# Deep Learning-Based Signal-to-Noise Ratio Prediction for Realistic Wireless Communication

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*—Artificial intelligence (AI) based channel state information (CSI) prediction for frequency division duplexing (FDD) massive multiple-input multiple-output (MIMO) systems have attracted growing attention recently. Accurate channel prediction can effectively improve the quality of CSI and can help optimize system transmission schemes, such as the throughput and transmission efficiency. The aim of this paper is to propose an efficient deep learning algorithm for signal-to-noise ratio (SNR) prediction in the real world, and a method for measuring SNR data from a universal software radio peripheral (USRP)-based software-defined radio platform. The results verify that the proposed channel measurement method is efficient for getting real-world channel data, and the deep learning-based algorithm has a strong ability on the real-world channel prediction.**

***Index Terms*—LTE, 5G, 6G, MIMO, SNR, Channel Prediction, LSTM, srsLTE, Deep Learning**

## I. Introduction

Research for the next-generation (6G) mobile networks has already begun with the examination and evaluation of candidate technologies and architectures [1], [2]. Massive MIMO (mMIMO) has successfully emerged in the 5G communication systems [3]. The campus network with connected moving automated guided vehicles (AGVs) and the vehicle-to-everything communication (V2X) are important technologies in the future of intelligent networking. Composed by moving user terminals and basic communication units and base stations (BSs), it is characterized by flexible networking, the coexistence of multiple communication modes, fast node movement, and predictable trajectory [4]. Adapting wireless transmission based on the received signal properties is one of the key paradigms enabling us to achieve communication performance of near the Shannon limit. Information on the received SNR for example is the basis of the whole span of transmission rate adaptation protocols. Meanwhile, the moving user terminal with high speed leads to the fast time-varying characteristics of the channel, which seriously makes the influence of the imperfect of CSI. It has been well recognized that the imperfect CSI has an overwhelming impact on the performance of a wide variety of adaptive transmission systems, spanning from antenna selection to physical layer security [5].

Recently, deep learning methods have been successfully applied in the field of wireless communications, and effective results have been obtained in channel prediction schemes. Among them, the outdated CSI is processed to forecast the future CSI, so that the channel predictor can achieve a very high prediction accuracy in a fast fading channel without any prior knowledge [6], [7]. Furthermore, the long short-term memory (LSTM) network is used to predict the SNR in V2X communication systems [8]. C. Luo [9] proposed an OCEAN model to predict CSI in a 5G wireless communication system. It can be concluded that LSTM is an efficient improved recurrent neural network (RNN), which has a good effect in solving the long-term dependence problems in general RNN. It needs to be pointed out, that data from the above approaches are 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., $h \sim \mathcal{CN}(0,1)$.

Performance assessment over wireless transmission technologies requires strict evaluation and real-world validation before deployment. While software for over-the-air simulation has evolved significantly over the years, it still cannot capture the complex real-world environment completely. Real-world evaluation over platforms with commercial LTE equipment, however, is restricted in individual configuration capabilities, mainly because of commercial considerations. This has resulted in the need for an open-source experimentation platform with a high degree of flexibility, that the researchers can commonly use for understanding the complexities associated with real-world settings while at the same time obtaining reproducible and verifiable results. An efficient way to perform SNR data measurement, prediction algorithm validation, and optimization are thus needed. In this paper, we develop a deep learning-based channel prediction algorithm, which focuses on the SNR prediction without the knowledge of channel estimates generated from the pilot signal. The channel data for training and testing the algorithms is measured and collected on a real over-the-air USRP-based LTE communication platform.

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

## II. An overview of System setup and SNR Measurement

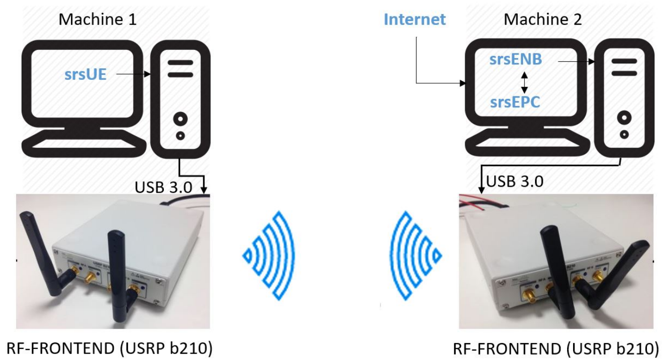


Fig. 1. The hardware setup of SDR-based LTE communication platform

The system setup can be seen in Fig. 1, where there are two computers for running srsENB with srsEPC, and srsUE respectively, which are full implementation of software-defined radio applications of srsRAN [10]. The srsUE is installed on an Intel NUC with Intel (R) 2.3 GHz i5-5300U. The srsENB and srsEPC are installed on a computer with Intel (R) 3.7GHz Xeon(R). Both Linux operation systems should have low latency kernel installed. The better the efficiency that CPU processes base band signals, the stabler the SDR platform will be. Both computers should have USB 3.0 ports for communication with USRP b210. Referred to the srsRAN architecture in [10], srsEPC is a lightweight implementation of a complete LTE core network (EPC). The srsEPC application runs as a single binary but provides the key EPC components of home subscriber service (HSS), mobility management entity (MME), service gateway (S-GW), and packet data network gateway (P-GW). The USRP receives the signal from the srsENB through the USB 3.0 interface and broadcasts the signal to the environment. Another USRP, which is connected to the computer installed with srsUE can receive the signal from the transmitter through IP assignment and Identification which is supported by srsEPC.

Followed by the srsRAN user manuals, the LTE communication platform can be established step by step, including the configuration and installation of eNB, USRP, EPC, and UE. Attentions are paid to avoid false settings from detail so that we can finally get stable LTE communication. A successful setup means, we can connect to another device from both sides of the computers through ping-test, and there is no latency, error, or sudden stop that occur while the system is running.

In a factory campus networking scenario, AGVs with different speeds communicate with the BSs using different bands. The AGVs need to predict the rapid changes of CSI in real-time, perform mobile edge computing (MEC), and work with BSs for the adaptive transmission scheme to improve the efficiency and throughput of wireless communication systems. Specifically, we consider the SNR in the communication channel as a prediction target. Then the AGVs can use the predicted SNR to 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 campus networks, we put our platform in a roomy demo laboratory, where there are also machines, pillars, tables, and walls with different materials, that can provide different types of reflections and create randomness as much as possible.

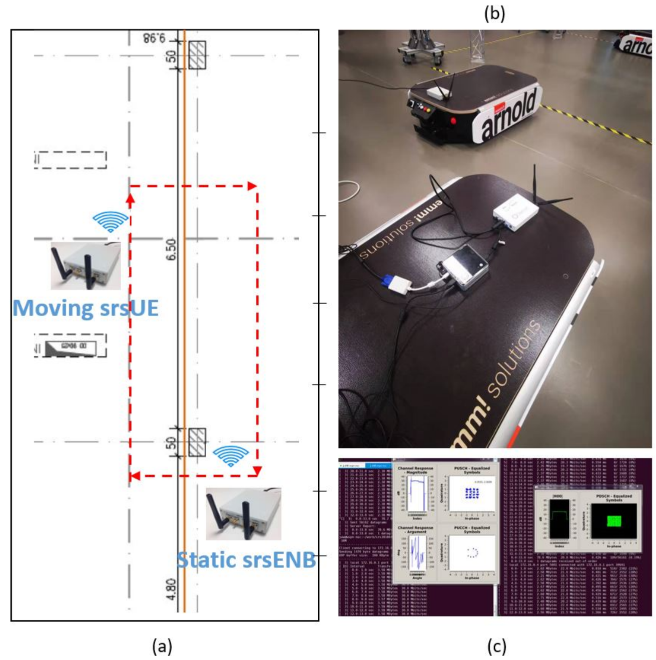


Fig. 2. Overview of SNR Measurement

As shown in Fig. 2, the Fig. 2.$(a)$ is a top view of the lab. There are two USRP devices, which connect to the srsUE and the srsENB machine respectively. We keep the srsENB device static in the corner of the lab. The srsUE device is then placed on an AGV, as shown in Fig. 2.$(b)$, which runs along the route as shown by the red dotted line in Fig. 2.$(a)$. So we have created a relatively dynamic communication scenario. In srsRAN, on both sides of UE and eNB, a tracer for channel state information is implemented, we can easily monitor the data like reference signal received power (RSRP), received signal strength indicator (RSSI), SNR, bit rate, etc, as the system is running. Fig. 2.$(c)$ shows the window of running status and plots of some of the channel states.

As default, the srsRAN monitors the CSI lists every second in Linux Terminal. For getting more information in the time series, we need to collect the CSI more frequently. On the basis of 3GPP LTE standards [11], in the FDD system, each radio frame is 10 ms long and consists of 10 subframes. Our srsRAN platform is running on the FDD mode with 5MHz bandwidth. 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. There are two ways to achieve this configuration. We can either set the `metrics_period_secs` in the configuration file or directly in the `main.cc` file. We have used the iperf tool for the uplink transmission test. Here we set the uplink transmission rate to be 5Mbit/sec so that the channel between the transceiver and receiver is fully loaded. We also set the time of the iperf test for 1000s in order to collect enough data, since the srsRAN system stops automatically if there is no data transmission between the srsENB and srsUE. As an example, Fig. 3 visualizes a small piece of the collected uplink SNR data over the communication channel, which consists of 1000 consecutive SNR value. We can find a relatively smooth changes of the SNR value in the time series along with the movements of AGV.

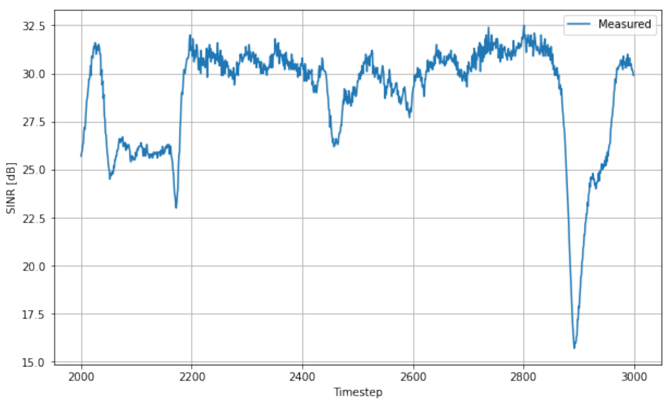


Fig. 3. An exemplified plot of SNR data with timestep of every 10ms

## III. Deep Learning-based channel predictor

Up-to-date knowledge of channel properties can greatly enhance wireless communication by allowing sophisticated rate adaptation and resource allocation. Often, the adaptation is performed at the transmitter based on the measured CSI piggybacked from the receiver. In a dynamic environment, such as those in vehicular communication, the CSI may already be outdated by the time it reaches the sender. Therefore, in this section, we look deep into the opportunities for predicting CSI in a dynamic network setting with moving devices.

Our predictor is based on CNN and LSTM, since CNN has excellent capability on feature extraction, and LSTM has remarkable ability on time sequences to extract the channel information hidden in the received channel data. Fig. 4 shows the architecture of this learning framework, which consists of a 1D CNN network, an LSTM network, and a dense layer for output. Based on the knowledge of state of the art results of the papers discussed in section II, we only take one layer for each of CNN, LSTM, and a dense output layer for simplicity and efficiency of the hole deep learning architecture, since a single LSTM layer is sufficient for simulated time series data forecasting [7].

The 1D CNN network works for giving an architecture to learn smoothing parameters. The first two layers of a convolutional neural network are generally a convolutional layer and a pooling layer: both perform smoothing. 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.

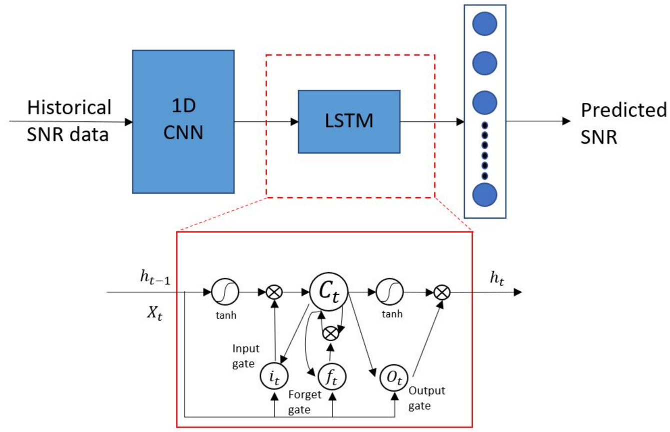


Fig. 4. The learning framework of the predictor

The LSTM network is used for state vector prediction. Compared to traditional RNN networks, LSTM is able to overcome the vanishing gradient problem. The LSTM network consists of several LSTM units, each of which has an input gate, a forget get, an output gate, and a memory cell. Input gate $\mathbf{i}$ controls the level of cell state update. Forget gate $\mathbf{f}$ controls the level of cell state reset. Output gate $\mathbf{o}$ controls the level of cell state added to hidden state. Each component has the input weights $\mathbf{W}$, the recurrent weights $\mathbf{R}$, and the bias $\mathbf{b}$, and calculated 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 $\sigma_g$ denotes the state activation function which is the hyperbolic tangent function and $\sigma_c$ 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 $t$ is given 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.

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 and analysis

As described in section II, we have taken SNR data collection through operating the srsRAN for 1000s. We collected the SNR data from every frame ($10ms$), so we have around

100000 samples of SNR data. We take 14000 consecutive samples from the dataset for training and testing the channel predictor. In Fig. 5 we show the variation of 14000 SNR samples obtained in a real-world moving AGV communication scenario. We use the 12000 samples of the data for training and the remaining 2000 samples for testing.

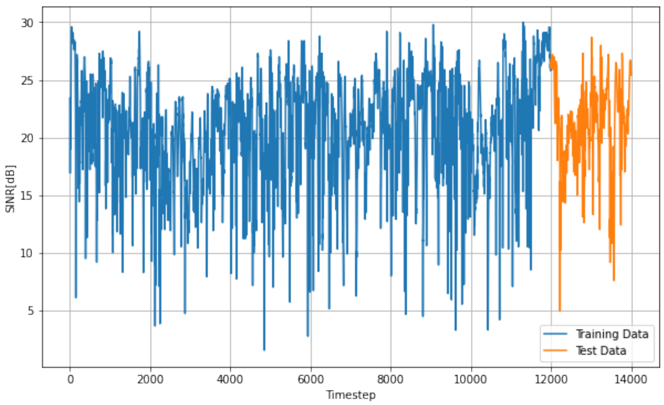


Fig. 5. SNR variation of frames in time series in the moving AGV communication scenario

The main parameters of the predictor are shown in Table I, where we have taken 32 filters each with 5 Kernels for CNN and 64 units for LSTM as an initial parameter configuration. Starting from an initial state with random values, the weights and biases are iteratively updated by Adam optimizer [12]. The learning rate of 0.001 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 true value, which is defined as

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \quad (3)$$

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

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 | 64 |
| Optimizer | adam |
| Loss | mae |

As a baseline, we have taken 10-time steps of SNR observations to predict 1-time step SNR ahead. The same parameters are taken as above with a maximum epoch of 50 and 200 steps of fitting per epoch, we can achieve an MAE of $0.2661dB$ on the 2000 samples of testing data. Fig. 6 shows that the predictor can converge within 10 epochs during training. Fig. 7 depicts the true and predicted SNR of 500 samples of test data. It can be seen from this figure that the predicted SNR is almost the same as the true value, which shows that the proposed model is very effective for predicting dynamic changing channels.

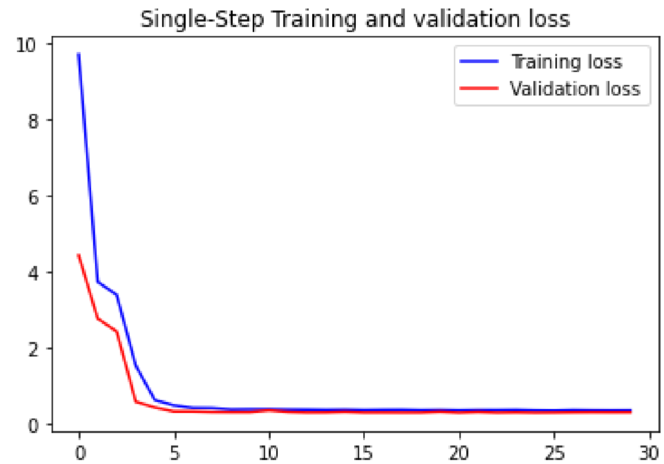


Fig. 6. The MAE loss curve of the prediction model for single time step prediction

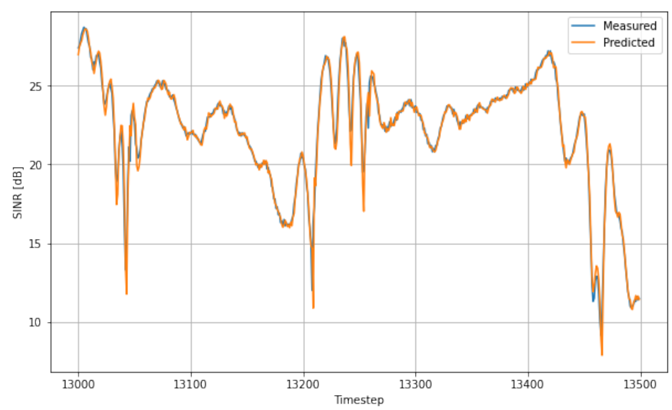


Fig. 7. Plot of SNR prediction results of 500 samples from test data

To further verify the performance of the channel prediction method, we have extended the prediction to multi-steps ahead prediction by using a different number of historical observations. Fig. 8 shows the results of MAE calculated between the predicted SNR and true SNR from different multi-steps predictions with different number of historical SNR values. Considering the algorithm parameters are set the same as shown in Table I, we can observe from this figure that the MAEs are around 0.28dB for the 1-time step ahead prediction, 0.6dB for the 3-time steps ahead prediction, 1dB for the 5-time steps ahead prediction, and 1.7dB for the 10-time steps ahead prediction. With the increasing number of utilized historical observations, the MAEs decrease, but only a bit. This figure has shown a great performance of our proposed model on the dataset collected on our proposed method, that a high accuracy with the resultant MAE of 1.7dB for 10 time steps (100ms) ahead prediction. We have also tried to change the hidden units and layers of the algorithms, as shown in Fig. 9, we have taken the 5-time steps prediction from 10-time steps observations as a example, where we change the number of units in LSTM layer. We can find a slight improvement of the accuracy by increasing the LSTM units. But those are actually no big improvements, even though two layers of LSTM is used. Therefore, the number of hidden units has a minimal effect on the prediction performance when the training data are large enough. As a result, choosing a small number of hidden units and historical observations is possible and should be used for channel prediction to reduce the computation complexity needed for neural network training. Besides, the predictor with 1-dimension CNN as an input layer can decrease the MAE about 0.02dB. If we compare the MAE of the predictor using one layer CNN with 32 filters and one layer LSTM with 32 units, and another predictor using two layers of LSTM, both with 32 units. The former gets the MAE of 1.0309dB, while the latter achieves the MAE of 1.0388dB, which has verified that the LSTM predictor with CNN layer for data prepossessing outperforms the pure LSTM predictor.

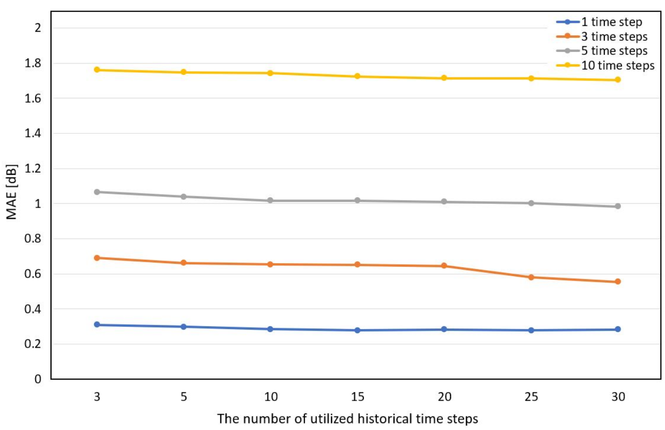


Fig. 8. MAE calculated between the SNR prediction results and the actual value from different multi-steps prediction with different number of utilized observations

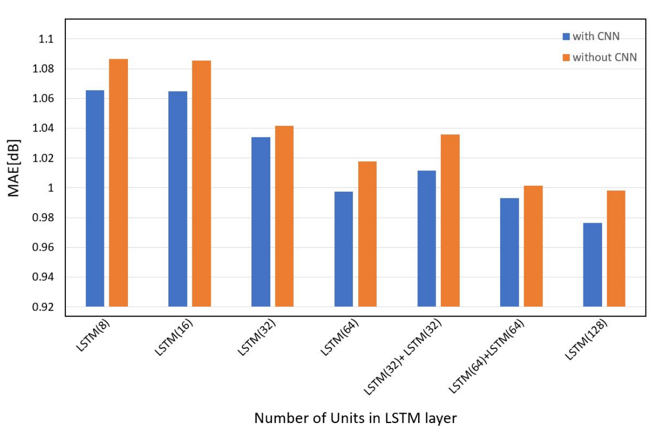


Fig. 9. MAE results of using different units in LSTM layer for the case of predicting 5 time steps SNR ahead by using 10 historical measurements

## V. Conclusion

In this paper, we have developed an USRP-based SDR communication platform for collecting dynamic changing SNR data. We have then investigated the performance of the deep learning-based channel prediction algorithm. We can get any real communication channel state information as we want through an open-source RAN platform to provide datasets for strict evaluations. The channel predictor is verified, that without any prior knowledge of the channel, it can achieve a high prediction accuracy. This predictor can be applied to a wide variety of wireless techniques 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, for example, location and information on the surrounding environment, which is collected by on-board sensors or cameras. Performance evaluation on more complex channel structures, such as mMIMO, 5G new radio will also be done accordingly.

## 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] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for Next Generation Wireless Systems," IEEE Communications Magazine, vol. 52, no. 2, pp. 186–195, Feb. 2014.
[4] F. Zeng, R. Zhang, X. Cheng, and L. Yang, "Channel Prediction Based Scheduling for Data Dissemination in VANETs," IEEE Communications Letters, vol. 21, no. 6, pp. 1409–1412, Jun. 2017.
[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., "Recurrent neural network-based frequency-domain channel prediction for wideband communications." in Proc. IEEE Vehicular Tech. Conf. (VTC). Kuala Lumpur, Malaysia May 2019.
[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] C. Luo, et al., "Channel state information prediction for 5g wireless communications: A deep learning approach," IEEE Trans. on Network Science and Engineering, vol. 7, no. 1, pp. 227–236, 2020.
[10] 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.
[11] 3GPP TS 36.211 version 12.3.0, "Evolved Universal Terrestrial Radio Access (E-UTRA);Physical channels and modulation," pp.10-12, October 2014
[12] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980