_g\left(W_o \times x_t + U_o \times h_{t-1} + b_o\right) \\ c'_t &= \sigma_c\left(W_c \times x_t + U_c \times h_{t-1} + b_c\right) \\ c_t &= f_t \cdot c_{t-1} + i_t \cdot c'_t \\ h_t &= o_t \cdot \sigma_c\left(c_t\right), \end{aligned} \tag{1}$$

where x_t is the output of the CNN layers, h_{t-1} and c_{t-1} are the inputs from the previous timestep of the LSTM, output gate $o(t)$ is the output of the LSTM for the timestep t. Additionally, on every step of LSTM, the cell state c_t, forget state f_t, and the hidden state h_t are also generated. W and U are the weight matrices for the respective gates, while b_* are the bias vectors. Finally, σ_g denotes the sigmoid nonlinearity, and

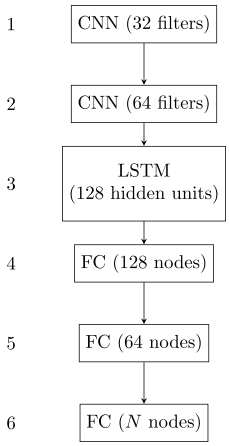


FIGURE 2. Neural network architecture.

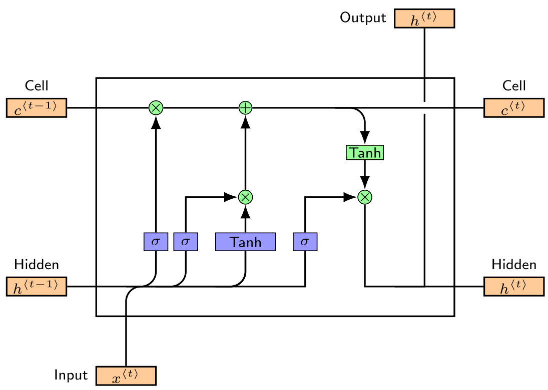


FIGURE 3. The LSTM scheme [30].

$sigma_c$ is the hyperbolic tangent. The symbol $\cdot$ denotes the element-wise multiplication (Hadamard product), and $\times$ is the matrix multiplication. We use the output vector of the cell at the last timestep to feed it to the final part of the network, as it is the final output vector that contains the important temporal information.

The final part of the network consists of three fully connected layers with intermediate ReLU activation and linear activation at the final layer, with 128, 64, and N nodes, respectively, where N is the number of RB groups for which we predict the CQI values. To prevent overfitting during training, we use Dropout, a technique that zeroes out a certain percentage of units in the LSTM layer, and the Fully-Connected (FC) layer in Fig.2. We set the dropout percentage to 0.2. This architecture means that we can predict the CQI values for an arbitrary number of steps ahead by feeding the prediction back into the neural network, with each step equal to the training data step (0.5 ms). However, with the longer prediction horizon, the prediction error also increases.

B. ASYMMETRIC LOSS DURING TRAINING

A significant problem to take into account for URLLC is that even a small overestimated prediction can lead to packet loss and delays, possibly violating the QoS requirements, while the underestimated prediction leads to a bit higher channel resource consumption. We consider a promising way to address this issue by utilizing the asymmetric loss function during the training process. To the best of our knowledge, there were no previous studies on applying asymmetric loss functions to the channel adaptation task in wireless networks. The use of asymmetric loss during training is a key feature of ALPACA, which gives the name of the designed algorithm. Specifically, we use the quantile regression loss inspired by [37] function to train the network, which can be written as

$$l = \sum_{i=0}^{N} \max(q \cdot (y_i - \hat{y}_i), (q-1) \cdot (y_i - \hat{y}_i)), \tag{2}$$

where l is the weighted error of the prediction with different weights for overestimation and underestimation, q is the quantile we aim to predict, y_i is the element of the true value vector, and $\hat{y}_i$ is the element of the neural network output vector, and the size of truth values and predictions vector is N. Unlike previous works [16], [23], which also use LSTM for channel quality prediction, we can explicitly optimize for the necessary target reliability during algorithm training and reduce the chance of overestimating the channel quality. Moreover, the proposed approach can be extended to multiple quantile output values to target multiple reliability requirements with a negligible additional incurred cost. Additionally, should the need to change the probability quantile arise, it could be done on the already pre-trained neural network using methods such as transfer learning with minimal computational overhead on the new data.

We train the created network using the stochastic gradient descent (SGD) method [38] with the Adam optimizer [39] to improve convergence. The initial learning rate of SGD is 0.001. The training spans 100 epochs. We reduce the learning rate twofold if the validation loss has not improved for five epochs, as well as introduce early stopping. To adapt the neural network for different channel conditions, we also introduce augmentations to the dataset, namely adding or subtracting a constant SINR from input and target data, simulating different path loss values in the scenario. Figure 4 shows the losses for the training and validation steps. The results show that the early stopping allows us to select the neural network state with a reasonably low test loss quite quickly.

C. SCHEDULING AND MCS SELECTION STEP

When the scheduling algorithm decides on allocating RBs to UEs, it should take into account the channel conditions in a certain subband corresponding to the moment of actual transmission, i.e., use the predictions. However, during CQI quantization some channel information is lost, as the continuous SNR values are converted to values of 0 to 15. Similarly to [1] and [23], we use the following approach to scheduling and MCS selection based on given predictions.

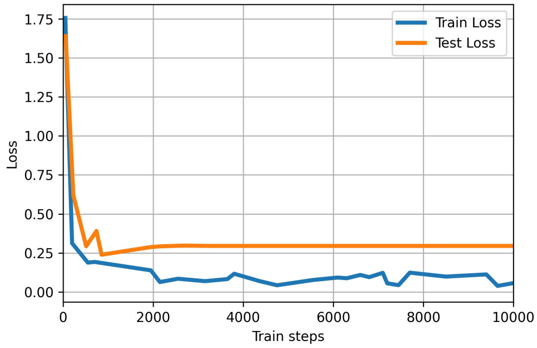


FIGURE 4. Train and test losses during training.

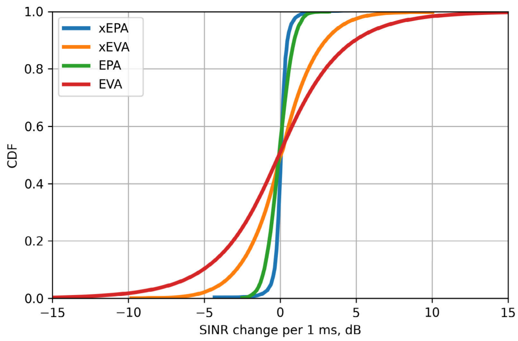


FIGURE 5. The CDF of SINR changes per 1 ms in different channels.

1) Calculate the scheduling metric for each RB and UE.
2) For each RB, sort UEs according to this metric and find the highest-rated UE according to the metric, further referred to as leaders.
3) Sort RBs in the descending order of CQIs reported by corresponding leaders.
4) Starting from the first RB (with the highest CQI), allocate RBs to the corresponding leader.
5) For each UE, calculate the maximal transport block (TB) for the transmission. All RBs assigned to the UE at Step (5) and not used are considered for further scheduling.
6) Remove from consideration the UEs that can transmit all buffered data in already allocated RBs.
7) Go to Step (3) and continue the procedure until all RBs are allocated or none of the remaining RBs can increase the TB size for any UE.

The TB size and MCS selection step can be also described algorithmically:

1) For each RB a_i from the set, we calculate the CQI estimation according to the predictions from the ALPACA neural network and map the obtained CQIs $\{CQI^1, \ldots, CQI^n\}$ to SINRs $\{SINR^1, \ldots, SINR^n\}$ using the inverted SINR to CQI table specified by the 3GPP [40]. Below we assume that RBs are sorted in the descending order of SINRs.
2) A single effective $SINR_k$ (see [41]) is calculated for each subset $A_k = \{a_1, \ldots, a_k\}, k = \{1, \ldots, n\}$.
3) Using the effective SNR_k, we find such an MCS MCS_k that allows obtaining BLER less than the given target value.
4) Assuming that MCS_k is used in all RBs, we calculate the TB size TB_k for each subset A_k.
5) We find the subset A_k providing the maximal TB size.

As these steps apply to other channel adaptation algorithms, such as [1] and [42], and the computational overhead presented by it is insignificant, as no additional computation is required by the neural network after the CQI prediction.

V. NUMERICAL RESULTS

A. PRELIMINARY ANALYSIS OF CHANNEL PROPERTIES

Before evaluating the system-layer performance of ALPACA, let us investigate the properties of the measured channels and the 3GPP channel models [43] for similar scenarios. Specifically, we consider Non-Light-Of-Sight (NLOS) Extended Pedestrian A (EPA) and Extended Vehicle A (EVA) channel models with user device velocities of 3 and 60 kmph, respectively, which should have properties similar to the experimental scenarios xEVA and xEPA.

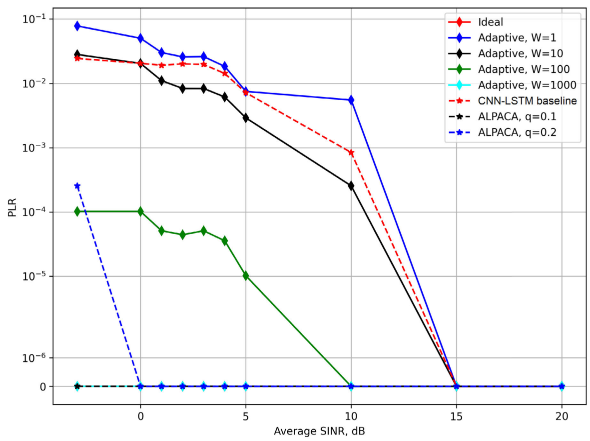
We calculate how much SINR changes with time to provide a simple overview of channels. Specifically, we measure the absolute SINR changes during every millisecond for a given RB and show the CDF of these changes in Fig. 5.

These changes grow with the Doppler frequency and, therefore, the speed of the user in the scenario. The pedestrian channel changes slightly because of its lower Doppler frequency. Moreover, as we can see, the change of SINR per one millisecond is smaller than the differences between two SINR thresholds corresponding to neighboring MCSs, meaning that transmission with the MCS computed with the last measurement is likely successful. Thus, if there were no URLLC, the usage of the last SRS measurement would work. However, for URLLC, significant rare losses are still possible, which violate QoS requirements, and thus shall be taken into account.

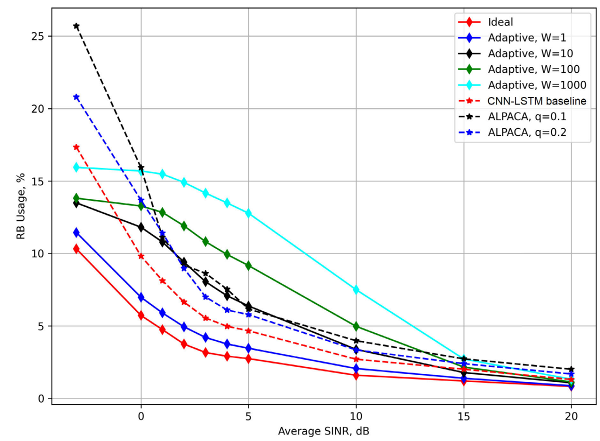
The differences in SINR changes between the 3GPP model and real-life measurements can be attributed to differences in the propagation environment, which influences the number of scattering surfaces and the way the signal is weakened and changed during propagation.

B. SIMULATION SCENARIO

To evaluate the performance of the developed algorithm at the system layer, we use the ns-3 simulation tool with channel traces obtained with measurements and channel models. The simulation scenario consists of a base station and a UE, receiving a constant bitrate URLLC traffic of 500-byte packets with the intensity of 200 packets per second. To satisfy the timing requirement of 10 ms only one HARQ retransmission attempt is allowed for each frame. After the neural network is trained on the training set, we measure the PLR, i.e., the percentage of packets not delivered in time, and the RB usage, i.e., the percentage of RBs allocated to the UE receiving URLLC packets.



(a) PLR in the xEVA scenario



(b) RB usage in the xEVA scenario

FIGURE 6. PLR (a) and RB usage (b) in the xEVA scenario.

We compare the following MCS Selection algorithms:

- Ideal algorithm, which has perfect channel knowledge and selects the MCS as the maximal one that provides an error rate below 10^{-5};
- Adaptive algorithm, described in [1], which estimates the possible channel quality degradation using a sliding window approach and makes adjustments to the prediction accordingly. The size of the sliding window W is a hyperparameter of this algorithm, and a longer window leads to more conservative predictions. We evaluate the algorithm using windows from 1 to 1000. The MCS is selected by the algorithm output, as it is an end-to-end predictor.
- The proposed algorithm, ALPACA, with asymmetric loss parameter q of 0.1, 0.2, 0.5. We note, that the quantile regression with the quantile parameter of 0.5 is equivalent to the mean absolute error (MAE) loss, which is symmetric, and forecasts the median, making this approach similar to the LSTM-based approaches described in papers [16], [23], hence we name the algorithm with $q = 0.5$ as "CNN-LSTM". The MCS is selected according to the CQI prediction to provide the 10^{-5} error rate.

For each algorithm, we evaluate PLR and the number of used RBs, i.e., RB usage, for different average values of SINR by applying an offset to all SINRs in the scenario. This approach imitates the performance for varying path loss and distance between the base station and the user. The SINR shift values belong to the interval -3 to 20 dB as the typical values for cellular systems [44].

C. SIMULATION RESULTS

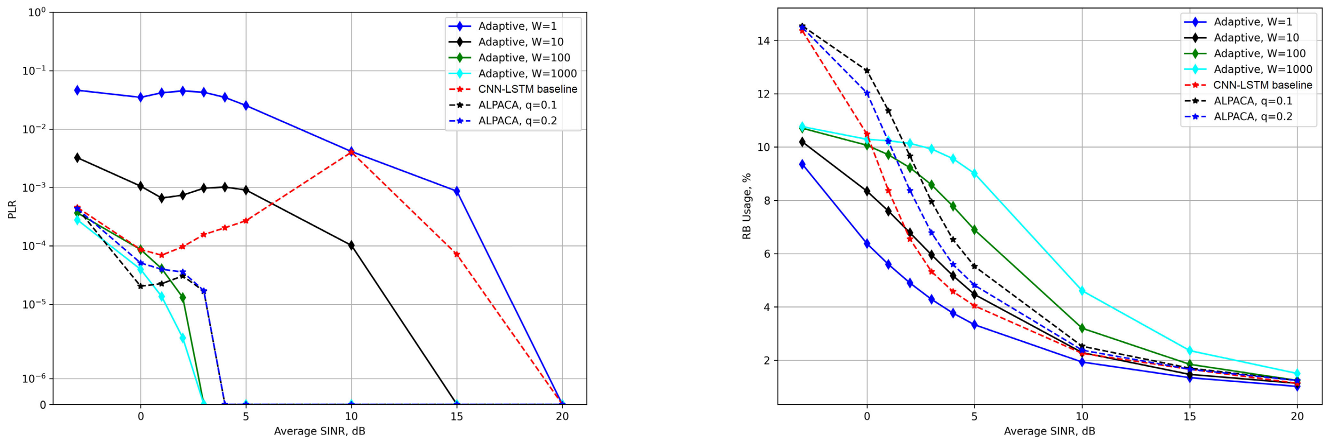
1) xEVA SCENARIO

The xEVA scenario has a fast-fading channel, making it a complicated target to predict channel quality. Both ALPACA and the reference Adaptive algorithm have the configuration parameters, i.e., the window W or quantile q, that determine how conservative the predictions are. In other words, they determine the tradeoff between the PLR and RB usage, see Fig. 6. Specifically, the increase in the window parameter W from 1 to 1000 makes predictions more conservative, increasing the RB usage and reducing the PLR down to the values of the Ideal prediction.

Having fixed $W = 100$, as a parameter of the Adaptive algorithm that provides PLR $\leq 10^{-5}$ starting from the average SINR of 5dB, we can conclude that ALPACA significantly improves the network performance in terms of coverage and RB usage.

Specifically, as shown in Fig. 6(a), ALPACA with $q = 0.2$ satisfies the PLR requirement for the average SINR above 0 dB, while the Adaptive algorithm with $W = 100$ requires the SINR higher than 5 dB. Thus, ALPACA provides a larger coverage. Also, Fig. 6(b) confirms that ALPACA provides a significant channel resource economy while satisfying the QoS requirements for edge users. The gain in the RB usage exceeds 40% compared with the Adaptive algorithm with $W = 100$.

Note that ALPACA with $q = 0.5$ (that does not have asymmetry in the loss function and is denoted as CNN-LSTM in this section) provides too optimistic predictions. Although it reduces channel resource consumption, it does not satisfy the PLR requirement by orders of magnitude, which highlights the necessity of accounting for unequal risks. Additionally, it sheds light on the learned patterns of the neural network layers: we can see that in both of the key metrics (PLR and RB usage), the CNN-LSTM are located near the results of the Adaptive algorithm with $W = 1$ and $W = 10$, meaning that during training with a symmetric loss, the network has been trained to look only at recent events. Contrary to that, with the asymmetric loss, ALPACA is capable of looking over its entire receptive field of 128 previous CQIs to find the relevant channel patterns. Similarly to the Adaptive algorithm with $W = 100$ and $W = 1000$, it considers rare risky events but does it more accurately.



(a) PLR in the EVA scenario

(b) RB usage in the EVA scenario

FIGURE 7. PLR (a) and RB usage (b) in the EVA scenario.

2) EVA SCENARIO

The EVA scenario is designed to be similar to that of the xEVA channel measurements in terms of channel parameters such as user speed and environment parameters using a 3GPP channel model.

Figures 7(a) and 7(b) show that ALPACA with $q = 0.2$ provides sufficient coverage and minimizes the resource consumption, respectively, reducing it up to 30% as compared to the Adaptive algorithm with $W = 100$.

Small performance degradation of the ALPACA algorithm in the high-SINR area can be explained by the fact that each CQI corresponds to a certain SINR interval, and the measurements outside these intervals result in CQI value saturation, reducing the amount of information the neural network can extract from it. However, the absolute value of the increase in RB usage compared to the counterpart algorithms is negligible. Thus, potentially more accurate CQI feedback may be considered as a possible way to further increase the accuracy of the channel prediction algorithms.

3) NETWORK CAPACITY IN URLLC SCENARIOS

For both of these scenarios, one of the metrics chosen was the RB usage by the user. This metric, while simple to compute and easy to compare (in case of equal average SINR), is not as interpretable, as the upper bound of the Network Capacity, which is the inverse of RB usage (per 1 user). This determines the number of users that can be supported while providing necessary QoS.

We note that both in our experiments and in [44] the average SINR for the user is distributed in the 5-15 dB region, and take this interval as the bound for estimating this KPI for both the xEVA and EVA scenarios.

We select the Adaptive algorithm with $W = 100$ as the baseline algorithm with the lowest resource usage while

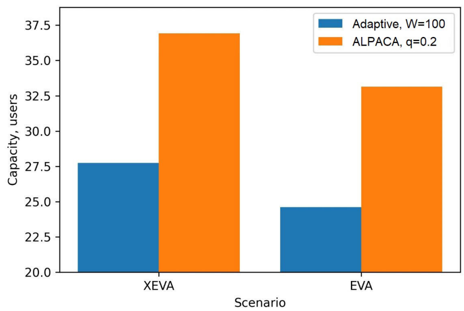


FIGURE 8. Cell capacity plot for different URLLC scenarios.

providing the necessary QoS within the whole interval, and select the ALPACA with a quantile value of 0.2 for the same reasons out of the proposed algorithms.

Figure 8 shows the bar plot comparing the network capacity achieved by the two algorithms in the xEVA and EVA scenarios. This figure demonstrates, that the proposed algorithm can significantly increase the capacity of the cell (up to 40% compared with the Adaptive algorithm with $W = 100$) in this SINR region both in the experimental and simulated scenario.

VI. CONCLUSION

This paper proposes a novel algorithm for channel prediction in URLLC named ALPACA: Asymmetric Loss Prediction Algorithm for Channel Adaptation.

ALPACA uses a convolutional-recurrent neural network to predict the channel quality in the future and adjust the MCS accordingly. This type of neural network is selected for its ability to work with sequential data and extract both local time-frequency patterns and more global time-series features. The key feature of ALPACA is the focus on the reliability requirements for URLLC. To satisfy these requirements, ALPACA uses a specifically designed asymmetric loss based on quantile regression.

We have evaluated the ability of the designed algorithm to provide the necessary QoS with a low amount of the used channel resources in typical URLLC deployment scenarios with the ns-3 simulation framework. However, in contrast to traditional simulation studies, we used both real (experimental) channel traces and simulated ones. For each scenario, we studied the performance of ALPACA in different channel quality regions, as well as measured the performance of the baseline algorithms and the upper bound.

The numerical results show that the developed algorithm satisfies the strict requirements of URLLC in tested scenarios, as well as provides an up to 30%…40% reduction of channel resource consumption compared with both heuristic and neural-network-based state-of-the-art approaches. Also, it increases the network capacity for URLLC users by about 40% in the relevant SINR region.

An important property of ALPACA is its stable gains and its applicability to various scenarios. Specifically, ALPACA is tested with both modeled channels and the channels collected in real measurements in a city center with different surrounding conditions, including buildings, parks, and the number of cars on the road. Similar gains observed in these experiments show the capabilities of ALPACA to be generalized to different scattering environments, as well as varying velocities.

Finally, ALPACA is lightweight and can be deployed in real-time on real hardware.

VII. FUTURE WORK

In future work, we are going to study the use of neural channel adaptation in Massive MIMO URLLC systems, which is considered to be a part of 6G systems [45]. The Massive MIMO-URLLC could benefit from the flexibility and efficiency of neural networks in handling complex and dynamic wireless channel conditions. Such neural network-based approaches can encompass not only the CQI but also some information about the MIMO channel matrix to extract more information about the channel state. Additionally, CSI MIMO data itself can be added to the dataset.

Another direction would be to examine possible neural network design improvements, such as incorporating various attention mechanisms or using different types of transmission feedback to further improve the MCS selection process.

Finally, a promising research direction is the adaptation of channel prediction algorithms to a rapidly changing environment, with both the delay profile and Doppler frequency varying during user movement. While currently, separate models were trained for the xEVA and EVA scenarios, it would be interesting to investigate neural network transfer and adaptation techniques that can effectively handle the quickly evolving channel environment while being simultaneously sufficiently robust for URLLC.

REFERENCES

[1] A. Belogaev, E. Khorov, A. Krasilov, D. Shmelkin, and S. Tang, "Conservative link adaptation for ultra reliable low latency communications," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, Jun. 2019, pp. 1–5, doi: 10.1109/BLACKSEACOM.2019.8812824.

[2] K. Glinskiy, A. Kureev, A. Krasilov, and E. Khorov, "PABAFT: Channel prediction approach based on autoregression and flexible TDD for 5G systems," *Electronics*, vol. 11, no. 12, p. 1853, Jun. 2022, doi: 10.3390/electronics11121853.

[3] I. C. Wong and B. L. Evans, "Sinusoidal modeling and adaptive channel prediction in mobile OFDM systems," *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1601–1615, Apr. 2008, doi: 10.1109/tsp.2007.909002.

[4] M. E. Morocho Cayamcela and W. Lim, "Artificial intelligence in 5G technology: A survey," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2018, pp. 860–865, doi: 10.1109/ICTC.2018.8539642.

[5] J. Joo, M. C. Park, D. S. Han, and V. Pejovic, "Deep learning-based channel prediction in realistic vehicular communications," *IEEE Access*, vol. 7, pp. 27846–27858, 2019, doi: 10.1109/ACCESS.2019.2901710.

[6] Y. Yang, F. Gao, X. Ma, and S. Zhang, "Deep learning-based channel estimation for doubly selective fading channels," *IEEE Access*, vol. 7, pp. 36579–36589, 2019, doi: 10.1109/ACCESS.2019.2901066.

[7] H. He, S. Jin, C.-K. Wen, F. Gao, G. Y. Li, and Z. Xu, "Model-driven deep learning for physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 77–83, Oct. 2019, doi: 10.1109/MWC.2019.1800447.

[8] X. Ye, Y. Yu, and L. Fu, "Deep reinforcement learning based link adaptation technique for LTE/NR systems," *IEEE Trans. Veh. Technol.*, vol. 76, no. 2, pp. 7364–7379, Jun. 2023, doi: 10.1109/TVT.2023.3236791.

[9] C. Eom and C. Lee, "Hybrid neural network-based fading channel prediction for link adaptation," *IEEE Access*, vol. 9, pp. 117257–117266, 2021, doi: 10.1109/ACCESS.2021.3106739.

[10] S. Avallone, N. Pasquino, S. Zinno, and D. Casillo, "Smartphone-based measurements of LTE network performance," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2017, pp. 1–6, doi: 10.1109/I2MTC.2017.7969854.

[11] M. Lerch, "Experimental comparison of fast-fading channel interpolation methods for the LTE uplink," in *Proc. 57th Int. Symp. ELMAR (ELMAR)*, Sep. 2015, pp. 5–8, doi: 10.1109/ELMAR.2015.7334482.

[12] N. Rupasinghe and I. Güvenç, "Capturing, recording, and analyzing LTE signals using USRPs and LabVIEW," in *Proc. SoutheastCon*, Apr. 2015, pp. 1–7, doi: 10.1109/SECON.2015.7132939.

[13] G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator*. Berlin, Germany: Springer, 2010, pp. 15–34.

[14] *Study on Channel Model for Frequencies from 0.5 to 100 GHz*, Technical Report (TR) document 38.901, Version 16.1.0, 3rd Generation Partnership Project (3GPP), Jan. 2020. [Online]. Available: http://www.3gpp.org/DynaReport/38901.htm

[15] H. Yin, H. Wang, Y. Liu, and D. Gesbert, "Addressing the curse of mobility in massive MIMO with Prony-based angular-delay domain channel predictions," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2903–2917, Dec. 2020, doi: 10.1109/JSAC.2020.3005473.

[16] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5G wireless communications: A deep learning approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 227–236, Jan. 2020, doi: 10.1109/TNSE.2018.2848960.

[17] W. Jiang and H. D. Schotten, "Neural network-based fading channel prediction: A comprehensive overview," *IEEE Access*, vol. 7, pp. 118112–118124, 2019, doi: 10.1109/ACCESS.2019.2937588.

[18] Q. Zhou, W. Jiang, D. Wang, and H. D. Schotten, "Deep learning-based signal-to-noise ratio prediction for realistic wireless communication," in *Proc. IEEE 95th Veh. Technol. Conf. (VTC-Spring)*, Jun. 2022, pp. 1–5, doi: 10.1109/VTC2022-Spring54318.2022.9860403.

[19] A. K. Gizzini and M. Chafii, "A survey on deep learning based channel estimation in doubly dispersive environments," *IEEE Access*, vol. 10, pp. 70595–70619, 2022, doi: 10.1109/ACCESS.2022.3188111.

[20] C. Parera, A. E. C. Redondi, M. Cesana, Q. Liao, and I. Malanchini, "Transfer learning for channel quality prediction," in *Proc. IEEE Int. Symp. Meas. Netw. (M&N)*, Jul. 2019, pp. 1–6, doi: 10.1109/IWMN.2019.8805017.

[21] L. Xiong, Z. Zhang, and D. Yao, "A novel real-time channel prediction algorithm in high-speed scenario using convolutional neural network," *Wireless Netw.*, vol. 28, no. 2, pp. 621–634, Feb. 2022, doi: 10.1007/s11276-021-02849-y.

[22] V. Sharma, R. K. Arya, and S. Kumar, "Efficient channel prediction technique using AMC and deep learning algorithm for 5G (NR) mMTC devices," *IEEE Access*, vol. 10, pp. 45053–45060, 2022, doi: 10.1109/ACCESS.2022.3167442.

[23] G. Bartoli and D. Marabissi, "CQI prediction through recurrent neural network for UAV control information exchange under URLLC regime," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 5101–5110, May 2022, doi: 10.1109/TVT.2022.3152408.

[24] D. Madhubabu and A. Thakre, "Long-short term memory based channel prediction for SISO system," in *Proc. Int. Conf. Commun. Electron. Syst. (ICCES)*, Jul. 2019, pp. 1–5, doi: 10.1109/ICCES45898.2019.9002073.

[25] Y. Zhu, X. Dong, and T. Lu, "An adaptive and parameter-free recurrent neural structure for wireless channel prediction," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 8086–8096, Nov. 2019, doi: 10.1109/TCOMM.2019.2935714.

[26] G. Liu, Y. Xu, Z. He, Y. Rao, J. Xia, and L. Fan, "Deep learning-based channel prediction for edge computing networks toward intelligent connected vehicles," *IEEE Access*, vol. 7, pp. 114487–114495, 2019, doi: 10.1109/ACCESS.2019.2935463.

[27] T. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based CSI feedback approach for time-varying massive MIMO channels," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 416–419, Apr. 2019, doi: 10.1109/LWC.2018.2874264.

[28] W.-S. Son and D. S. Han, "Analysis on the channel prediction accuracy of deep learning-based approach," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Apr. 2021, pp. 140–143, doi: 10.1109/ICAIIC51459.2021.9415201.

[29] Z. Mai, Y. Chen, H. Zhao, L. Du, and C. Hao, "A UAV air-to-ground channel estimation algorithm based on deep learning," *Wireless Pers. Commun.*, vol. 124, no. 3, pp. 2247–2260, Jun. 2022, doi: 10.1007/s11277-021-09459-z.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[31] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[32] N. Simmons, S. B. F. Gomes, M. D. Yacoub, O. Simeone, S. L. Cotton, and D. E. Simmons, "AI-based channel prediction in D2D links: An empirical validation," *IEEE Access*, vol. 10, pp. 65459–65472, 2022, doi: 10.1109/ACCESS.2022.3182713.

[33] H. Jiang, M. Cui, D. W. K. Ng, and L. Dai, "Accurate channel prediction based on transformer: Making mobility negligible," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2717–2732, Sep. 2022, doi: 10.1109/JSAC.2022.3191334.

[34] S. Yang, S. Aghaeinezhadfirouzja, E. Chen, and M. Tao, "Channel measurement and performance of multicast beamforming in campus network," in *Proc. 8th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2016, pp. 1–6, doi: 10.1109/WCSP.2016.7752680.

[35] K. Glinskiy, A. Kureev, and E. Khorov, "SDR-based testbed for real-time CQI prediction for URLLC," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2021, pp. 1–2, doi: 10.1109/INFOCOMWKSHPS51825.2021.9484554.

[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.

[37] R. Koenker and G. Bassett, "Regression quantiles," *Econometrica*, vol. 46, no. 1, p. 33, Jan. 1978, doi: 10.2307/1913643.

[38] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[40] A. Ghosh and R. Ratasuk, *Essentials of LTE and LTE-A*. Cambridge, U.K.: Cambridge Univ. Press, 2011, doi: 10.1017/cbo9780511997082.

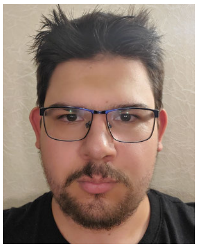
[41] S. N. Donthi and N. B. Mehta, "An accurate model for EESM and its application to analysis of CQI feedback schemes and scheduling in LTE," *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3436–3448, Oct. 2011.

[42] M. Q. Abdulhasan, M. I. Salman, C. K. Ng, N. K. Noordin, S. J. Hashim, and F. B. Hashim, "A channel quality indicator (CQI) prediction scheme using feed forward neural network (FF-NN) technique for MU-MIMO LTE system," in *Proc. IEEE 2nd Int. Symp. Telecommun. Technol. (ISTT)*, Nov. 2014, pp. 17–22, doi: 10.1109/ISTT.2014.7238169.

[43] *Evolved Universal Terrestrial Radio Access (E-UTRA)*, 3GPP document Version 10.7.0, Feb. 2013.

[44] F. Afroz, R. Subramanian, R. Heidary, K. Sandrasegaran, and S. Ahmed, "SINR, RSRP, RSSI and RSRQ measurements in long term evolution networks," *Int. J. Wireless Mobile Netw.*, vol. 7, no. 4, pp. 113–123, Aug. 2015, doi: 10.5121/ijwmn.2015.7409.

[45] H. Alves, G. Do Jo, J. Shin, C. Yeh, N. H. Mahmood, C. H. M. de Lima, C. Yoon, G. Park, N. Rahatheva, O.-S. Park, S. Kim, E. Kim, V. Niemel, H. W. Lee, A. Pouttu, H. K. Chung, and M. Latva-aho, "Beyond 5G URLLC evolution: New service modes and practical considerations," *ITU J. Future Evolving Technol.*, vol. 3, no. 3, pp. 545–554, Oct. 2022, doi: 10.52953/pfek2948.



KIRILL GLINSKIY received the M.S. degree (Hons.) in applied mathematics and physics from the Moscow Institute of Physics and Technology, in 2022. He is currently pursuing the Ph.D. degree, under the supervision of Evgeny Khorov. He is a Researcher with the Wireless Networks Laboratory, Institute for Information Transmission Problems of the Russian Academy of Sciences. His research interests include the application of machine learning methods in wireless networks and testbeds.

ALEKSEY KUREEV received the B.S. and M.S. degrees (Hons.) in applied mathematics and physics from the Moscow Institute of Physics and Technology, in 2015 and 2017, respectively, and the Ph.D. degree in telecommunications, under the supervision of Evgeny Khorov. He is currently a Senior Researcher with the Wireless Networks Laboratory, Institute for Information Transmission Problems of the Russian Academy of Sciences, and the Telecommunications Systems Laboratory, Higher School of Economics. He is the author of more than 20 research articles and patents. He supervised students and lectures on the fundamentals of telecommunications and SDR prototyping. His research interests include testbeds, software-defined radios, massive machine-to-machine communication, and ultra-dense networks.

EVGENY KHOROV (Senior Member, IEEE) received the Ph.D. and D.Sc. degrees, in 2012 and 2022, respectively. He is currently the Head of the Wireless Networks Laboratory, Institute for Information Transmission Problems of the Russian Academy of Sciences. He is also a Full Professor with MIPT. He has led dozens of academic and industrial projects. His main research interests include 5G/6G systems, next-generation Wi-Fi, the Wireless Internet of Things, and QoS-aware optimization. Being a Voting Member of IEEE 802.11, he contributed to the Wi-Fi 6 Standard. He has authored more than 180 articles, which received several best paper awards. Also, he was awarded national and international prizes in science and technology. He gives tutorials and participates in panels at large IEEE events. He chaired the TPC of various IEEE and IETF conferences and workshops. In 2020, he was awarded as the Editor of the Year of *Ad Hoc Networks*.