

Learning Without Forgetting: Predicting the Reliability of V2X Wireless Communication

Anja Dakić, Benjamin Rainer, Thomas Zemen

AIT Austrian Institute of Technology GmbH, Vienna, Austria
Email: anja.dakic@ait.ac.at

Abstract—Effective communication between vehicles and road users is essential for reducing accidents and congestion. Reliable wireless communication is crucial for decision-making in advanced driver assistance systems and autonomous vehicles. In this work, we propose a convolutional neural network to predict the frame error rate in vehicle-to-infrastructure scenarios. Using a geometry-based stochastic channel model and hardware-in-the-loop emulation, we generate a dataset on which our model achieves 90% validation accuracy. To adapt the model to new data, such as vehicle-to-vehicle scenarios, and to reduce computational costs for retraining the entire model from scratch, we explore methods like fine-tuning, transfer learning, and learning without forgetting (LwF). While these methods improve performance on new data, they reduce accuracy on the original data. To address this, we modify LwF by including some original data, achieving a balanced accuracy of 81.96%.

Index Terms—frame error rate, geometry-based stochastic channel model, convolutional neural network, learning without forgetting

I. INTRODUCTION

Vehicle-to-everything (V2X) communication enables autonomous vehicles (AVs) to perform reliably even in areas where other sensors like LIDAR, RADAR, or cameras may fail, primarily due to non-line-of-sight (NLOS) conditions. By exchanging environmental data and information about the position and velocity of vehicles, road safety can be significantly improved and traffic congestion reduced [1]. For the control and decision-making system of AVs to utilize this information, reliable communication must be guaranteed. Estimating and predicting the reliability of wireless communication channels, i.e., in terms of the frame error rate (FER), is key in many aspects. Predicting the FER is important for (a) determining how often important information has to be broadcast such that it is received with high probability within a certain urban area, (b) generating meaningful V2X test scenarios for verification and validation of advanced driver assistance systems (ADAS) relying on information received via V2X communication.

The FER in V2X scenarios, where the radio channels are time-varying and non-stationary [2], [3], [4], depends on different channel characteristics, transmitter (Tx) parameters, and receiver (Rx) architecture. Therefore, in [5] we propose a FER lookup table based on five condensed channel parameters, while in [6] we use the channel transfer function (CTF) of a stationarity region to predict the FER via a deep

neural network (DNN). The network is trained on vehicle-to-infrastructure (V2I) scenarios. The CTFs are obtained using a geometry-based stochastic channel model (GSCM), and the FER labels are measured via a hardware-in-the-loop (HiL) framework for a specific implementation of the IEEE 802.11p standard [7]. With this dataset generation setup, we are able to measure the FER of a stationarity region with an accuracy that does not depend on the length of the stationarity region. Despite the good results, we noticed that the trained network does perform poorly on other V2X communication scenarios, i.e., vehicle-to-vehicle (V2V), although one may assume that channel state information obtained in both scenarios may be similar, the scenarios differ significantly in terms of electromagnetic wave propagation conditions.

Our goal is to eventually deploy a pre-trained network into V2X hardware or on the engine control unit. Therefore, it is not feasible to retrain the neural network each time new data samples are available. In this paper, we investigate well-established adaptation methods such as transfer learning [8], fine-tuning [9], and learning without forgetting [10]. Transfer learning is already used for different problems in wireless communications [11]. In [12], transfer learning is used to predict the quality of channels in commercial 4G networks where limited data is available. Also, it is applied to channel state information based positioning in time-variant channels, where the knowledge from one environment is transferred to another [13], [14]. Furthermore, the authors in [15] use transfer learning for automatic modulation classification in time-variant channels and show how it reduces the computational complexity by training only some of the model parameters.

In summary, we aim to maintain a high prediction accuracy in already trained scenarios (V2I) while also adapting to new scenarios (V2V). First, we improve our DNN described in [6] by including convolution layers obtaining a higher accuracy in V2I scenarios. Second, we want to generalize the trained convolutional neural network (CNN) to V2V scenarios. However, the methods mentioned above do not work very well for our purpose. Initially, we train our neural network with a large dataset but for adapting to another scenario only a small number of samples are available. Therefore, in this paper, we propose a modified learning without forgetting method that outperforms existing ones and shows promising results in our scenarios.

Scientific Contributions of the Paper:

- We design a CNN for predicting FER classes significantly improving the accuracy in V2I scenarios, finally reaching an overall accuracy of 90% and strong improvements in the NLOS-LOS transition regions.
- We investigate well-established model adaptation methods to incorporate V2V scenarios while keeping computational and implementation complexity low. We present a variant of learning without forgetting that maintains accuracy in previously trained scenarios but allows adaptation to new scenarios.

II. NEURAL NETWORK AND BASELINE FER PREDICTION FOR V2I SCENARIOS

Despite the continuous nature of the FER as a measure of reliability of the wireless communication channel, we define the prediction of the FER as a classification task, where each class represents a FER region. We justify this by investigating the FER for different scenarios of the publicly available dataset described in [16]. The analysis reveals that FER regions provided in Table II cover the dynamics of the FER in most of the scenarios [6]. In this paper, we employ the V2I dataset introduced in [6]. It consists of channel information of each stationarity region for V2I scenarios which are generated from our GSCM and a stochastic channel model. The corresponding labels, i.e. the FER classes, are obtained by measurements performed with the HiL setup presented in [7], which consists of a wireless channel emulator [17] and two IEEE 802.11p compliant Cohda Wireless modems [18]. The parameters for the HiL setup are summarized in Table I.

For the task of predicting the FER, given the channel impulse response (CIR), we utilize a CNN with a dense part for classification. The convolutional layers represent the feature extraction and by cascading them one learns to extract features at different abstraction levels [19] from the input data.

The CIR is represented as a two-dimensional complex valued matrix $\mathbf{I} \in \mathbb{C}^{M \times P}$, where one dimension $M = 200$ corresponds to time samples, and the other, $P = 41$, to delay samples. We split the complex values of the CIR into real and imaginary parts, obtaining an input tensor $\mathbf{X}_{j,k,l}^{\text{CIR}}$ of rank three, with dimensions (2, 200, 41). We cascade five convolutional layers, while saving the output of each convolutional layer, as is shown in Fig 1. For our two-dimensional input (real and imaginary parts are treated separately), the convolution operation is given by

$$\mathbf{S}(i, r) = (\mathbf{I} * \mathbf{K})(i, r) := \sum_{m=1}^M \sum_{p=1}^P \mathbf{I}(m, p) \mathbf{K}(i - m, r - p), \quad (1)$$

where we call \mathbf{S} a feature map, \mathbf{I} denotes one channel of the input sample, and \mathbf{K} denotes the kernel. The kernel sizes of the convolutional layers have been chosen such that the neural network learns to extract delay-varying and time-varying information. After the last convolutional layer, we flatten all feature maps and forward them to the dense layer part where we learn to classify the CIR according to the

prescribed FER classes. The dense part of our networks has been taken from [6] where we have shown its capability to predict FER classes.

Parameter	Value
Transmit power, P_{Tx}	10 dBm
Modulation and coding rate	QPSK, 1/2
Number of frames, F	20000
Frame size	100 bytes
Frame rate	2200 frames/s
Stationarity time, T_{stat}	100 ms
Bandwidth, channel emulator	10 MHz
Time spacing, channel emulator	50 ns
Frequency spacing, channel emulator	156.25 kHz

TABLE I: Parameters used for measuring the FER via HiL framework.

The CNN is trained on 70% of the V2I dataset which consists in total of 15196 samples. The remaining 30% are used for model validation. For training the CNN we employ the Adam optimizer [20] with a learning rate of 10^{-4} . Training is performed on a batch level, utilizing a batch size of 10 for the training dataset and 8 for the validation dataset. The cross-entropy loss function is used to optimize model performance. We achieve a total accuracy of 90.14% on the validation dataset. We notice that adding the convolutional layers gained us an improvement of 4% compared to the dense model from [6]. Most notably, we achieve an improved accuracy in classes γ_2 and γ_3 . These classes cover the NLOS-LOS transition regions. The accuracy per class is shown in Table II.

Class	Accuracy	
	[6]	CNN
1: $\gamma_1 := (0, 5 \cdot 10^{-4}]$	84.73 %	90.61 % (+5.88 %)
2: $\gamma_2 := (5 \cdot 10^{-4}, 10^{-1}]$	78.42 %	83.93 % (+5.51 %)
3: $\gamma_3 := (10^{-1}, 5 \cdot 10^{-1}]$	77.38 %	86.10 % (+8.72 %)
4: $\gamma_4 := (5 \cdot 10^{-1}, 1]$	95.47 %	96.88 % (+1.41 %)

TABLE II: Accuracy per class of the validation dataset.

III. THE V2V SCENARIOS AND ITS DATASET

For the V2V scenarios, we use the publicly available dataset [16], which is available at <https://nextg.nist.gov/>. From this dataset we select two V2V measurements, V2V scenario 1 (sc1) and scenario 2 (sc2). V2V sc1 represents an urban environment on a small road located behind the AIT building in Vienna. The data for V2V sc2 has also been collected in the near vicinity of the AIT building, but on a four lane road. Fig. 2 shows the trajectories of the vehicles in both scenarios. In both cases, the vehicles approach each other, and while in V2V sc1 the maximum velocity is 30 km/h, in V2V sc2 the maximum velocity is 50 km/h. The wireless channel measurements are conducted at a carrier frequency of $f_C = 5.9$ GHz, with a bandwidth of $B = 150.25$ MHz

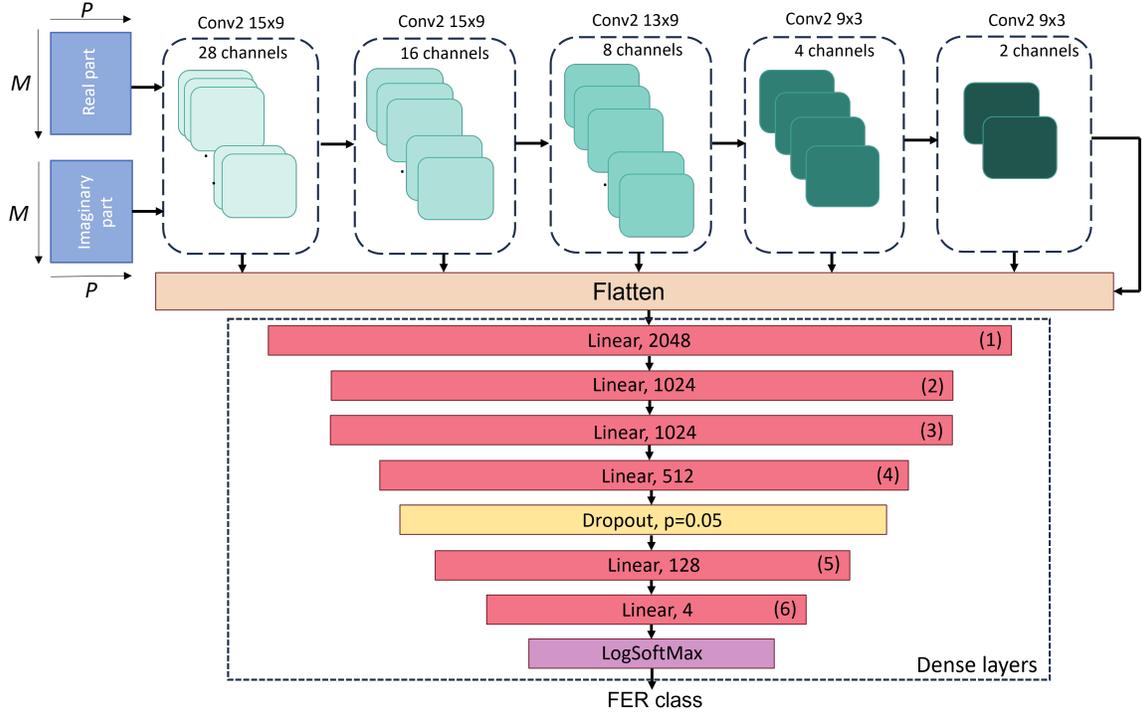


Fig. 1: Proposed CNN architecture.

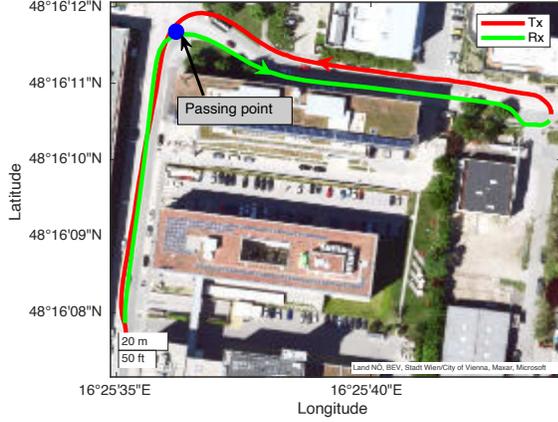
(subcarrier spacing $\Delta f = 250$ kHz), and a snapshot duration of $T_s = 500 \mu s$. These measurements are further used for GSCM modeling and CIR generation for these environments. The procedure of modeling the wireless channel with our GSCM, as well as the GSCM parameters for the V2V sc1 has been discussed in [21]. For V2V sc2, we slightly adapt the GSCM parameters, namely: the reference Rx power, G_0 for the LOS component and diffuse scatterers is -7 dB and -20 dB respectively, and the path loss exponent n_p for the LOS component is 1.8 and for diffuse scatterers we use 3. Like in [6] we obtain the FER for stationarity regions with a duration of 100 ms. The parameters for the IEEE 802.11p compliant V2X modems are the same as for the data acquisition of the V2I dataset.

We merge the data generated and labeled for the two scenarios, V2V sc1 and V2V sc2. The dataset contains the CIRs for each stationarity region and the corresponding FER obtained with the HiL framework [7]. Their overall duration is 50 and 60 s respectively, but for half of its duration the vehicles have no line of sight. This results in a high FER belonging to class γ_4 for half of the total number of stationarity regions. We balance the dataset by using only 30 s around the passing point for both scenarios. This leaves us with 600 new samples. We split this dataset into training, test and validation parts with a ratio of 60 – 20 – 20%. These datasets are also balanced to ensure that each class is represented by the same percentage across the training, validation, and test datasets for both scenarios.

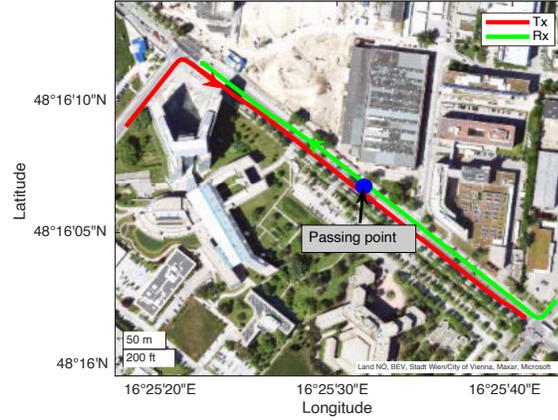
IV. MODEL ADAPTATION

The V2I and V2V scenarios differ significantly in terms of electromagnetic propagation conditions [22], despite the data for both scenarios being collected in the same urban area. Therefore, our goal is firstly to predict the FER also for inputs originating from a V2V communication scenario. Secondly, the accuracy in V2I scenarios shall remain high. Thirdly, for practical feasibility, we would like to only use training data from the new scenario and obtain good accuracy with a small number of training epochs. In the following, we study how fine-tuning, transfer learning, and learning without forgetting perform in our case. For this we consider the model in Fig. 1 trained for the V2I scenario denoted by $\mathcal{M}_{V2I} : X \rightarrow S$, mapping a sample $x \in X$ of our input space to $s \in S$, where S is the set of distributions over our FER classes. In our case the model is not altered as the task of classifying the FER w.r.t. the four classes (cf. Table II) remains the same. We further denote by \mathcal{M}_{V2V} the model trained using the V2V dataset only. We will express the dependence of the V2I model on the learnable/trainable parameters $\theta \in \mathbb{R}^N$ by $\mathcal{M}_{V2I}(\cdot; \theta)$. We apply the aforementioned learning methods as follows.

1) *Fine-tuning*: Here, we start with the pre-trained model \mathcal{M}_{V2I} and assume that the feature extraction (convolutional layers) remains the same for the new scenario. This is equivalent to stating the hypothesis that the learned kernels also extract meaningful features for the new scenario. Hence, we *freeze* the convolutional layers excluding them from the stochastic gradient descent updates. Since the dense part



(a) V2V sc1



(b) V2V sc2

Fig. 2: Scenarios and trajectories used for V2V scenarios.

of our network is responsible for learning the classification which we assume to be scenario dependent. We further freeze specific dense layers and denote the models by $\mathcal{M}_{2:4}$ and $\mathcal{M}_{4:5}$, respectively, where the subscript denotes which and how many dense layers have been frozen. The parameters of the remaining layers are initialized randomly.

2) *Transfer Learning*: Similar to fine-tuning, certain layers of \mathcal{M}_{V2I} are frozen and remain unchanged during subsequent training. We freeze all convolutional layers and we add new dense layers specific to the *new* task, effectively replacing all layers greater or equal than layer 5 (cf. Fig. 1). Here, we add the following fully connected dense layers with rectified linear unit (ReLU) activation functions in between: $(A_h, B_h) : (128, 64), (64, 32), (32, 16)$ and denote the model by \mathcal{M}_{TFL} . The remaining dense layers remain untouched and we start with the already trained parameters. We train the network using the V2V dataset only.

3) *Learning without Forgetting (LwF)*: Introduced in the seminal work of Li et al. [10], the concept is to take a trained network, add or extend specific layers fulfilling the requirements of the new task at hand and continue training with the new dataset only. The key idea of LwF is that

during training a combined loss of the initial model and the model under training is used. In this work, we adapt the original method presented in [10]. Let $X_{V2V} \subset X$ be our new training data with the corresponding labels $Y_{V2V} \in \mathbb{N}^{M \times 4}$, let $\theta_{V2I} \in \mathbb{R}^N$ denote the trained parameters using the V2I dataset. Further, let f_{loss} and g_{loss} be two loss functions, preferably with the same range. Algorithm 1 briefly out-

Algorithm 1 Modified Learning without forgetting

Input: $(X_{V2V}, Y_{V2V}), \mathcal{M}_{V2I}, \theta_{V2I}, \alpha, f_{\text{loss}}, g_{\text{loss}}, N_{\text{epochs}}$
Output: $\theta_{V2I, V2V}$

- 1: $\theta_1 = \theta_{V2I}; i = 1$
- 2: **for** $j = 1$ to N_{epochs} **do**
- 3: **for** each batch $B_{V2V} \subseteq X_{V2V}$ **do**
- 4: $\theta_{i+1} = \text{SGDStep}(\alpha f_{\text{loss}}(\mathcal{M}_{V2I}(B_{V2V}; \theta_i), Y_{V2V_{B_{V2V}}}) + (1 - \alpha)g_{\text{loss}}(\mathcal{M}_{V2I}(B_{V2V}; \theta_i), \mathcal{M}_{V2I}(B_{V2V}; \theta_1)))$
- 5: $i = i + 1$
- 6: **end for**
- 7: **end for**
- 8: **return** θ_i

lines the training procedure. The stochastic gradient descent update (SGDStep) is done using a convex combination of losses. The learning without forgetting parameter $\alpha \in [0, 1]$ allows to steer how strong the impact of the loss for the *new* scenario $f_{\text{loss}}(\mathcal{M}_{V2I}(B_{V2V}; \theta_i), Y_{V2V_{B_{V2V}}})$ is when updating the model parameters. Choosing a small α puts more emphasis on $g_{\text{loss}}(\mathcal{M}_{V2I}(B_{V2V}; \theta_i), \mathcal{M}_{V2I}(B_{V2V}; \theta_1))$, where $\mathcal{M}_{V2I}(B_{V2V}; \theta_1)$ denotes the prediction of the baseline given V2V input data. The rationale is, depending on α , either we learn faster from new data or we try to keep already learned weights but try to explore new weights improving the loss on the new dataset. In the first iteration the contribution of g_{loss} will be zero, thus allowing to step away from the learned weights θ_1 .

We further propose to adapt α using the gradient of the losses, ∇f_{loss} and ∇g_{loss} as follows

$$\alpha := \frac{1}{C\sqrt{n}} \left\| \sum_{i=1}^{|B_{V2V}|} \nabla_{\theta} f_{\text{loss}}(x_i) \right\|_{l_2}^{\sqrt{n}},$$

where

$$C = \left\| \sum_{i=1}^{|B_{V2V}|} \nabla_{\theta} f_{\text{loss}}(x_i) \right\|_{l_2} + \left\| \sum_{i=1}^{|B_{V2V}|} \nabla_{\theta} g_{\text{loss}}(x_i) \right\|_{l_2},$$

here n denotes the current epoch. In the case of training using batch gradients, we use the mean of gradients evaluated at the samples in B_{V2V} . The more epochs we train, the harder it becomes to achieve improvements in a new scenario. We call learning without forgetting using a convex combination of the losses as α -LwF and with adaptive α A-LwF. For both loss functions, we use the cross-entropy. However, the proposed LwF uses the knowledge distillation loss [23] for the *old* task. It represents the modified cross-entropy, which has the intention to reduce the gap between the probabilities

of correct and wrong classes. Hence, the probability of each class is calculated as:

$$q_j = \frac{\exp(z_j/T)}{\sum_j \exp(z_j/T)}, \quad (2)$$

where z_j represents the predicted vector and T is the so called temperature. For higher values of T the probability distribution over classes becomes softer. We denote this method by DL-LwF.

4) Learning without Forgetting with partial old data:

Similar to LwF, except that we introduce the loss on the old dataset (V2I) during training for the new data. We consider a subset of the old dataset $(X'_{V2I}, Y'_{V2I}) \subseteq (X_{V2I}, Y_{V2I})$. The subset is chosen such that it includes the same proportion of samples for each class relative to the number of samples available for each class. We modify g_{loss} loss in Algorithm 1 accounting for the samples from the old dataset which then reads $g_{\text{loss}}(\mathcal{M}_{V2I}(X'_{V2I}; \theta_i), Y'_{V2I})$. It calculates the loss of the prediction on old data when updating the model parameters.

A. Results

We evaluate the different training methods with respect to the accuracy in each scenario and the combined accuracy. For each of the models/training methods, we select a reasonable number of epochs. We summarize the obtained results in table III for each case, respectively. Training our model on the single datasets yields an acceptable accuracy in the V2I and V2V scenarios, respectively. But, the performance in the other scenario is in both cases not acceptable. We observe that the accuracy for V2V data under \mathcal{M}_{V2I} is higher than the accuracy of V2I data under \mathcal{M}_{V2V} . This discrepancy is due to the small amount of V2V data we used for training \mathcal{M}_{V2V} . The results using fine-tuning are also not convincing. Freezing the dense layers two to four $\mathcal{M}_{2:4}$ and training with the V2V dataset heavily tends to improve V2V accuracy but diminishes V2I accuracy. The same holds for freezing layers four and five of the dense layers $\mathcal{M}_{4:5}$. Starting with \mathcal{M}_{V2I} and applying transfer learning \mathcal{M}_{TFL} , we end up with worse performance than fine-tuning and training from scratch on the V2V dataset. For reference, we also train the proposed neural network with V2I and V2V together, denoted by $\mathcal{M}_{V2V\&V2I}$. We notice that merging the two datasets increases the V2I accuracy compared to \mathcal{M}_{V2I} . LwF, like introduced in [10] using the knowledge distillation loss ($\mathcal{M}_{\text{DL-LwF}}$), $\lambda_0 = 1$ and the temperature $T = 2$, achieves an accuracy of 82% in V2I and approx. 69% in V2V. In this case, LwF is able to achieve a high accuracy in the V2I test cases but does not perform well in V2V cases.

Our adapted learning without forgetting with fixed $\alpha = 0.005$ and training with V2V data and adaptive α outperforms the previous methods. We found the value for α by a grid search. We obtain an accuracy of approx. 78% for V2I data and approx. 80% for V2V data, respectively. For comparison, we add a portion of the V2I training samples to the V2V training samples. Already a small portion like 1/50 (222 samples) provide good results in both scenarios, respectively.

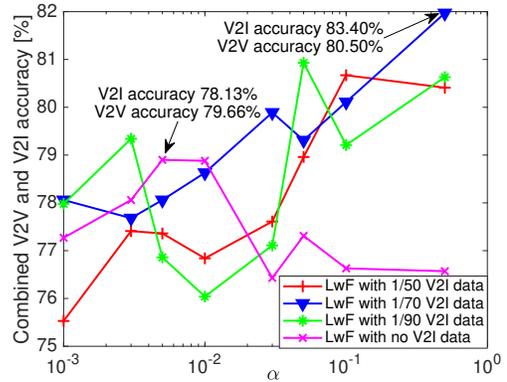


Fig. 3: Combined V2V and V2I accuracy for different α values.

Further results for different values of α and portions of V2I training samples are shown in Fig. 3. Fixing $\alpha = 0.5$ and taking 1/70 (152 samples) of V2I data yields an combined accuracy of 81.95%. Lastly, we evaluate the performance of our adaptive LwF considering old data and without old data. With a fraction of 1/50 of old data, we obtain slightly better results, namely 81.96%, than with fixed alpha. These two methods outperform all the other methods where no or only partial V2I training data is used. But, here we want to highlight that A-LwF does not introduce new parameters. Without partial V2I training, $\mathcal{M}_{\text{A-LwF}}$, we achieve a similar combined accuracy as for $\mathcal{M}_{\text{DL-LwF}}$. However, the accuracy in V2V and V2I are more balanced.

V. CONCLUSION

In this paper, we introduced a CNN that outperforms our previous work on predicting the FER in V2I scenarios, achieving an accuracy of 90% with improved accuracy in each FER class. In this paper, however, we were mainly interested in investigating methods for training an already trained network for new scenarios, i.e., V2V scenarios. These methods focus on low implementation and computational complexity. The goal was to obtain a high accuracy in the new scenarios with a small number of new data samples while maintaining high accuracy in the previously trained scenarios. We showed that well-established methods such as fine-tuning and transfer learning do not work well in our case. We further investigated LwF and proposed changes that reduce or eliminate the need of additional parameters. The accuracy for both old and new scenarios, as well as combined scenarios, were compared for different methods. Finally, we showed that we obtained the best trade-off with 81.96% combined accuracy for the A-LwF using 1/50 of the old training dataset. Without any data from previous scenarios we achieved an combined accuracy of 75.71%. In both cases only a small number of epochs, 20 to 30, was needed to achieve these results.

Training/Model	V2I accuracy [%]	V2V accuracy [%]	Combined accuracy [%]	Number of epochs
\mathcal{M}_{V2I}	90.14	56	73.07	100
\mathcal{M}_{V2V}	36.2	82.2	59.20	100
$\mathcal{M}_{V2V\&V2I}$	92.7	74.57	83.63	100
$\mathcal{M}_{2:4}$	36.62	74.57	55.59	15
$\mathcal{M}_{4:5}$	36.23	78.81	57.52	15
\mathcal{M}_{TFL}	33.46	71.18	52.32	15
\mathcal{M}_{DL-LwF}	82	69	75.49	20
$\mathcal{M}_{\alpha-LwF, 1/70 V2I}$	83.40	80.50	81.95	30
$\mathcal{M}_{A-LwF, 1/50 V2I}$	82.57	81.35	81.96	30
$\mathcal{M}_{\alpha-LwF}$	78.13	79.66	78.90	30
\mathcal{M}_{A-LwF}	76	75.42	75.71	20

TABLE III: Summary of Validation performance of the model adaptation methods on the V2V and V2I datasets, respectively.

VI. ACKNOWLEDGMENTS

This work is funded by the European Commission within the European Union’s Horizon 2020 research innovation programme funding ECSEL Joint Undertaking project AI4CSM under Grant Agreement No. 101007326 and within the Principal Scientist grant Dependable Wireless 6G Communication Systems (DEDICATE 6G) at the AIT Austrian Institute of Technology and from the ECSEL Joint Undertaking (JU) under grant agreement No. 101007350 (AIDOaRt project). The JU receives support from the European Union’s Horizon 2020 research and innovation program and Sweden, Austria, Czech Republic, Finland, France, Italy, Spain.

REFERENCES

- [1] U. D. of Transportation, “Saving lives with connectivity: A plan to accelerate V2X deployment,” Aug. 2024. [Online]. Available: https://www.its.dot.gov/research_areas/emerging_tech/pdf/Accelerate_V2X_Deployment_final.pdf
- [2] G. Matz, “On non-WSSUS wireless fading channels,” *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2465–2478, 2005.
- [3] L. Bernadó, T. Zemen, F. Tufvesson, A. F. Molisch, and C. F. Mecklenbräuker, “Delay and Doppler spreads of non-stationary vehicular channels for safety-relevant scenarios,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 82–93, Jan 2014.
- [4] L. Bernadó, T. Zemen, J. Karedal, A. Paier, A. Thiel, O. Klemp, N. Czink, F. Tufvesson, A. F. Molisch, and C. F. Mecklenbräuker, “Multi-dimensional K-factor analysis for V2V radio channels in open sub-urban street crossings,” in *21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2010.
- [5] A. Dakić, M. Hofer, B. Rainer, S. Zelenbaba, L. Bernadó, and T. Zemen, “Real-time vehicular wireless system-level simulation,” *IEEE Access*, vol. 9, pp. 23 202–23 217, 2021.
- [6] A. Dakić, B. Rainer, M. Hofer, and T. Zemen, “Frame error rate prediction for non-stationary wireless communication links,” in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Toronto, Canada, September 2023. [Online]. Available: <https://arxiv.org/abs/2304.05743>
- [7] A. Dakić, B. Rainer, M. Hofer, S. Zelenbaba, S. Teschl, G. Nan, P. Priller, X. Ye, and T. Zemen, “Hardware-in-the-loop framework for testing wireless V2X communication,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2023.
- [8] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, “A comprehensive survey on transfer learning,” *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.
- [9] K. P. Murphy, *Machine learning : a probabilistic perspective*. Cambridge, Mass. [u.a.]: MIT Press, 2013. [Online]. Available: https://www.amazon.com/Machine-Learning-Probabilistic-Perspective-Computation/dp/0262018020/ref=sr_1_2?ie=UTF8&qid=1336857747&sr=8-2
- [10] Z. Li and D. Hoiem, “Learning without forgetting,” in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 614–629.
- [11] C. T. Nguyen, N. Van Huynh, N. H. Chu, Y. M. Saputra, D. T. Hoang, D. N. Nguyen, Q.-V. Pham, D. Niyato, E. Dutkiewicz, and W.-J. Hwang, “Transfer learning for wireless networks: A comprehensive survey,” *Proceedings of the IEEE*, vol. 110, no. 8, pp. 1073–1115, 2022.
- [12] C. Parera, A. E. Redondi, M. Cesana, Q. Liao, and I. Malanchini, “Transfer learning for channel quality prediction,” in *2019 IEEE International Symposium on Measurements & Networking (M&N)*, 2019.
- [13] S. D. Bast, A. P. Guevara, and S. Pollin, “CSI-based positioning in massive mimo systems using convolutional neural networks,” in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020.
- [14] A. Foliadis, M. H. Castañeda, R. A. Stirling-Gallacher, and R. S. Thomä, “Transfer learning for CSI-based positioning with multi-environment meta-learning,” *arXiv preprint arXiv:2405.11816*, 2024.
- [15] P. Zhang and Z. Zhu, “Transfer learning for cnn based modulation classification in time-variant AWGN channels,” in *2022 IEEE 8th International Conference on Computer and Communications (ICCC)*, 2022.
- [16] B. Rainer, S. Zelenbaba, A. Dakić, M. Hofer, D. Löschenbrand, T. Zemen, X. Ye, G. Nan, S. Teschl, and P. Priller, “WiLi - vehicular wireless channel dataset enriched with LiDAR and Radar data,” in *GLOBECOM- IEEE Global Communications Conference*, 2022.
- [17] M. Hofer, Z. Xu, D. Vlastaras, B. Schrenk, D. Löschenbrand, F. Tufvesson, and T. Zemen, “Real-time geometry-based wireless channel emulation,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1631–1645, Feb 2019.
- [18] Cohda Wireless MK5 OBU specifications. [Online]. Available: <https://cohdawireless.com/solutions/hardware/mk5-obu/>
- [19] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.
- [20] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [21] A. Dakić, B. Rainer, P. Priller, G. Nan, A. Momić, X. Ye, and T. Zemen, “Wireless V2X communication testbed for connected, cooperative and automated mobility,” in *2024 IEEE Vehicular Networking Conference (VNC)*, 2024.
- [22] C. F. Mecklenbrauker, A. F. Molisch, J. Karedal, F. Tufvesson, A. Paier, L. Bernadó, T. Zemen, O. Klemp, and N. Czink, “Vehicular channel characterization and its implications for wireless system design and performance,” *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1189–1212, July 2011.
- [23] G. Hinton, O. Vinyals, and D. Jeff, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.