Performance Evaluation over DL-Based Channel Prediction Algorithm on Realistic CSI

Qiuheng Zhou†, Wei Jiang†, Donglin Wang*, Hans D. Schotten*†
†*German Research Center for Artificial Intelligence (DFKI GmbH), Kaiserslautern, Germany*
Emails: {qiuheng.zhou, wei.jiang, hans.schotten}@dfki.de
**Institute for Wireless Communication and Navigation, University of Kaiserslautern*
{dwang, schotten}@eit.uni-kl.de

***Abstract*—With the development of smart connected automated guided vehicles (AGVs) and robots, many new services and applications occur, which require flexible wireless end-to-end communication, high data transmission, and intensive computation. To achieve such a high demanding communication system, it is very important to predict the wireless channel parameters, which can help schedule the system resource management and optimize the system performance in advance, such as throughput and transmission efficiency. In this paper, we present our efforts towards proposing a deep learning-based channel prediction algorithm, which is then evaluated on the data set measured with different system state report frequencies from our implemented software-defined radio platform in different indoor environments. Results showed that the proposed channel predictor has a convincing ability on the real-world channel prediction.**

***Index Terms*—LTE, beyond 5G, MIMO, SNR, Channel Prediction, CNN, LSTM, srsLTE, Deep Learning**

I. INTRODUCTION

The approaching new wireless communication generation needs more efficient methods to deal with the tremendous growth of intelligent devices and applications, and the complexities of the system [1], [2]. Massive MIMO (mMIMO) has successfully emerged in the 5G communication systems [3]. Wireless transmission adaptation based on the received channel state information (CSI) is one of the key solutions, which enables the communication system to achieve communication performance near the Shannon limit. Information on the received signal-to-noise ratio (SNR) is the basis of the whole span of transmission rate adaptation protocols [4]. The moving UEs can cause fast time-varying characteristics of the channel, which seriously lead to the imperfection of CSI. It has been well recognized that the imperfect CSI has negative impact on the performance of adaptive transmission systems, spanning from antenna selection to physical layer security [5].

Considering the future massive networks and the nonlinear varying channels, deep learning is a promising tool to provide competitive performance to existing approaches, with affordable and reasonable computational costs in complicated multiuser systems [6]. The observed outdated CSI is processed to forecast the future CSI. The channel predictor can achieve a very high prediction accuracy in a fast fading channel without any prior knowledge of the channel [6], [7]. The long short-term memory (LSTM) network is used to predict the SNR in V2X communication systems [8]. It can be concluded that LSTM is an efficient improved recurrent neural network (RNN), which has a better effect on solving the long-term dependence problems than general RNN. The authors in [9] have discussed the strengths and bottlenecks of some new approaches and improved the performance of channel prediction by increasing the number of input features on simulated channels. It needs to be pointed out, that data from the above approaches are mostly from simulated fading channel, which follows the Rayleigh distribution with an average power gain of 0 dB, where its channel gain h is zero-mean circularly-symmetric complex Gaussian random variable with the variance of 1, i.e., $h \sim \mathcal{CN}(0, 1)$.

The performance evaluation of new communication technology requires rigorous evaluation and practical verification before deployment. The author in [10] has taken the advantage of spectrum analyzer to estimate and extract the CSI in Vehicle-to-Vehicle driving scenario, which is however very space and time consuming, because of the huge amount of data streams. To look deep inside into the performance of channel predictor on real channel data, in this paper, we develop a deep learning-based channel prediction algorithm, which focuses on SNR prediction. The channel data for training and testing the algorithm is measured on a real over-the-air software-defined radio (SDR) communication platform. Performance evaluation is held in different indoor environments. Because the CSI shows different sensitivities to the environment, the frequency of channel data collection will have a considerable effect on the accuracy of channel prediction. The performance of the proposed channel predictor is also compared with baseline methods.

The paper is organized as follows. Section II introduces the overview of the communication system and the process of SNR measurement. Section III introduces the proposed model in detail, including the architecture of the learning framework and the prediction scheme. Further, we present the method of SNR prediction and the analysis of the prediction results in Section IV. Finally, conclusions are given in Section V.

This work of the authors was supported by the German Federal Ministry of Education and Research (BMBF) through the *KICK* project (Grant no. *16KIS1105*).

II. An overview of experiment platform and SNR Measurement

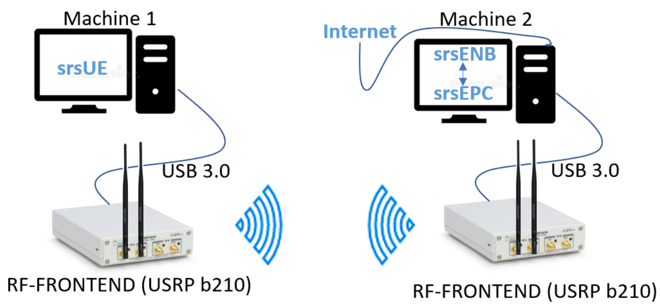

Fig. 1. The hardware setup of SDR-based LTE communication platform

The system setup can be seen in Fig. 1, where the srsENB and srsUE are installed on two separate computers respectively, which are full implementation of software-defined radio applications of srsRAN [11]. Both operate on Linux systems, which have low latency kernel and boosted CPU performance. The Universal Software Radio Peripheral (USRP) receives the signal from the srsENB through the USB 3.0 interface and broadcasts the signal to the environment by using VERT2450 antenna. Another USRP, which is connected to the srsUE can receive the signal from the transmitter through IP assignment and Identification which is supported by srsEPC.

In a factorial networking scenario, the moving AGVs need to predict the rapid changes of CSI in real-time, perform mobile edge computing (MEC), and work with base stations (BSs) for the adaptive transmission scheme to improve the transmission efficiency and throughput of wireless systems. In this paper, the SNR in the communication channel is selected as a prediction target. Because with the predicted SNRs, AGVs with communication end devices can then switch proper modulation modes to improve communication quality. The BSs can also adapt the transmission rate according to the predicted SNR.

For an optimal simulation of AGV communication in a complex indoor environment, we put our platform in different indoor environments. One is a roomy demo laboratory, where there are machines, pillars, tables, and walls with different materials in a relatively open indoor space, that can provide different types of reflections and create randomness as much as possible. Another one is in the office scenario, where there are narrow corridors and walls in a closed space. We assume that, because of the effect of small-scale fading, the channel might show different sensitivities in different environments while the end devices are moving. This means, that the frequency of the historical CSI observation should be adjusted with the changes of the environment. The channel prediction algorithms can then easily learn the collected CSI from a new environment in a short time and make hyper-parameter updates for better prediction. Because there are already plenty of indoor localization methods using sensors or computer vision with high accuracy, which can be applied to detect the location of the end device. Under this assumption, we have tested the channel prediction algorithm on the data set collected with different frequencies in two above environments.

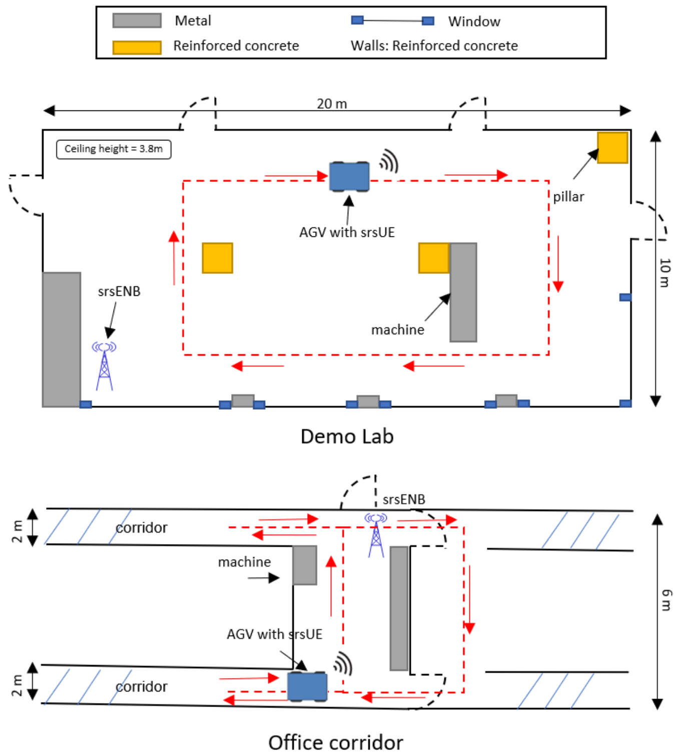

Fig. 2. Measurement site (top view)

Fig. 3. Measurement setup

As shown in Fig. 2, there are two top views of the measurement environment. We keep the srsENB device static in fixed positions. The srsUE device is then placed on an AGV,

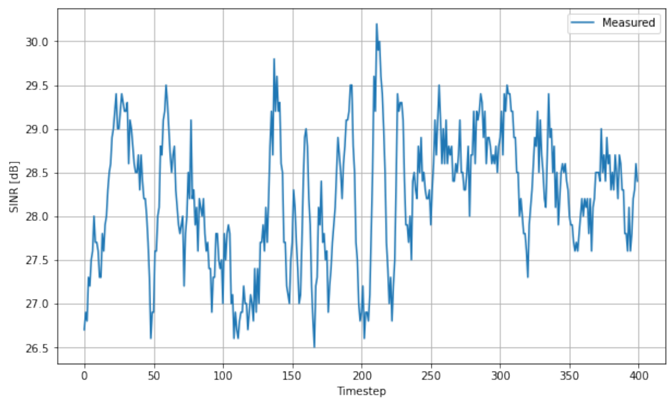

Fig. 4. An exemplified plot of SNR data with time step of every 1s in demo lab

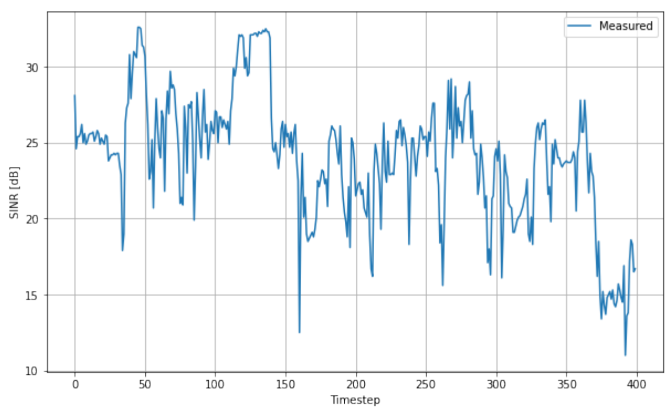

Fig. 5. An exemplified plot of SNR data with time step of every 1s in office corridor

as shown in Fig. 3, which runs along the red dotted routes with 0.5 m/s moving speed. So we have created a relatively dynamic communication scenario. In srsRAN, on both sides of UE and eNB, a tracer for CSI is implemented, we can easily monitor the channel state data as desired while the system is running.

As a system default setting, the srsRAN monitors the CSI data lists every second in Linux Terminal. For getting more or less information in the time series, we need to collect the CSI data more or less frequently. On the basis of 3GPP LTE standards, in the Frequency Division Duplexing (FDD) system, each radio frame is 10 ms long and consists of 10 subframes. We have changed the frequency of the uplink CSI lists report on the side of srsENB to every 10ms, so that we get the CSI value of each uplink frame. We have used the iperf tool for the uplink transmission test. We also set the time of the iperf-UDP test for a long enough time in order to collect more data, since the srsRAN system stops automatically if there is no data transmission between the srsENB and srsUE. As an example, Fig. 4 and Fig. 5 visualize small pieces of the collected uplink SNR data over the communication channel in two different scenarios as illustrated above, each of which consists of 400 consecutive SNR values. The step size is 1s. During this 400s measurement, we can find a relatively bigger fluctuation of the SNR value in the time series along with the movements of AGV in the office scenario, which has 22.1 dB difference value between the min-max values. The difference in the laboratory is only 3.7 dB instead. So from the observation of the measured data, it can be concluded that the sensitivity of the channel versus the environment in the office corridor is higher than in the demo laboratory.

III. Channel Predictor Architecture

In this section, we look deep into the CSI prediction in dynamic networks with moving devices. Our predictor is based on one-dimensional (1D) CNN and LSTM. Fig. 6 shows the architecture of this learning framework. Based on the knowledge of state-of-the-art results of the papers discussed in section II, a single LSTM layer is sufficient for simulated time series data forecasting [7]. we add a CNN layer before the LSTM other than a pure LSTM channel predictor, since CNN has excellent capability on feature extraction, it can help to process the raw data from real-world measurement.

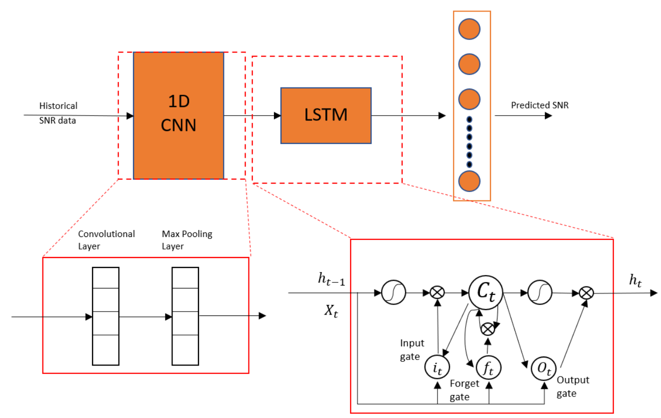

Fig. 6. The learning framework of the predictor

The 1D CNN network works for giving an architecture to learn smoothing parameters. It contains a 1D convolutional layer, a 1D max-pooling layer, and a flattening layer. The main difference between 1D and 2D is that these filters stride through the one-dimensional vectors along with one dimension, instead of two dimensions. Because they are part of the same function that outputs predictions, by optimizing the neural network loss, one optimizes smoothing parameters directly to perform well on a prediction task. The later layers then use the smoothed raw data and handle the main part of the time series forecasting problem.

The next component is an LSTM network for state vector prediction. LSTM has a stronger ability to overcome the vanishing gradient problem than the traditional RNN networks.

The mathematical description of the LSTM structure is as follows:

$$\begin{aligned}
\mathbf{i}_t &= \sigma_g(\mathbf{W}_i\mathbf{x}_t + \mathbf{R}_i\mathbf{h}_{t-1} + \mathbf{b}_i), \\
\mathbf{f}_t &= \sigma_g(\mathbf{W}_f\mathbf{x}_t + \mathbf{R}_f\mathbf{h}_{t-1} + \mathbf{b}_f), \\
\mathbf{g}_t &= \sigma_c(\mathbf{W}_g\mathbf{x}_t + \mathbf{R}_g\mathbf{h}_{t-1} + \mathbf{b}_g), \\
\mathbf{o}_t &= \sigma_g(\mathbf{W}_o\mathbf{x}_t + \mathbf{R}_o\mathbf{h}_{t-1} + \mathbf{b}_o),
\end{aligned} \tag{1}$$

where $\mathbf{i}$, $\mathbf{f}$, $\mathbf{o}$ are input gate, forget gate, output gate. Each component has the input weights $\mathbf{W}$, the recurrent weights $\mathbf{R}$, and the bias $\mathbf{b}$. σ_g denotes the state activation function which is the hyperbolic tangent function and σ_c is the gate activation function which is the sigmoid function. Therefore, the cell state $\mathbf{c}_t$ and the hidden state $\mathbf{h}_t$ at the time t can be calculated by

$$\begin{aligned}
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \\
\mathbf{h}_t &= \mathbf{o}_t \odot \sigma_c(\mathbf{c}_t)
\end{aligned} \tag{2}$$

where $\odot$ denotes the Hadamard product which takes two same-dimensional matrices and generates another matrix where each element i, j is the product of elements i, j of the original two matrices. The update of each LSTM unit can be briefly summarized as follows:

$$\mathbf{h}_t = LSTM(\mathbf{h}_{t-1}, \mathbf{x}_t, \theta) \tag{3}$$

where θ represents all the parameters in the LSTM network. In the rest of this paper, we employ this prediction model for channel prediction and evaluate its performance on different steps of prediction with a different number of historical data.

IV. Experimental Results of Channel Prediction

As described in section II, we perform SNR data collection by operating on the srsRAN for more than 1000s of each time of measurement. We have collected the SNR data from both every 1s and every 10ms in the two environments, so we have enough samples.

Table I lists the main parameters of the predictor, where a layer of 1D CNN with 32 filters each with 5 Kernels is set for raw data processing and one layer LSTM with a different number of units is used for time series prediction. Starting from an initial state with random values, the weights and biases are updated by Adam optimizer [12]. The learning rate is derived from callbacks of the learning rate scheduler during training. We use the mean absolute error (MAE) to calculate the error between the predicted value and the observed value, which is defined as

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{4}$$

where n is the total number of SNR samples used for testing, y_i denotes the predicted results at time step i, and x_i stands for its actual value.

We first examine the ability of the proposed predictor to predict the SNR data, which is measured in the laboratory with 10ms time step. In Fig. 7 we show the variation of SNR value obtained in the laboratory scenario. We use the first 12000 data as a training data set and the remaining 2000 samples as a test data set for prediction.

Fig. 8 shows the MAE of the channel prediction results built with adifferent number of hidden units in the LSTM layer. 10 historical SNR observations are used to predict different time steps in the future. The time steps values in the x-axis denote the times of 10ms in the future that the predictor is making a prediction for. Interestingly, the performance doesn't turn much better with the increased hidden units, which might be explained by enough data sets for learning. Therefore choosing a small number of hidden units is possible and should be used for the channel prediction to reduce the amount of computation

TABLE I
TRAINING PARAMETERS

Parameter	Value
Training environment	python 3.9.7 tensorflow-gpu 2.6.0
Learning rate	0.001
CNN filters & kernel	32 & 5
LSTM units	3/10/32/64/128/200
Optimizer	adam
Loss	mae

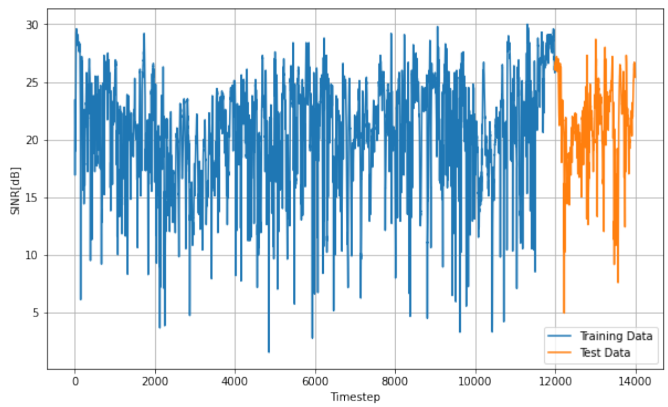

Fig. 7. SNR variation with 10ms time step in laboratory scenario. 12000 samples are used for training and 2000 samples are used for testing

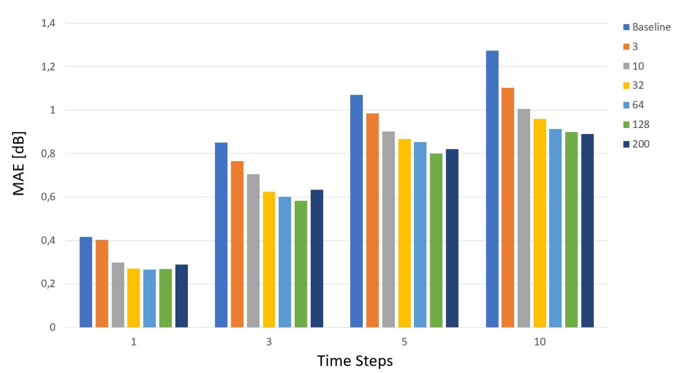

Fig. 8. MAE calculated between the channel prediction results and the actual value in lab scenario. Legend describes the number of hidden units for LSTM layer.

needed for network training. We compare the performance to the baseline determined by the last observed (non-predicted) channel state value. We can find a huge performance improvement by using our proposed method. The channel predictor has shown its effectiveness for predicting dynamic changing channels. To further verify the performance of the channel prediction method, we have tested the prediction algorithm with and without the CNN layer. As shown in Fig. 9, we have taken the case of using 10 SNR observations to predict a 1-time step ahead SNR. Parameters are still the same as given in Table I. From this figure, we can conclude, that 1D CNN helps to improve the prediction accuracy if the LSTM with a small number of hidden units can't achieve high accuracy. The predictor consists of CNN layer and LSTM with 32 units also outperforms the pure LSTM layer with 64 units.

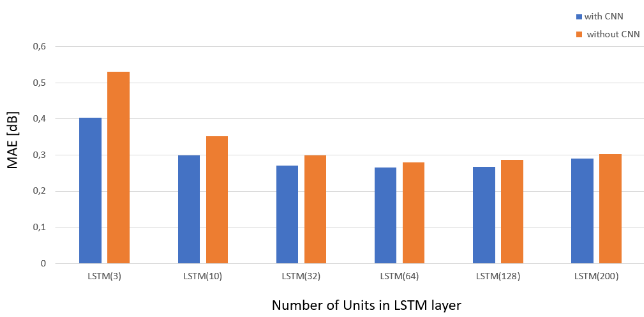

Fig. 9. Comparison of MAE results between algorithm with and without CNN layer

We now turn back to our assumption in this paper. We test our algorithm with the same process as above and take the fixed number of LSTM units. That is 1D CNN with 32 filters and LSTM with 64 hidden units. We now compare the prediction results on the data sets that are collected from both laboratory and office scenarios with a measurement frequency of 1s and 10ms respectively. For the prediction, we still use the 10 historical SNR data to predict 1-time step ahead SNR. The results are depicted in Fig. 10. It is obvious that the predictor can achieve high prediction accuracy in both the laboratory and office scenario on the data set with a 10ms time step. However, there is a tremendous performance degradation for the prediction case on the data set with 1s time step in the Office scenario. The performance on the data set with 1s time step in the laboratory has only a slight degradation, which has verified our assumption. In conclusion, in a relatively open indoor space, the system can under certain circumstances reduce the frequency of the report of channel data to release the load. However, in a closed indoor environment, channel data collection should be on frame level or even more frequently.

V. Conclusion

In this paper, we have developed an open-source based communication platform for collecting dynamic changing channel data. We can get any real communication channel state information as we want through an open-source RAN platform to provide data sets for strict evaluations. We have

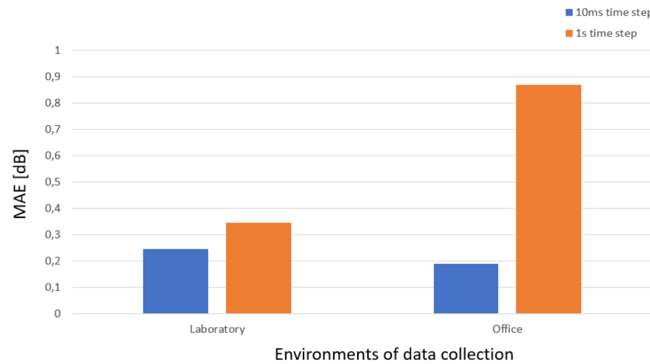

Fig. 10. MAE results of channel predictor on data set collected from two environments with different frequency of the measurement

then investigated the performance of the deep learning-based channel prediction algorithm. The channel predictor is evaluated and compared with baseline. We have also verified the channel prediction algorithm on data set collected from different indoor environment with different frequency of data collection. This predictor can be applied to a wide variety of wireless techniques include outdoor vehicular networks that need to improve performance on throughput, quality of service, etc.

Future work includes improving the performance of channel prediction by adding additional information to the neural networks. Performance evaluation on more complex channel structures, such as Multi-User mMIMO and 5G new radio.

References

[1] W. Jiang, B. Han, M. A. Habibi and H. D. Schotten, "The Road Towards 6G: A Comprehensive Survey," IEEE Open Journal of the Communications Society, vol. 2, pp. 334-366, 2021.

[2] C, Lipps, et al. "Towards the Sixth Generation (6G) Wireless Systems: Thoughts on Physical Layer Security." Mobile Communication-Technologies and Applications; 25th ITG-Symposium. VDE, 2021.

[3] A. Ghazanfari, et al. "Model-Based and Data-Driven Approaches for Downlink Massive MIMO Channel Estimation," in IEEE Transactions on Communications, vol. 70, no. 3, pp. 2085-2101, March 2022.

[4] S. He, et al. "State-Aware Rate Adaptation for UAVs by Incorporating On-Board Sensors," in IEEE Transactions on Vehicular Technology, vol. 69, no. 1, pp. 488-496, Jan. 2020.

[5] W. Jiang et al., "Neural network based wireless channel prediction," in Machine Learning for Future Wireless Communications: 303-325. 2020

[6] W. Jiang et al., "Neural network-based fading channel prediction: A comprehensive overview."[J]. IEEE Access, 2019, 7: 118112-118124.

[7] W. Jiang et al., "Deep learning for fading channel prediction." IEEE Open Journal of the Communications Society 1: 320-332. 2020

[8] G. Liu, et al., "Deep Learning-Based Channel Prediction for Edge Computing Networks Toward Intelligent Connected Vehicles," IEEE Access, vol. 7, pp. 114487–114495, 2019.

[9] M.K.Soszka, et al., "Fading Channel Prediction for 5G and 6G Mobile Communication Systems." International Journal of Electronics and Telecommunications 68.1 (2022): 153-160.

[10] J. Joo, et al., "Deep learning-based channel prediction in realistic vehicular communications". IEEE Access, 2019, 7: 27846-27858.

[11] G. Ismael, et al., "srsLTE: An open-source platform for LTE evolution and experimentation." Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization. 2016.

[12] D. P. Kingma , et al., "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980, 2004