



HAL
open science

CNN aided Weighted Interpolation for Channel Estimation in Vehicular Communications

Abdul Karim Gizzini, Marwa Chafii, Ahmad Nimr, Raed M Shubair, Gerhard Fettweis

► **To cite this version:**

Abdul Karim Gizzini, Marwa Chafii, Ahmad Nimr, Raed M Shubair, Gerhard Fettweis. CNN aided Weighted Interpolation for Channel Estimation in Vehicular Communications. IEEE Transactions on Vehicular Technology, 2021, pp.1-1. 10.1109/TVT.2021.3120267 . hal-03381681

HAL Id: hal-03381681

<https://hal.science/hal-03381681v1>

Submitted on 17 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CNN aided Weighted Interpolation for Channel Estimation in Vehicular Communications

Abdul Karim Gizzini, Marwa Chafii, Ahmad Nimr, Raed M. Shubair, and Gerhard Fettweis

Abstract—IEEE 802.11p standard defines wireless technology protocols that enable vehicular transportation and manage traffic efficiency. A major challenge in the development of this technology is ensuring communication reliability in highly dynamic vehicular environments, where the wireless communication channels are doubly selective, thus making channel estimation and tracking a relevant problem to investigate. In this paper, a novel deep learning (DL)-based weighted interpolation estimator is proposed to accurately estimate vehicular channels especially in high mobility scenarios. The proposed estimator is based on modifying the pilot allocation of the IEEE 802.11p standard so that more transmission data rates are achieved. Extensive numerical experiments demonstrate that the developed estimator significantly outperforms the recently proposed DL-based frame-by-frame estimators in different vehicular scenarios, while substantially reducing the overall computational complexity.

Index Terms—Channel estimation, deep learning, IEEE 802.11p standard, vehicular communications.

I. INTRODUCTION

In practical vehicular applications, the communication reliability between vehicles is very important. In general, communication reliability can be evaluated by the accurate data recovery of the transmitted data at the receiver which depends primarily on the channel estimation accuracy. A precisely-estimated channel is critical for the equalization, demodulation, and decoding operations to follow. Hence, an accurate and robust channel estimation is essential for improving the overall system performance. Accurate channel estimation in vehicular communication is a challenging task, since in vehicular environments wireless channels have a high Doppler shift and a large delay spread [1], [2]. As a result, the channel shows time and frequency selective fading characteristics due to motion and multi-path components. This double selectivity arises in high mobility vehicular environment where the channel rapidly varies within the transmitted frame.

IEEE 802.11p frame structure [3] allocates two full preamble symbols that are used for basic least square (LS) estimation, and four pilot subcarriers within the transmitted data symbols which are used for channel variation tracking over time. This basic estimation at the beginning of the frame is simple,

but it becomes invalid for the equalization of the successive transmitted symbols as a result of high channel variation in vehicular environments. To improve the basic LS estimation performance, there exist two channel estimation categories: (i) symbol-by-symbol (SBS) estimators, where the channel is estimated for each received symbol separately [4]–[6]. (ii) frame-by-frame (FBF) estimators, where the previous, current and future pilots are employed in the channel estimation for each received symbol [7]. The well known FBF estimator is the conventional 2D linear minimum mean-squared error (LMMSE) where the channel and noise statistics are utilized in the estimation, and thus, leading to comparable performance to the ideal case. However, the 2D LMMSE suffers from high computational complexity making it impractical in real case scenarios. Therefore, there is a need for robust and low complexity FBF estimators.

Recently, deep learning (DL) algorithms have been integrated into wireless communications physical layer applications [8], [9] including channel estimation [10]–[12], due to its great success in improving the overall system performance, especially when used on top of low-complexity conventional estimators. DL algorithms are characterized by robustness, low-complexity, and good generalization ability making the integration of DL into communication systems beneficial. Motivated by these advantages, DL algorithms have been integrated in IEEE 802.11p channel estimators in two different manners: (i) Feed-forward deep neural networks with different architectures and configurations are employed on top of SBS estimators [13]–[15], where the channel is estimated for each received data symbol directly. (ii) Convolutional neural networks (CNNs) are integrated in the FBF estimators as a good alternative to the 2D LMMSE estimator, where the estimated channel for the whole frame is considered as a 2D low resolution noisy image and CNN-based processing is applied as super resolution and denoising techniques. Unlike the 2D LMMSE estimator, the CNN processing achieves considerable performance gain while preserving low computational complexity. The higher performance accuracy can be achieved by utilizing FBF estimators, since the channel estimation of each symbol takes advantage from the knowledge of previous, current, and future allocated pilots within the frame. Unlike, SBS estimators, where only the previous and current pilots are exploited in the channel estimation for each received symbol. In this paper, we will focus on FBF estimators, especially, the recently proposed CNN-based ones.

In [16], the authors propose CNN-based channel estimator denoted as channel network (ChannelNet), that applies radial basis function (RBF) interpolation as an initial channel estimation, after that the RBF estimated channel is considered as a

Authors acknowledge the CY INEX for the support of the project through the ASIA Chair of Excellence Grant (PIA/ANR-16-IDEX-0008).

Abdul Karim Gizzini is with ETIS, UMR8051, CY Cergy Paris Université, ENSEA, CNRS, France (e-mail: abdulkarim.gizzini@ensea.fr).

Marwa Chafii and Raed M. Shubair are with the Department of Electrical and Computer Engineering, New York University (NYU), Abu Dhabi 129188, UAE (e-mail: {marwa.chafii, raed.shubair}@nyu.edu).

Ahmad Nimr and Gerhard Fettweis are with the Vodafone Chair Mobile Communication Systems, Technische Universität Dresden, Germany (e-mail: {ahmad.nimr, gerhard.fettweis}@tu-dresden.de).

low resolution image, where super resolution CNN (SR-CNN) followed by denoising CNN (DN-CNN) are integrated on top of the RBF estimated channel. ChannelNet estimator suffers mainly from high computational complexity and considerable performance degradation in high mobility vehicular scenarios. Temporal spectral ChannelNet (TS-ChannelNet) has been proposed in [17], where average decision-directed with time truncation (ADD-TT) interpolation that is based on the demodulation and averaging of each received symbol. After that, super resolution convolutional long short-term memory (SR-ConvLSTM) is used to improve ADD-TT interpolation accuracy. TS-ChannelNet estimator outperforms the ChannelNet estimator especially in low signal-to-noise ratio (SNR) regions. Moreover, the TS-ChannelNet estimator has lower computational complexity than the ChannelNet estimator since only one CNN network is considered instead of two CNNs as proposed in the ChannelNet estimator. Nevertheless, TS-ChannelNet also suffers from a considerable performance degradation in high mobility vehicular scenarios. It is worth mentioning that both ChannelNet and TS-ChannelNet estimators suffer from high latency at the receiver, since the receiver should wait for the arrival of the whole frame in order to start the channel estimation process.

In order to overcome the high complexity and performance degradation in high vehicular scenarios, we propose in this paper novel hybrid and adaptive weighted interpolation (WI) channel estimators that utilize new pilot allocation schemes for the IEEE 802.11p that insert pilot symbols into the transmitted frame. Unlike ChannelNet and TS-ChannelNet estimators, where the applied interpolation techniques are performed in time and frequency, the developed WI estimators require only weighted time interpolation of the inserted pilot symbols. The number of inserted pilots is controlled and adapted according to the mobility condition. In particular, one orthogonal frequency-division multiplexing (OFDM) symbol with all pilots is inserted at the end of the transmitted frame in low mobility scenario and as the mobility increases, more pilot OFDM symbols are required. All the other symbols within the transmitted frame are fully allocated by data subcarriers and considered as data symbols. The estimated channel for the data symbols is considered as a weighted summation of the estimated channels at the inserted pilot symbols. According to the used frame structure, the receiver treats the received frame as subframes, thus achieving lower latency at the receiver, since the channel estimation starts upon receiving each subframe instead of waiting for the whole frame. Moreover, in order to gain more transmission data rate (TDR), few pilots can be inserted within the pilot OFDM symbols depending on the channel delay profile.

Further performance improvement can be achieved by integrating optimized SR-CNN or DN-CNN as post processing modules after the WI estimators. The proposed WI estimators enjoy low-complexity and robustness, and achieve good performance in high-mobility vehicular scenarios. Additionally, the proposed WI estimators contribute in latency reduction at the receiver, besides gaining more transmission data rates.

To the best of our knowledge, there is no recent FBF estimators that modify the IEEE 802.11p pilot allocation, how-

ever, the work proposed in [18], [19] employ modified frame structure but for SBS channel estimation, where a midamble pilot symbol is inserted frequently within the transmitted frame and used for estimating the channel for all successive symbols.

The contributions of this paper can be summarized as follows:

- Proposing a hybrid, adaptive, low-complexity, and robust WI channel estimators that modify the pilot allocation schemes of the IEEE 802.11p within the transmitted frame and adapt the employed scheme according to the mobility condition.
- Deriving analytically the expression of the employed interpolation matrix for the proposed pilot allocation schemes.
- Providing a brief overview of the CNN networks, especially those used in the state-of-the-art of the channel estimation, such as SR-CNN, DN-CNN.
- Integrating an optimized SR-CNN and DN-CNN networks on top of the WI estimators to enhance the bit error rate (BER) and normalized mean-squared error (NMSE) performance.
- Showing that the proposed WI estimators outperform ChannelNet and TS-ChannelNet estimators in terms of latency and transmission data rates.
- Providing a detailed computational complexity analysis for the studied channel estimators, where we show that the proposed WI estimators outperform the benchmarked estimators with substantial reduction in complexity.

The remainder of this paper is organized as follows: in Section II, the IEEE 802.11p standard and the system model are described. Section III provides a detailed description of the recently proposed ChannelNet and TS-ChannelNet estimators. The proposed WI estimators, as well as the analytical interpolation matrices derivations are presented in Section IV. Then, a detailed overview of the CNN based channel estimation and the proposed optimized architectures is provided in Section V. In Section VI, simulation results are presented for different vehicular channel models conditions using different modulation orders, where the performance of the proposed estimators is evaluated in terms of BER, NMSE, TDR, and latency. Detailed computational complexity analysis is provided in Section VII. Finally, conclusions are given in Section VIII.

II. SYSTEM MODEL DESCRIPTION

IEEE 802.11p is an international standard that manages wireless access in vehicular environments. IEEE 802.11p defines the communications between high-speed vehicles (vehicle-to-vehicle (V2V)) and between the vehicles and the roadside infrastructure (vehicle-to-infrastructure (V2I)) in the licensed intelligent transportation systems band [20]. IEEE 802.11p uses OFDM transmission scheme with $K = 64$ total subcarriers. $K_{\text{on}} = 52$ active subcarriers are used, and they are divided into $K_{\text{d}} = 48$ data subcarriers and $K_{\text{p}} = 4$ pilot subcarriers. The remaining 12 subcarriers are used as a guard band. IEEE 802.11p standard employs two long training symbols (LTS) preamble symbols at the beginning of the transmitted frame that are used for signal detection

Table I: IEEE 802.11p physical layer specifications.

| Parameter | IEEE 802.11p |
|--------------------------------|--------------|
| Bandwidth | 10 MHz |
| Guard interval duration | 1.6 μ s |
| Symbol duration | 8 μ s |
| Short training symbol duration | 1.6 μ s |
| Long training symbol duration | 6.4 μ s |
| Total subcarriers | 64 |
| Pilot subcarriers | 4 |
| Data subcarriers | 48 |
| Null subcarriers | 12 |
| Subcarrier spacing | 156.25 KHz |

and channel estimation at the receiver. Moreover, it supports transmitting data at different data rates employing different modulation orders. Table I shows the IEEE 802.11p physical layer main specifications. A detailed discussion of the IEEE 802.p standard and all its features is presented in [3].

In this paper, we assume perfect synchronization at the receiver, and we consider a frame that consists of two LTS at the beginning followed by I OFDM data symbols. The received OFDM symbol $\tilde{\mathbf{y}}_i[k]$ can be expressed in terms of the transmitted OFDM symbol $\tilde{\mathbf{x}}_i[k]$ as follows

$$\tilde{\mathbf{y}}_i[k] = \tilde{\mathbf{h}}_i[k]\tilde{\mathbf{x}}_i[k] + \tilde{\mathbf{v}}_i[k], \quad k \in \mathcal{K}_{\text{on}}. \quad (1)$$

Here, $\tilde{\mathbf{v}}_i[k]$ denotes the noise of the k -th subcarrier in the i -th OFDM symbol. $\tilde{\mathbf{h}}_i[k]$ represents the time variant frequency response of the channel for the i -th OFDM symbol. Moreover, let $\tilde{\mathbf{Y}}[k, i] \in \mathbb{C}^{K_{\text{on}} \times I}$ and $\tilde{\mathbf{X}}[k, i] \in \mathbb{C}^{K_{\text{on}} \times I}$ be the received and transmitted OFDM frame respectively. Then, (1) can be expressed in a matrix form as follows

$$\tilde{\mathbf{Y}}[k, i] = \tilde{\mathbf{H}}[k, i]\tilde{\mathbf{X}}[k, i] + \tilde{\mathbf{V}}[k, i], \quad k \in \mathcal{K}_{\text{on}}, \quad (2)$$

where $\tilde{\mathbf{V}}[k, i] \in \mathbb{C}^{K_{\text{on}} \times I}$ and $\tilde{\mathbf{H}} \in \mathbb{C}^{K_{\text{on}} \times I}$ denote the noise and the time variant frequency response of the channel for all symbols within the transmitted OFDM frame respectively.

III. SOA OF FBF CHANNEL ESTIMATORS

In the section, the conventional 2D LMMSE estimator and recently proposed CNN-based estimators schemes are presented and discussed.

A. Conventional 2D LMMSE estimator

The basic idea of the conventional 2D LMMSE estimation is to employ the channel correlation matrices and the noise power in the channel estimation of each data subcarrier within the received OFDM frame, such that

$$\hat{\mathbf{h}}_{\text{LMMSE}_i} = \mathbf{R}_{\tilde{\mathbf{h}}_i, \tilde{\mathbf{h}}_p} (\mathbf{R}_{\tilde{\mathbf{h}}_p, \tilde{\mathbf{h}}_p} + \sigma^2 \mathbf{I}')^{-1} \tilde{\mathbf{h}}_{\text{LS}}. \quad (3)$$

$\mathbf{R}_{\tilde{\mathbf{h}}_i, \tilde{\mathbf{h}}_p} \in \mathbb{C}^{K_p I \times 1}$ represents the k -th column of the cross correlation matrix between the real channel vectors $\tilde{\mathbf{h}}_i$ and $\tilde{\mathbf{h}}_p$ at the data and the pilot subcarriers within the received OFDM frame respectively. Moreover, $\mathbf{R}_{\tilde{\mathbf{h}}_p, \tilde{\mathbf{h}}_p}$ denotes the auto

correlation matrix of $\tilde{\mathbf{h}}_p$, $\mathbf{I}' \in \mathbb{R}^{K_p I \times K_p I}$ is the identity matrix, and $\tilde{\mathbf{h}}_{\text{LS}} = \text{vec} \left\{ \hat{\tilde{\mathbf{H}}}_{\text{LS}} \right\} \in \mathbb{C}^{K_p I \times 1}$ represents the LS estimated channel vector at the pilot subcarriers within the received OFDM frame, where

$$\hat{\tilde{\mathbf{H}}}_{\text{LS}}[k, i] = \frac{\tilde{\mathbf{Y}}[k, i]}{\tilde{\mathbf{P}}[k, i]}, \quad k \in \mathcal{K}_p, \quad 1 \leq i \leq I. \quad (4)$$

$\tilde{\mathbf{P}}[k, i]$ is the frequency domain pre-defined pilot subcarriers and \mathcal{K}_p denotes the allocated sparse pilots indices within the received OFDM symbol.

The conventional 2D LMMSE estimator achieves almost similar performance as the ideal channel, however it suffers from high computational complexity, therefore, making the it impractical in real-time scenario. In contrast, CNN provides a good performance-complexity trade-off by learning the channel statistics while recording an acceptable low computational complexity. Therefore, it is employed in the FBF channel estimation as an alternative to the 2D LMMSE estimator.

B. ChannelNet estimator

In [16], the authors propose a CNN-based channel estimator denoted as ChannelNet scheme, where 2D RBF interpolation is applied as an initial channel estimation. The basic motivation of the 2D RBF interpolation is to approximate multidimensional scattered unknown data, from their surrounding neighbors known data, employing the radial basis function [21]. To do so, the distance function is calculated between every data point to be interpolated and its neighbours, where closer neighbors are assigned higher weights. After that, the RBF interpolated frame is considered as a low resolution image, where SR-CNN is utilized to get better estimation. Finally, in order to alleviate the impact of noise within the high resolution estimated frame, DN-CNN is implemented on top of the SR-CNN resulting in a high resolution, noise alleviated estimated channels. The ChannelNet estimator considers sparsely allocated pilots within the IEEE 802.11p frame and it first applies the LS estimation to the pilot subcarriers within the received OFDM frame as illustrated in (4). After that, The 2D RBF interpolation is obtained by the weighted summation of the distance between each data subcarrier to be interpolated and all the pilot subcarriers within the received OFDM frame, such that

$$\hat{\mathbf{H}}_{\text{RBF}}[k, i] = \sum_{j=1}^{K_p I} \omega_j \Phi(|k - \mathcal{K}_f[j]|, |i - \mathcal{K}_t[j]|). \quad (5)$$

$\mathcal{K}_f = [\mathcal{K}_{p_1}, \dots, \mathcal{K}_{p_I}] \in \mathbb{R}^{1 \times K_p I}$ and $\mathcal{K}_t = [(1) \times K_p, \dots, (I) \times K_p] \in \mathbb{R}^{1 \times K_p I}$ denote the frequency and time indices vectors of the sparsely allocated pilot subcarriers within the received OFDM frame. ω_j represents the RBF weight multiplied by the RBF interpolation function $\Phi(\cdot)$ between the (k, i) data subcarrier and the $(\mathcal{K}_f[j], \mathcal{K}_t[j])$ pilot subcarrier. In [16], the RBF gaussian function is applied, such that

$$\Phi(x, y) = e^{-\frac{(x+y)^2}{r_0}}, \quad (6)$$

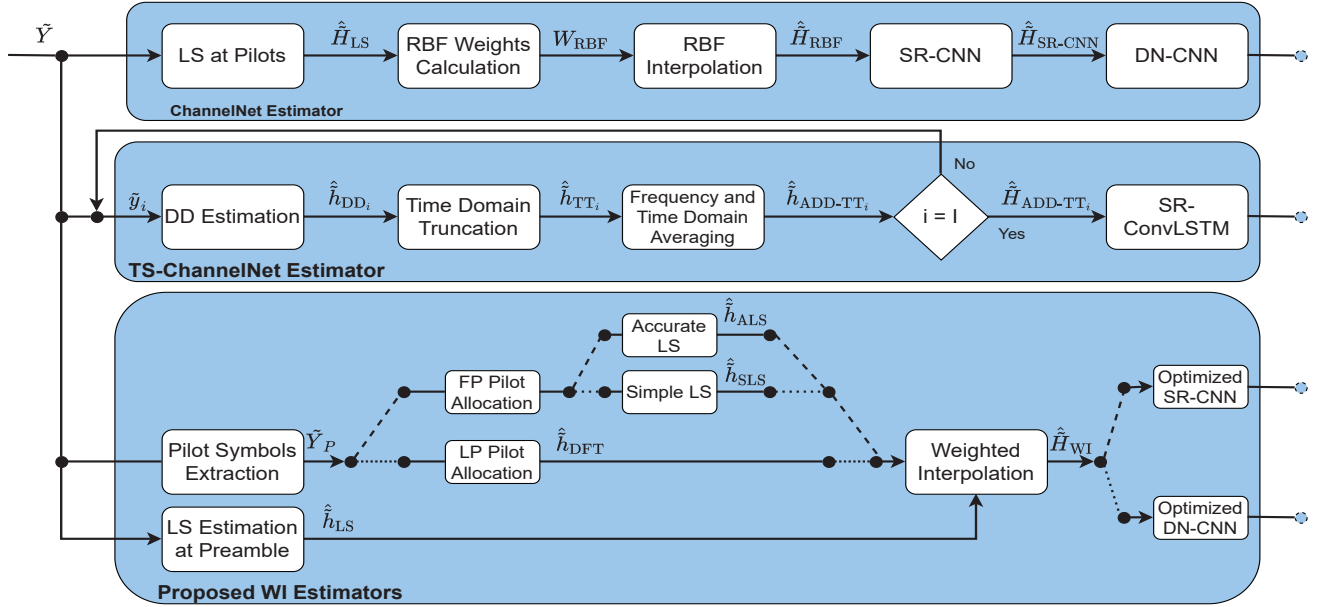


Fig. 1: Proposed WI estimators block diagram.

where r_0 is the 2D RBF scale factor and it varies according to the used RBF function. We note that changing the value of r_0 changes the shape of the interpolation function. Moreover, the RBF weights $\mathbf{w}_{\text{RBF}} = [\omega_1, \dots, \omega_{K_p I}] \in \mathbb{R}^{K_p I \times 1}$ are calculated using the following relation

$$\mathbf{A}_{\text{RBF}} \mathbf{w}_{\text{RBF}} = \bar{\mathbf{h}}_{\text{LS}}, \quad (7)$$

where $\mathbf{A}_{\text{RBF}} \in \mathbb{R}^{K_p I \times K_p I}$ is the RBF interpolation matrix of the pilots subcarriers, with entries $a_{i,j} = \Phi(\mathcal{K}_f[i], \mathcal{K}_t[j])$ where $i, j = 1, \dots, K_p I$. After computing \mathbf{W}_{RBF} , the RBF estimated channel for every data subcarriers within the received OFDM frame can be calculated as shown in (5). Finally, the RBF interpolation estimated frame $\hat{\mathbf{H}}_{\text{RBF}}$ is fed as an input to SR-CNN and DN-CNN in order to improve the channel estimation accuracy, and alleviate the noise impact.

The ChannelNet estimator limitations lie in: (i) 2D RBF interpolation high computational complexity that results from the computation of (7) for the channel estimation of each data subcarrier. (ii) The 2D RBF function and scale factor should be optimized according to the channel variations. (iii) The integrated SR-CNN and DN-CNN architectures have considerable computational complexity. We note that, the ChannelNet estimator uses a fixed RBF function and scale factor, therefore, it suffers from considerable performance degradation especially in low SNR regions where the impact of noise is dominant, and high mobility vehicular scenarios, where the channel varies rapidly within the OFDM frame.

C. TS-ChannelNet estimator

TS-ChannelNet [17] is based on applying ADD-TT interpolation to the received OFDM frame. After that, an accurate estimation is obtained by implementing SR-ConvLSTM network in order to track vehicular channel variations by learning the time and frequency correlations of the vehicular channel. First, the LS estimation is applied using the two LTS

received preambles denoted as $\tilde{\mathbf{y}}_{\text{LTS}_1}[k]$, and $\tilde{\mathbf{y}}_{\text{LTS}_2}[k]$, and the predefined frequency domain preamble sequence $\tilde{\mathbf{p}}[k]$ such that

$$\hat{\mathbf{h}}_{\text{LTS}}[k] = \frac{\tilde{\mathbf{y}}_{\text{LTS}_1}[k] + \tilde{\mathbf{y}}_{\text{LTS}_2}[k]}{2\tilde{\mathbf{p}}[k]}. \quad (8)$$

After that, decision-directed (DD) channel estimation is applied by employing the demapped data subcarriers of the previous received OFDM symbol to estimate the channel for the current OFDM symbol. The DD estimation consists of the following steps

- 1) Equalization: the i -th received OFDM symbol is equalized by the previously DD estimated channel, such that

$$\tilde{\mathbf{y}}_{\text{eq}_i}[k] = \frac{\tilde{\mathbf{y}}_i[k]}{\hat{\mathbf{h}}_{\text{ADD-TT}_{i-1}}[k]}, \quad \hat{\mathbf{h}}_{\text{ADD-TT}_0}[k] = \hat{\mathbf{h}}_{\text{LS}}[k]. \quad (9)$$

- 2) Demapping: $\tilde{\mathbf{y}}_{\text{eq}_i}[k]$ is demapped to the nearest constellation point to obtain $\tilde{\mathbf{d}}_i[k]$.
- 3) DD estimation: update the DD estimated channel for the i -th received OFDM symbol, by dividing $\tilde{\mathbf{y}}_i[k]$ with $\tilde{\mathbf{d}}_i[k]$ as expressed below

$$\hat{\mathbf{h}}_{\text{DD}_i}[k] = \frac{\tilde{\mathbf{y}}_i[k]}{\tilde{\mathbf{d}}_i[k]}. \quad (10)$$

It is worth mentioning that the DD estimation suffers from a considerable accumulated demapping error that is enlarged within the frame, especially at low SNR region. Therefore, to reduce its impact, this error $\hat{\mathbf{h}}_{\text{DD}_i}[k]$ is converted to $\hat{\mathbf{h}}_{\text{DD}_i}$ in the time domain using inverse fast Fourier transformation (IFFT), such that

$$\hat{\mathbf{h}}_{\text{DD}_i} = \mathbf{F}_K^H \hat{\mathbf{h}}_{\text{DD}_i}, \quad (11)$$

where $\mathbf{F}_K \in \mathbb{C}^{K \times K}$ denotes the K -DFT matrix. Then, by truncating $\hat{\mathbf{h}}_{\text{DD}_i}$ to the significant L channel taps, i.e.

$$\hat{\mathbf{h}}_{\text{DD}_i,L} = \hat{\mathbf{h}}_{\text{DD}_i}(1:L). \quad (12)$$

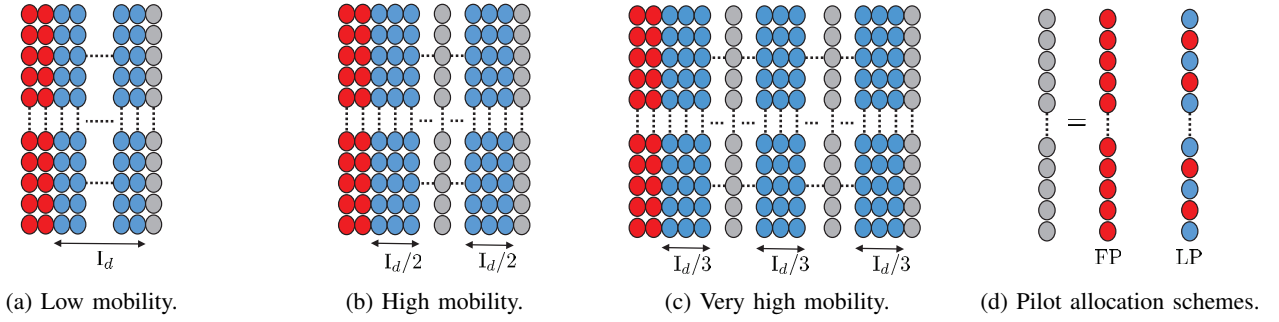


Fig. 2: Proposed IEEE 802.11p frame structure employing different pilot allocation schemes.

After that, $\hat{\mathbf{h}}_{\text{DD}_{i,L}}$ is converted back to the frequency domain such that

$$\hat{\mathbf{h}}_{\text{TT}_i} = \mathbf{F}_{\text{on}} \hat{\mathbf{h}}_{\text{DD}_{i,L}}, \quad (13)$$

where $\mathbf{F}_{\text{on}} \in \mathbb{C}^{K_{\text{on}} \times L}$ represents the scaled DFT matrix that is used to convert $\hat{\mathbf{h}}_{\text{DD}_{i,L}}$ back to frequency domain, which is obtained by selecting K_{on} rows, and L columns from the K -DFT matrix. Applying the average time truncation operation to $\hat{\mathbf{h}}_{\text{DD}_i}[k]$ alleviates the effect of noise and enlarged demapping error. Moreover, $\hat{\mathbf{h}}_{\text{TT}_i}[k]$ estimated channel is further improved by applying frequency and time domain averaging consecutively as follows

$$\hat{\mathbf{h}}_{\text{FTT}_i}[k] = \sum_{\lambda=-\beta}^{\lambda=\beta} \omega_{\lambda} \hat{\mathbf{h}}_{\text{TT}_i}[k + \lambda], \quad \omega_{\lambda} = \frac{1}{2\beta + 1}. \quad (14)$$

The final ADD-TT channel estimates is updated using time averaging between the previously ADD-TT estimated channel and the frequency averaged channel in (14), such that

$$\hat{\mathbf{h}}_{\text{ADD-TT}_i}[k] = (1 - \alpha) \hat{\mathbf{h}}_{\text{ADD-TT}_{i-1}}[k] + \alpha \hat{\mathbf{h}}_{\text{FTT}_i}[k]. \quad (15)$$

Motivated by the fact that the vehicular time-variant channel can be modeled as a time-series forecasting problem, where historical data can be used to predict future observations [22]. The authors in [17] apply SR-ConvLSTM network on top of the ADD-TT interpolation, where convolutional layers are added to the LSTM network in order to capture more vehicular channel features, hence improving the estimation performance. Therefore, the ADD-TT estimated channel for the whole received frame is modeled as a low resolution image, and then SR-ConvLSTM network is used after the ADD-TT interpolation. Unlike ChannelNet estimator where two CNNs are employed, TS-ChannelNet estimator uses only SR-ConvLSTM network, thus the overall computational complexity is relatively decreased. However, TS-ChannelNet still suffers from high computational complexity due to integrating both LSTM and CNN in one network.

IV. PROPOSED WEIGHTED INTERPOLATION ESTIMATORS

The proposed WI estimators employ different pilot allocation schemes to the IEEE 802.11p frame structure as shown in Fig. 2. These frame structures are proposed motivated by the fact the vehicle velocity is a known parameter that can be exchanged between all vehicular network nodes. For

example, in urban environments (inside cities) the car velocity must not exceed 40 Kmphr, thus vehicles and road side units within this environment will use the low mobility frame structure, and similarly for highways environment. Therefore, the frame structure selection is performed in an adaptive manner according to the vehicle velocity. The proposed frame structures preserve the first two LTS preamble symbols similar to IEEE 802.11p standard, so that the synchronization and frame detection processes will not be affected by the proposed modifications. Moreover, only P pilot OFDM symbols are required in the transmitted frame, such that $\tilde{\mathbf{Y}}_P = [\tilde{\mathbf{y}}_1^{(p)}, \dots, \tilde{\mathbf{y}}_q^{(p)}, \dots, \tilde{\mathbf{y}}_P^{(p)}] \in \mathbb{C}^{K_{\text{on}} \times P}$. q denotes the inserted pilot symbol index, where $1 \leq q \leq P$. The other $I_d = I - P$ OFDM data symbols are preserved for actual data transmission.

The proposed estimators proceed according to two criteria: (i) Vehicular mobility condition, where three frame structures are defined as shown in Fig. 2. (ii) The pilot allocation schemes, where there exist two schemes as shown in 2d, such that

- **Full pilot allocation (FP):** where K_{on} pilots are inserted within each pilot symbol and LS estimation is applied to estimate the channel for each inserted pilot symbol. The IEEE 802.11p basic LS denoted as *simple LS (SLS)* estimation is applied using the two LTS received preambles $\tilde{\mathbf{y}}_{\text{LTS}_1}[k]$, and $\tilde{\mathbf{y}}_{\text{LTS}_2}[k]$ as shown in (8), and using each received pilot symbol such that

$$\hat{\mathbf{h}}_{\text{SLS}_q}[k] = \frac{\tilde{\mathbf{y}}_q^{(p)}[k]}{\tilde{\mathbf{p}}[k]} = \tilde{\mathbf{h}}_q[k] + \tilde{\mathbf{v}}_q[k], \quad k \in \mathcal{K}_{\text{on}}. \quad (16)$$

$\tilde{\mathbf{v}}_q[k]$ represents the noise at the q -th received pilot symbol. On the other hand, the *accurate LS (ALS)* relies on the fact that $\tilde{\mathbf{h}}_q = \mathbf{F}_{\text{on}} \mathbf{h}_{q,L}$, where $\mathbf{h}_{q,L} \in \mathbb{C}^{L \times 1}$ denotes the channel impulse response at the q -th received pilot symbol that can be estimated as follows

$$\hat{\mathbf{h}}_{q,L} = \mathbf{F}_{\text{on}}^{\dagger} \hat{\mathbf{h}}_{\text{SLS}_q} = \mathbf{h}_{q,L} + \mathbf{F}_{\text{on}}^{\dagger} \tilde{\mathbf{v}}_q, \quad k \in \mathcal{K}_{\text{on}}, \quad (17)$$

where $\mathbf{F}_{\text{on}}^{\dagger} = [(\mathbf{F}_{\text{on}}^H \mathbf{F}_{\text{on}})^{-1} \mathbf{F}_{\text{on}}^H]$ is the pseudo inverse matrix of \mathbf{F}_{on} . After that, the accurate LS estimation can be obtained by applying the discrete Fourier transform (DFT) interpolation of $\hat{\mathbf{h}}_{q,L}$ as follows

$$\hat{\mathbf{h}}_{\text{ALS}_q} = \mathbf{F}_{\text{on}} \hat{\mathbf{h}}_{q,L} = \mathbf{F}_{\text{on}} \mathbf{h}_{q,L} + \mathbf{W}_{\text{ALS}} \tilde{\mathbf{v}}_q, \quad k \in \mathcal{K}_{\text{on}}. \quad (18)$$

$\mathbf{W}_{\text{ALS}} = \mathbf{F}_{\text{on}} \mathbf{F}_{\text{on}}^\dagger$ denotes the accurate LS DFT interpolation matrix.

- **L pilot allocation (LP):** this allocation is motivated by the fact that vehicular channel models mainly consists of 12 taps, then calculating $\mathbf{h}_{q,L}$ requires only L pilots in each pilot symbol, and can be computed as follows

$$\hat{\mathbf{h}}_{q,L} = \mathbf{F}_p^\dagger \hat{\mathbf{h}}_{\text{LS}_q} = \mathbf{h}_{q,L} + \mathbf{F}_p^\dagger \tilde{\mathbf{v}}_q, \quad k \in \mathcal{K}_p. \quad (19)$$

$\hat{\mathbf{h}}_{\text{LS}_q}$ here represents the LS estimated channel for the q -th pilot symbol at the L equally spaced inserted pilot subcarriers, where $K_p = L$ and $\mathbf{F}_p^\dagger = [(\mathbf{F}_p^H \mathbf{F}_p)^{-1} \mathbf{F}_p^H]$ is the pseudo inverse matrix of $\mathbf{F}_p \in \mathbb{C}^{K_p \times L}$ which refers to the truncated DFT matrices obtained by selecting \mathcal{K}_p rows, and L columns from the K -DFT matrix. After that, DFT interpolation based estimation can be used employing $\hat{\mathbf{h}}_{q,L}$ to estimate the channel for the q -th pilot symbol as follows

$$\hat{\mathbf{h}}_{\text{DFT}_q} = \mathbf{F}_{\text{on}} \hat{\mathbf{h}}_{q,L} = \mathbf{F}_{\text{on}} \mathbf{h}_{q,L} + \mathbf{W}_{\text{DFT}} \tilde{\mathbf{v}}_q, \quad k \in \mathcal{K}_{\text{on}}.$$

Here $\mathbf{W}_{\text{DFT}} = \mathbf{F}_{\text{on}} \mathbf{F}_p^\dagger$ is the DFT interpolation matrix.

After estimating the channel at the pilot symbols according to the selected configuration, the proposed WI estimators apply the following two steps

- **Grouping:** The estimated channels of the P pilot symbols are grouped into P matrices, such that

$$\hat{\mathbf{H}}_q = [\hat{\mathbf{h}}_{q-1}, \hat{\mathbf{h}}_q], \quad q = 1, \dots, P. \quad (21)$$

We note that $\hat{\mathbf{h}}_0 = \hat{\mathbf{h}}_{\text{LTS}}$ and $\hat{\mathbf{h}}_q$ refers to the implemented LS estimation according to the utilized pilot allocation scheme.

- **Weighted Interpolation estimation:** According to the grouped $\hat{\mathbf{H}}$ matrices, the received frame can be divided into P sub-frames, where f denotes the sub-frame index, such that $1 \leq f \leq P$. Therefore, the estimated channel for the i -th received OFDM symbol within each f -th sub-frame can be expressed as follows

$$\hat{\mathbf{H}}_{\text{WI}_f} = \hat{\mathbf{H}}_f \mathbf{C}_f \quad (22)$$

where $\hat{\mathbf{H}}_f \in \mathbb{C}^{K_{\text{on}} \times 2}$ denotes LS estimated channels at the pilot symbols within the f -th sub-frame. $\mathbf{C}_f \in \mathbb{R}^{2 \times I_f}$ denotes the interpolation weights of the I_f OFDM data symbols within the f -th sub-frame. We note that I_f is calculated according to the selected frame structure where it is equal to I_d , $\frac{I_d}{2}$, and $\frac{I_d}{3}$ in low, high, and very high mobility scenarios respectively.

Therefore, the estimated channel for the received OFDM data symbols can be seen as a weighted summation of the $\hat{\mathbf{H}}_f$ matrix. \mathbf{C}_f interpolation weights are calculated by minimizing the mean squared error (MSE) between the ideal channel $\tilde{\mathbf{H}}_f$, and the LS estimated channel at the OFDM pilot symbols $\hat{\mathbf{H}}_f$ as derived in [23]. This minimization results in \mathbf{C}_f expressed in (20), where $J_0(\cdot)$ is the zeroth order Bessel function of the first kind, T_s is the received OFDM data symbol duration, and E_q denotes the overall noise of the estimated channel at the q -th pilot symbol. E_q is calculated according to the chosen pilot allocation scheme, and employed LS estimation, where it equals to σ^2 , $\sigma^2 \text{trace} \{ \mathbf{W}_{\text{ALS}} \mathbf{W}_{\text{ALS}}^H \}$, and $\sigma^2 \text{trace} \{ \mathbf{W}_{\text{DFT}} \mathbf{W}_{\text{DFT}}^H \}$ for SLS, ALS and LP LS estimation respectively.

In fact, the weight elements of \mathbf{C}_f for all the sub-frames can be calculated offline for several vehicular scenarios by employing different f_d and I_d values, Therefore, decreasing the online complexity and making the proposed WI estimators more efficient in real cases scenarios. Moreover, a trade-off between the mobility condition controlled by f_d , the inserted pilot symbols P , and the used frame length I_d should be considered. As the mobility increases, more pilot symbols should be inserted within the transmitted frame. Therefore, this trade-off is managed by the vehicular application requirements, so that the transmitter will adapt the transmission parameters according to these requirements. Fig. 1 shows the proposed WI estimators block diagram.

V. CNN-BASED CHANNEL ESTIMATION

A. CNN Overview

A CNN is a neural network that consists of one or more layers and is used mainly for processing structured arrays of data such as images [24]. CNN network has generally become the state of the art for many visual applications such as image classification, due to its great ability in extracting patterns from the input image. CNN is a set of several layers stacked together in order to accomplish the required task. These layers include

- **Input layer:** It represents the 2D or 3D input image.
- **Convolutional layer:** It is used for feature extraction that is implemented using predefined filters called kernels. The kernel scans the input matrix to fill the output matrix denoted as feature map.
- **Pooling layer:** This layer is applied to reduce the number of parameters when the images are too large. Pooling operation is also called sub-sampling or down-sampling which reduces the dimensionality of each feature map but retains important information.

$$\begin{aligned} \mathbf{C}_f &= \mathbf{E} \left[\tilde{\mathbf{H}}_f \hat{\mathbf{H}}_f^H \right] \left[\mathbf{E} \left[\hat{\mathbf{H}}_f \hat{\mathbf{H}}_f^H \right] \right]^{-1} = \left[\mathbf{E} \left[\tilde{\mathbf{H}}_f \hat{\mathbf{h}}_q^H \right] \quad \mathbf{E} \left[\tilde{\mathbf{H}}_f \hat{\mathbf{h}}_{q+1}^H \right] \right] \begin{bmatrix} \mathbf{E} \left[\left\| \tilde{\mathbf{h}}_q \right\|^2 \right] + E_q & \mathbf{E} \left[\tilde{\mathbf{h}}_q \tilde{\mathbf{h}}_{q+1}^H \right] \\ \mathbf{E} \left[\tilde{\mathbf{h}}_{q+1} \tilde{\mathbf{h}}_q^H \right] & \mathbf{E} \left[\left\| \tilde{\mathbf{h}}_{q+1} \right\|^2 \right] + E_{q+1} \end{bmatrix}^{-1} \\ &= \left[J_0(2\pi f_d(f-1)T_s) \quad J_0(2\pi f_d(I_f+1-f)T_s) \right] \begin{bmatrix} 1 + E_{\Phi_q} & J_0(2\pi f_d I_f T_s) \\ J_0(2\pi f_d I_f T_s) & 1 + E_{q+1} \end{bmatrix}^{-1}. \end{aligned} \quad (20)$$

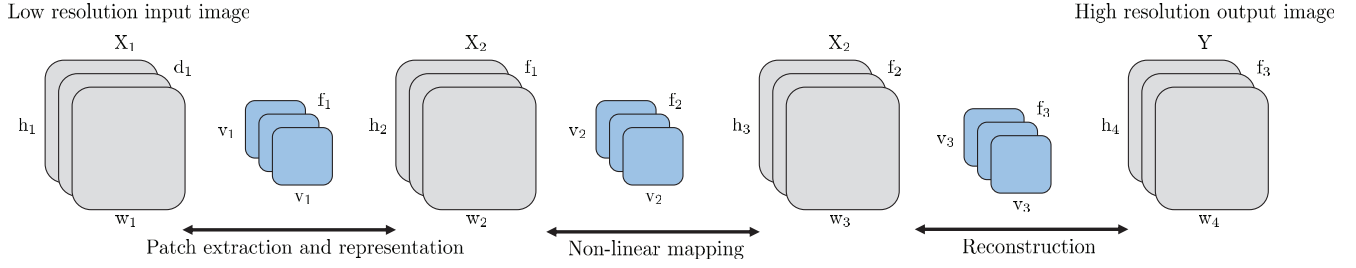


Fig. 3: SR-CNN architecture.

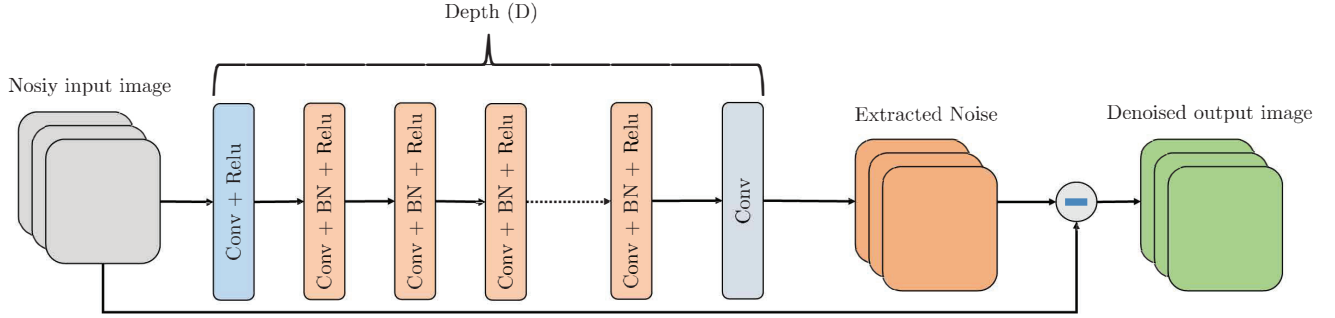


Fig. 4: DN-CNN architecture.

- **Activation layer:** An activation function is applied to the CNN layer's output. The most used common activation function is the ReLU function. It is used to introduce non-linear processing to the CNN architecture since the input-output CNN pairs relation might be non-linear.
- **Fully connected layer:** This layer forms the last block of the CNN architecture, and it is utilized mainly in classification problems. It is a simple feed forward neural network layer, consisting of one or more hidden layers.
- **Batch normalization:** It is used to make the CNN output more stable by normalizing the output of each layer. Moreover, batch normalisation layer reduces overfitting and speeds up the CNN training.
- **Output layer:** This layer is configured according to the studied problem, for example, in classification problems the CNN output layer is a fully connected layer with softmax activation function, while in regression problems, the CNN output does not use any activation function.

Automatically detecting meaningful features given only an image and its label is not a trivial task. Therefore, the convolutional layers are the most essential elements within any CNN since the convolutional layers learn such complex features by updating their kernels values during training phase to obtain the best input-output mapping. This update is performed by minimizing the CNN loss function that measures how far are the inputs from the outputs. After that, the performance of the trained CNN model is evaluated in the testing phase where new unobserved inputs are fed to the trained CNN model.

B. SR-CNN and DN-CNN Networks

Besides classifications problems, there are special CNN architectures denoted as SR-CNN and DN-CNN that are mainly

used for regression problems. SR-CNN [25] is used to improve the quality of the input image, where the input-output mapping is represented as a deep CNN that takes the low-resolution image as the input and outputs the high-resolution one. SR-CNN consists of three convolutional layers as shown in Fig. 3: (i) Patch extraction and representation, (ii) Non-linear mapping, and (iii) Reconstruction. Let $\mathbf{X}_m \in \mathbb{R}^{h_m \times w_m \times d_m}$ be the input image to the m -th SR-CNN convolutional layer, where h_m , w_m , and d_m denote the image height, width, and depth respectively, and $\mathbf{X}_1 \in \mathbb{R}^{h_1 \times w_1 \times d_1}$ represents the input image. The convolutional filters are denoted by the weights and biases matrices $\mathbf{W}_m \in \mathbb{R}^{v_m \times v_m \times d_m \times f_m}$ and $\mathbf{B}_m \in \mathbb{R}^{f_m \times 1}$, where v_m refers to the convolutional filter size, and f_m denotes the total number of filters used in the SR-CNN m -th convolutional layer. The m -th convolution layer applies f_m convolution operation to its \mathbf{X}_m input image as follows

$$\mathbf{Y}_{(m)} = \mathbf{F}_{(m)}\left(\mathbf{W}_{(m)} * \mathbf{X}_{(m)}\right), \quad \mathbf{X}_{(m+1)} = \mathbf{Y}_{(m)}, \quad (23)$$

where $\mathbf{F}_{(m)}$ represents the activation function applied to the m -th convolutional layer output. We note that, it is possible to add more convolutional layers to increase the non-linearity mapping between the SR-CNN convolutional layers, but this will increase the complexity of the SR-CNN model and thus demands more training time. The integrated SR-CNN architecture related mainly to the investigated problem and the dataset structure, whereas the best SR-CNN architecture and parameters can be fine tuned by intensive experiments or by using the grid search algorithm [26] that selects the best suitable SR-CNN hyper parameters in terms of both performance and complexity. The loss function of the SR-CNN is represented by the MSE between the reconstructed high

Table II: Optimized SR-CNN and DN-CNN parameters.

| Parameter | Values |
|--|------------------------------------|
| Input/Output dimensions | $2K_{\text{on}} \times I \times 1$ |
| SRCNN $(v_1, f_1; v_2, f_2; v_3, f_3)$ | (9,32; 1,16; 5,1) |
| DNCNN (v, f) | (3, 16) |
| DNCNN D | 7 |
| Activation function | ReLU |
| Number of epochs | 250 |
| Training samples | 8000 |
| Testing samples | 2000 |
| Batch size | 128 |
| Optimizer | ADAM |
| Loss function | MSE |
| Learning rate | 0.001 |
| Training SNR | 30 dB |

resolution images \mathbf{Y} , and the true images $\mathbf{Y}^{(T)}$, such that

$$\text{MSE}_{\text{SR-CNN}} = \frac{1}{N_{\text{Train}}} \sum_{i=1}^{N_{\text{Train}}} \left\| \mathbf{Y}_i - \mathbf{Y}_i^{(T)} \right\|^2, \quad (24)$$

where N_{Train} denotes the number of training samples. The minimization of $\text{MSE}_{\text{SR-CNN}}$ is achieved by updating \mathbf{W} and \mathbf{B} matrices such that the best input-output mapping is obtained. To do so, various optimizers can be exploited such that the stochastic gradient descent [27], root mean square prop [28], and adaptive moment estimation (ADAM) [29].

On the other hand, DN-CNN [30] improves the image quality by separating the noise from the input noisy image using a special CNN architecture. After that, the input noisy image is subtracted from the extracted noise resulting in the denoised image. In order to extract the noise from a noisy image, residual learning [31] is applied so that the noise included in the input image is learnt in the DN-CNN training phase by minimizing the following DN-CNN loss function

$$\text{MSE}_{\text{DN-CNN}} = \frac{1}{N_{\text{Train}}} \sum_{i=1}^{N_{\text{Train}}} \left\| \mathbf{V}_i - \mathbf{V}_i^{(T)} \right\|^2, \quad (25)$$

where \mathbf{V}_i represents the extracted noise included in the noisy input image \mathbf{X}_i , and $\mathbf{V}_i^{(T)}$ denotes the exact noise, such that $\mathbf{V}_i^{(T)} = \mathbf{Y}_i^{(N)} - \mathbf{X}_i$. As shown in Fig. 4, the DN-CNN employs in the first layer convolution and ReLU activation function to generate the initial feature maps from the input noisy image. After that, successive (Conv + BN + ReLU) layers are considered to extract the noise features, since the analysis performed in [30] shows that integrating convolution with batch normalization followed by ReLU, can gradually separate the clean image structure from the noisy observation through the DN-CNN hidden layers. Finally, a convolutional layer is used for output image reconstruction. In summary, the DN-CNN architecture has two main tasks: the residual learning is used to learn the noise features, and the batch normalization which is incorporated to speed up training as well as to boost the denoising performance. The main complexity of the DN-CNN architecture lies in its depth (D)

that denotes the number of used (Conv+ BN + ReLU) layers. As D increases the complexity of the DN-CNN architecture increases. Similarly to SR-CNN, all the DN-CNN parameters can be tuned using grid search algorithm.

Unlike ChannelNet estimator, where both SR-CNN and DN-CNN networks are used on top of the RBF interpolation, extensive experiments are applied in this paper using the hyper parameters tuning grid search algorithm [26] in order to select the best CNN network configuration corresponding to the mobility condition. Based on the selected CNN parameters, an optimized SR-CNN is employed on top of the WI estimators in a low mobility scenario, whereas an optimized DN-CNN is considered in high mobility one. By doing so, better performance can be achieved with a significant decrease in the overall computational complexity as we discuss in the next sections. We note that employing, low-complexity SR-CNN and DN-CNN architectures is due to the good performance of the initial WI estimation, since as the initial estimation accuracy increases, the CNN computational complexity decreases. Moreover, our investigations show that by employing the proposed WI estimators, there is no need to use both SR-CNN and DN-CNN, since the computational complexity increases without any significant performance gain.

We note that CNNs networks work with real valued data, therefore after applying the WI interpolation estimator, $\hat{\mathbf{H}}_{\text{WI}} \in \mathbb{C}^{K_{\text{on}} \times I_d}$ is converted from complex to real valued domain by stacking the real and imaginary values vertically in one matrix, such that $\hat{\mathbf{H}}_{\text{WI}}^{(R)} \in \mathbb{R}^{2K_{\text{on}} \times I_d}$. Then $\hat{\mathbf{H}}_{\text{WI}}^{(R)}$ is fed as an input to the optimized SR-CNN or DN-CNN according to the mobility scenario. Finally, the output of the employed network is converted back to the complex domain. The optimized SR-CNN and DN-CNN networks are trained on SNR= 30 dB for each mobility scenario, since in high SNR region, the CNN network is able to learn better the channel due to the low noise impact in high SNR region [32]. Moreover, ADAM optimizer is used with MSE loss function. Table II shows the proposed optimized SR-CNN and DN-CNN parameters.

VI. SIMULATION RESULTS

In this section, the performance of the proposed linear WI estimators is evaluated compared to the conventional 2D LMMSE that exploits all the pilots defined in the IEEE 802.11p standard, ChannelNet, and TS-ChannelNet estimators using BER and NMSE. The simulations are conducted employing three vehicular scenarios as shown in Table III. These scenarios are based on the tapped delay line (TDL) vehicular channel models proposed in [33], which are obtained by a measurement campaign that was implemented in metropolitan Atlanta, and can be defined as follows

- **Low mobility:** where VTV Urban Canyon (VTV-UC) vehicular channel model is considered. VTV-UC channel model has been measured between two vehicles moving in a dense urban traffic environment at $V = 45$ Kmph which is equivalent to $f_d = 250$ Hz.
- **High mobility:** This scenario measures the communication channel between two vehicles moving on a highway

Table III: Vehicular channel models characteristics following Jake's Doppler spectrum.

| Channel model | Channel taps | Vehicle velocity [kmph] | Doppler shift [Hz] | Average path gains [dB] | Path delays [ns] |
|---------------|--------------|-------------------------|--------------------|--|--|
| VTV-UC | 12 | 45 | 250 | [0, 0, -10, -10, -10, -17.8, -17.8, -17.8, -21.1, -21.1, -26.3, -26.3] | [0, 1, 100, 101, 102, 200, 201, 202, 300, 301, 400, 401] |
| VTV-SDWW | 12 | 100-200 | 500-1000 | [0, 0, -11.2, -11.2, -19, -21.9, -25.3, -25.3, -24.4, -28, -26.1, -26.1] | [0, 1, 100, 101, 200, 300, 400, 401, 500, 600, 700, 701] |

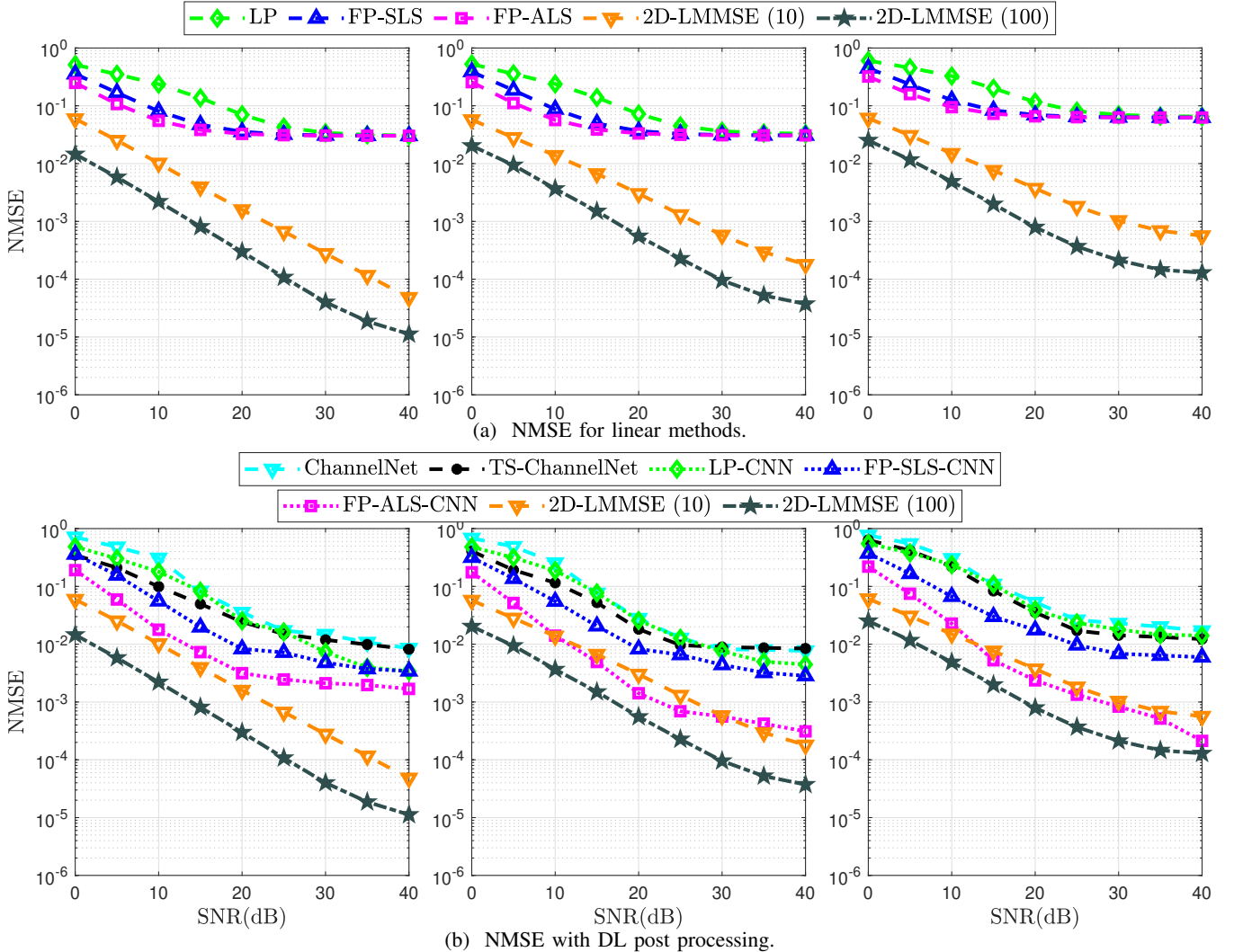


Fig. 5: NMSE employing $I = 100$, mobility from left to right: low ($v = 45$ Kmph, $f_d = 250$ Hz), high ($v = 100$ Kmph, $f_d = 500$ Hz), very high ($v = 200$ Kmph, $f_d = 1000$ Hz). The CNN refers to SRCNN and DNCNN in low and high/very high mobility scenarios, respectively.

having center wall between its lanes. Moreover, the vehicles are moving in the same direction at $V = 100$ Kmph which is equivalent to $f_d = 500$ Hz with a 300–400 m separation distance between them. This vehicular channel model is denoted as VTV Expressway Same Direction with Wall (VTV-SDWW).

- **Very high mobility:** In order to further evaluate the performance of the benchmarked channel estimators, VTV-SDWW vehicular channel model is used with $V = 200$ Kmph which implies $f_d = 1000$.

A. NMSE Evaluation

The NMSE performance of the proposed estimators depends mainly on the employed WI estimation, where using low number of pilots as the case in the proposed WI-LP estimator leads to considerable performance degradation as we can notice in Fig. 5a. Whereas, the proposed WI-FP-SLS and WI-FP-ALS achieve better performance than WI-LP due to the employment of full pilots symbols within the frame. The accuracy of WI-FP-ALS is higher at low SNRs due to the exploitation of the frequency correlation via interpolation at

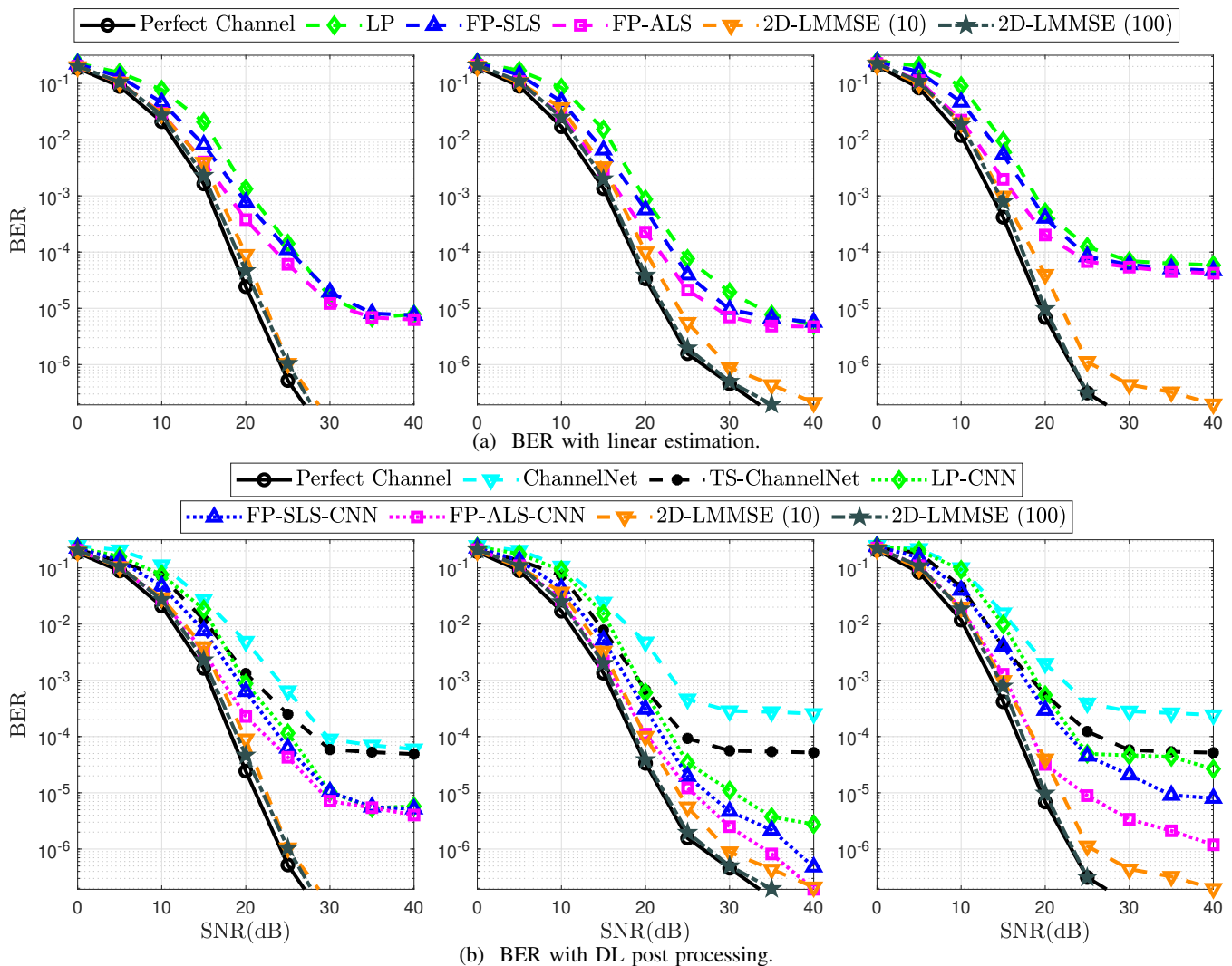


Fig. 6: BER for $I = 100$, QPSK, mobility from left to right: low ($v = 45$ Km/h, $f_d = 250$ Hz), high ($v = 100$ Km/h, $f_d = 500$ Hz), very high ($v = 200$ Km/h, $f_d = 1000$ Hz). The CNN refers to SRCNN and DNCNN in low and high/very high) mobility scenarios, respectively.

the cost of increased complexity. By employing all the pilots in the frame (4 pilots per OFDM symbol), 2D-LMMSE (100) significantly outperforms the WI estimators. In order to reduce the latency, we consider to apply the 2D-LMMSE on subframe basis with a subframe length of 10 OFDM symbols denoted as 2D-LMMSE (10). The estimation accuracy in this case is reduced by 10 dB with the decrease of the subframe length. Nevertheless, the accuracy of 2D-LMMSE (10) is better than the WI estimators due to the high correlation of the pilots.

The performance of the WI approaches is reduced by the increase of mobility. This mainly depends on the frame structure, and the Doppler spread through the coherence interval between the pilots symbols, which can be defined as

$$\Delta_C = \Delta_p I f_d T_s, \quad (26)$$

where T_s is the OFDM symbol duration, $\Delta_p I$ number of symbols between successive pilot symbols. A smaller value indicates more correlation between the pilots that can be exploited in the time-domain interpolation. Based on the frame

structure shown in Fig. 2, at very high mobility ($f_d = 1000$ Hz), $\Delta_p = 33$ symbols, which corresponds to $\Delta_C = 33 \times 10^3 T_s$, whereas in low and high mobility, $\Delta_C = 25 \times 10^3 T_s$. As a consequence, it is clear that the NMSE at very high mobility increases compared to the other cases because of the larger coherence interval between the pilots. The high mobility case is only influenced by the Doppler interference, which can be observed from the error floor at high SNRs. This is also the situation in both 2D-LMMSE estimators as the NMSE increases with the increase of mobility at high SNRs.

Fig. 5b shows the NMSE performance of the CNN-based estimators. It can be noticed that the proposed estimators are able to outperform the ChannelNet and TS-ChannelNet estimators in different mobility scenarios. It is worth mentioning that using CNN post processing after the WI-based estimators reveals a considerable robustness against mobility. This is due to the ability of the optimized SR-CNN and DN-CNN in significantly alleviating the impact of Doppler interference. The DL-based post processing networks provide a performance

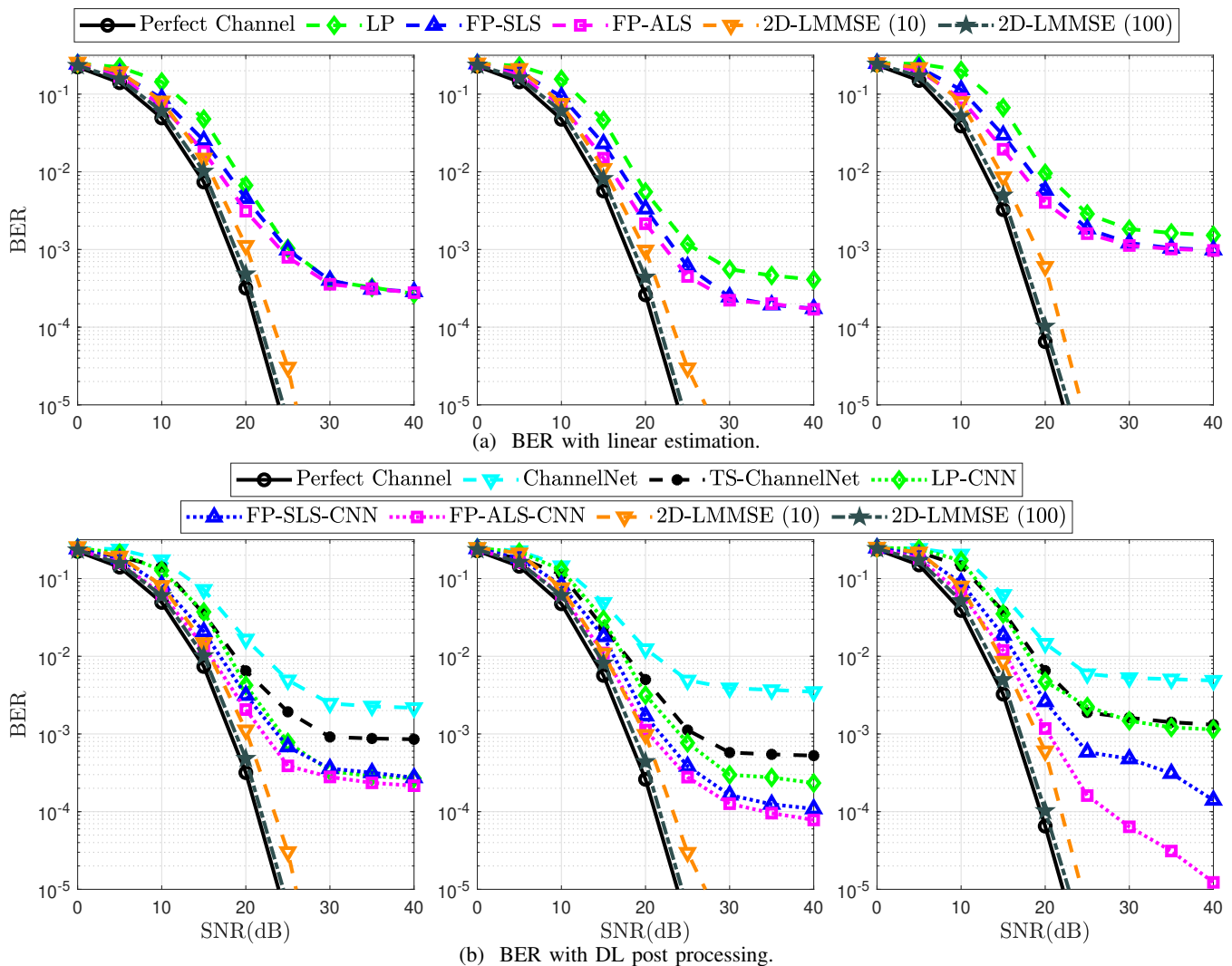


Fig. 7: BER for $I = 100$, 16QAM, mobility from left to right: low ($v = 45$ Kmph, $f_d = 250$ Hz), high ($v = 100$ Kmph, $f_d = 500$ Hz), very high ($v = 200$ Kmph, $f_d = 1000$ Hz). The CNN refers to SRCNN and DNCNN in low and high/very high) mobility scenarios, respectively.

trade-off between the linear WI and 2D-LMMSE using the full pilots in the frame.

B. BER Evaluation

Fig. 6a depicts the BER using the discussed linear estimation techniques employing QPSK. The relative performance follows the same trend of the channel estimation performance shown in Fig. 5a in each mobility case. In general, the impact of the estimation error is lower in low SNR region and this impact increases with the increase of the SNR. Although the NMSE gap between WI-FP-ALS, WI-FP-SLS, WI-LP decreases with the increase of the SNR, the BER gain achieved by WI-FP-ALS is maintained until reaching an error floor. In different mobility scenarios, the trade-off between estimation error and time diversity gain can be observed where the performance is attributed to the main following factors; i) channel estimation error, and ii) time diversity due to increased Doppler spread. The estimation error degrades the BER performance, whereas the time diversity improves it. In

total, the estimation error dominates over the diversity gain. Nevertheless, the case of 2D-LMMSE experiences improvement due to better channel estimation, although the codeword is smaller than that used in WI-FP-ALS.

The impact of the proposed DL-based post processing is shown in Fig. 6b. First, it can be clearly seen that the post processing enhances the BER as a result of enhancing the channel estimation, Fig. 5b. Next, we compare our proposed linear, and DL-enhanced estimation with the SoA DL-based ChannelNet and TS-ChannelNet. We can observe the significant BER performance superiority of the proposed estimators, where WI-LP records similar performance as TS-ChannelNet, while WI-FP-SLS estimator slightly outperforms WI-LP by around 1 dB gain in terms of SNR for a BER = 10^{-3} . On the other hand, the proposed WI-FP-ALS estimator outperforms both ChannelNet and TS-ChannelNet by around 6 dB and 3 dB gain in terms of SNR for a BER = 10^{-3} , respectively.

The performance of ChannelNet and TS-ChannelNet accounts of the predefined fixed parameters in the applied

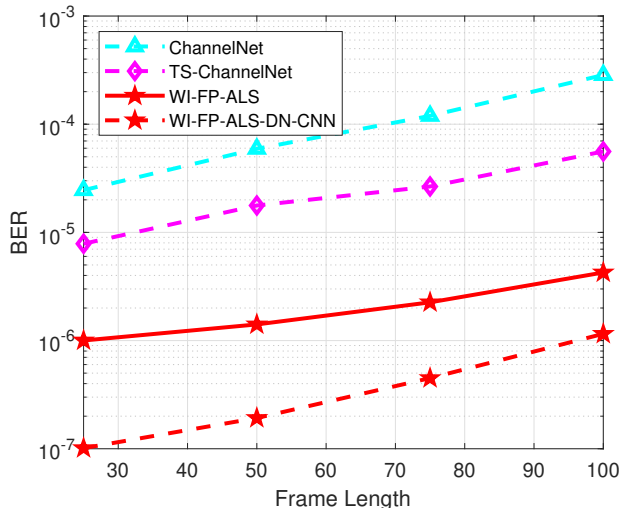


Fig. 8: BER performance of VTV-SDWW high mobility channel model employing QPSK and different frame lengths.

interpolation scheme, where the RBF interpolation function and the ADD-TT frequency and time averaging parameters need to be updated in a real-time manner. Moreover, the ADD-TT interpolation employs only the previous and the current pilot subcarriers for the channel estimation at each received OFDM symbol. On the contrary, in the proposed WI estimators there are no fixed parameters, the time correlation between the previous and the future pilot symbols is considered in the WI interpolation matrix (20), and the estimated channel at all channel taps is considered in the overall estimation. These aspects lead to the proposed estimators performance superiority.

In addition, ChannelNet and TS-ChannelNet estimators suffer from a considerable performance degradation that is dominant in very high mobility scenario. However, the proposed estimators show a robustness against high mobility, this is mainly due to the accuracy of the WI interpolation, combined with optimized SR-CNN and DN-CNN. Although CNN processing is applied in the ChannelNet and TS-ChannelNet, this post CNN processing is not able to perform well due to the high estimation error of the 2D RBF and ADD-TT interpolation techniques in the initial estimation. As a result, we can conclude that employing robust initial estimation as the proposed WI interpolation schemes allows the CNN to learn better the channel correlation with lower complexity, thus improving the channel estimation. Finally, we note that the performance of the 2D-LMMSE estimator is comparable to the performance of ideal channel but it requires huge complexity as we discuss in the next section, which is impractical in real scenario.

C. Frame Length

Fig. 8 shows the BER performance of high mobility vehicular scenario employing QPSK modulation and different frame lengths. It can be clearly noticed that the proposed WI-FP-ALS estimator is able to outperform ChannelNet and TS-

Table IV: BER performance gain (dB) of the proposed WI-FP-ALS compared to the TS-ChannelNet estimator in different mobility scenarios.

| Scheme | Low | | High | | Very High | |
|--------|-----|-------|------|-------|-----------|-------|
| | WI | SRCNN | WI | DNCNN | WI | DNCNN |
| QPSK | 4 | 5 | 3 | 4 | 2 | 5 |
| 16QAM | 2 | 6 | 3 | 4 | 5 | 10 |

ChannelNet for different frame lengths, this is due to the robustness of the proposed WI-FP-ALS estimator, unlike the 2D RBF and ADD-TT interpolation techniques that suffer from a considerable estimation error even when a short frame is considered, which affects the performance of ChannelNet and TS-ChannelNet. Moreover, employing the optimized DN-CNN after the WI-FP-ALS estimator improves significantly the BER performance. Table IV illustrates the performance gain of the proposed WI-FP-ALS-DN-CNN estimator compared to the TS-ChannelNet estimator in high mobility scenario, where employing the optimized DN-CNN leads to 5 dB and 10 dB gain in terms of SNR for a BER = 10^{-3} for QPSK and 16QAM modulation orders respectively as shown in Fig. 7b.

D. CNN Architecture

The ChannelNet estimator employs SR-CNN and DN-CNN after the 2D RBF interpolation. The used SR-CNN consists of three convolutional layers with $(v_1 = 9; f_1 = 64)$, $(v_2 = 1, f_2 = 32)$ and $(v_3 = 5, f_3 = 1)$ respectively. Moreover, the DN-CNN depth is $D = 18$ with $3 \times 3 \times 32$ kernels in each layer. On the other hand, SR-ConvLSTM network consists of three ConvLSTM layers of $(v_1 = 9; f_1 = 64)$, $(v_2 = 1, f_2 = 32)$ and $(v_3 = 5, f_3 = 1)$ respectively is integrated after the ADD-TT interpolation in the TS-ChannelNet estimator. We note that, the SR-ConvLSTM network combines both the CNN and the LSTM networks [34], which increases the overall computational complexity as we discuss later. In contrast, the employed optimized SR-CNN and DN-CNN decreases significantly the complexity due to the accuracy of the proposed WI estimators. In conclusion, as the accuracy of the pre-estimation increases, the complexity of the employed CNN decreases, since low-complexity architectures can be used and vice versa.

VII. COMPUTATIONAL COMPLEXITY ANALYSIS

In this section, a detailed computational complexity, TDR, and latency analysis of the 2D LMMSE estimator, ChannelNet, TS-ChannelNet, and the proposed WI estimators are presented.

A. Computational Complexity Analysis

The computational complexity is expressed in terms of real valued multiplication/division and summation/subtraction mathematical operations required for a full OFDM frame channel estimation. Each complex-valued division requires 6 real valued multiplications, 2 real valued divisions, 2 real valued summations, and 1 real valued subtraction. On the other hand, each complex valued multiplication can be expressed by 4 real valued multiplications, and 3 real valued summations.

Table V: CNN-based estimators Overall computation complexity in terms of the total required real valued operations.

| Scheme | Interpolation | | CNN | |
|---------------|---|-------------------------------|-------------------|------------------|
| | Mul./Div. | Sum./Sub. | Mul./Div. | Sum./Sub. |
| ChannelNet | $K_p^2 I^2 (4 + K_d I) + K_p I (2 + 3K_d I)$ | $K_p I (5K_p I + 5K_d I - 2)$ | $350144K_{on} I$ | $42432K_{on} I$ |
| TS-ChannelNet | $24K_{on} I + 4LK_{on} I$ | $18K_{on} I + 5K_{on} I L$ | $226880K_{on} I$ | $81472K_{on} I$ |
| FP-SLS-SR-CNN | $2K_{on} P + 2K_{on} + 4K_{on} I_d$ | $2K_{on} + 2K_{on} I_d$ | $7008K_{on} I_d$ | $1120K_{on} I_d$ |
| FP-ALS-SR-CNN | $4K_{on}^2 P + 2K_{on} P + 2K_{on} + 4K_{on} I_d$ | $5K_{on}^2 P + 2K_{on} I_d$ | | |
| LP-SR-CNN | $2LP + 4K_{on} LP + 2K_{on} + 4K_{on} I_d$ | $5K_{on} LP + 2K_{on} I_d$ | | |
| FP-SLS-DN-CNN | $2K_{on} P + 2K_{on} + 4K_{on} I_d$ | $2K_{on} + 2K_{on} I_d$ | $84096K_{on} I_d$ | $9856K_{on} I_d$ |
| FP-ALS-DN-CNN | $4K_{on}^2 P + 2K_{on} P + 2K_{on} + 4K_{on} I_d$ | $5K_{on}^2 P + 2K_{on} I_d$ | | |
| LP-DN-CNN | $2LP + 4K_{on} LP + 2K_{on} + 4K_{on} I_d$ | $5K_{on} LP + 2K_{on} I_d$ | | |

1) *2D LMMSE estimator*: The conventional 2D LMMSE estimator requires first the LS estimation as that requires $2K_p I$ divisions. Then, the matrix inverse operation requires $4K_p^3 I^3$ multiplications and $3K_p^3 I^3$ summations. Finally, the correlation matrices are multiplied by the LS estimated channel vector resulting in $K_p^2 I^2 + K_d^2 K_p^2 I^4$ multiplications. Therefore, the overall computational complexity of the conventional 2D LMMSE estimator is $4K_p^3 I^3 + K_p^2 I^2 + K_d^2 K_p^2 I^4 + 2K_p I$ multiplications and $3K_p^3 I^3 + 2K_p I$ summations. We note that, in case the full $\mathbf{W}_{2D-LMMSE}$ matrix is calculated offline, the computational complexity of the 2D-LMMSE estimator is reduced to $4K_d K_p^2 I^2 + 2K_p I$ multiplications and $3K_d K_p^2 I^2 + 2K_d K_p I^2 - 2K_d I$ summations. We can notice that the 2D LMMSE suffers from very high computational complexity that make it impractical estimator in real-time scenarios.

2) *ChannelNet estimator*: employs the RBF interpolation followed by SR-CNN and DN-CNN networks. Thereby, the computational complexity of the ChannelNet is given by

$$CC_{\text{ChannelNet}} = CC_{\text{RBF}} + CC_{\text{SR-CNN}} + CC_{\text{DN-CNN}}. \quad (27)$$

As shown in (4), the calculation of $\hat{\mathbf{H}}_{LS}$ requires $2K_p I$ divisions. \mathbf{w}_{RBF} calculation requires $4K_p^2 I^2$ multiplications/divisions and $5K_p^2 I^2 - 2K_p I$ summations/subtractions. On the other hand, $\hat{\mathbf{H}}_{\text{RBF}}$ computation requires $K_d I (K_p^2 I^2 + 3K_p I)$ multiplications/divisions and $5K_d K_p I^2$ subtractions/summations. Therefore the total computational complexity of the RBF interpolation can be expressed by $K_p^2 I^2 (4 + K_d I) + K_p I (2 + 3K_d I)$ multiplications/divisions and $K_p I (5K_p I + 5K_d I - 2)$ summations/subtractions. After that, the ChannelNet estimator applies SR-CNN followed by DN-CNN on top of the RBF interpolation. $CC_{\text{SR-CNN}}$ and $CC_{\text{DN-CNN}}$ can be computed as follows

$$\begin{aligned} CC_{\text{SR-CNN}} &= \sum_{j=1}^J h_j w_j d_j v_j^2 f_j + h_j w_j d_j f_j \\ &= \sum_{j=1}^J h_j w_j d_j f_j (v_j^2 + 1). \end{aligned} \quad (28)$$

$$CC_{\text{DN-CNN}} = \sum_{j=1}^J h_j w_j d_j f_j (v_j^2 + 1) + \sum_{j=1}^D 4h_j w_j d_j, \quad (29)$$

where J denotes the number of CNN layers. We note that the second term in $CC_{\text{DN-CNN}}$ represents the num-

ber of operations required by the batch normalization considered in the DN-CNN network. Therefore, the SR-CNN used in the ChannelNet estimator requires $16064K_{on} I$ multiplications/divisions and $4288K_{on} I$ summations/subtractions, while the ChannelNet DN-CNN computations require $334080K_{on} I$ multiplications/divisions and $38144K_{on} I$ summations/subtractions.

3) *TS-ChannelNet estimator*: applies the ADD-TT interpolation followed by the SR-ConvLSTM network. Accordingly, the overall computational complexity of the TS-ChannelNet estimator can be expressed as follows

$$CC_{\text{TS-ChannelNet}} = CC_{\text{ADD-TT}} + CC_{\text{SR-ConvLSTM}}. \quad (30)$$

The ADD-TT interpolation applies two equalization steps (9), and (10) after the LS estimation in (8) that requires $2K_{on}$ summations, and $2K_{on}$ divisions. Each equalization step consists of K_{on} complex valued division, therefore the equalization in (9), and (10) requires $16K_{on}$ multiplications/divisions and $6K_{on}$ summations/subtractions. The time domain truncation operation applied in (13) requires $4LK_{on}$ multiplications and $5K_{on} L - 2K_{on}$ summations. After time domain truncation, the ADD-TT interpolation applies frequency and time domain averaging. The frequency domain averaging (14) requires $10K_{on}$ summations, and $2K_{on}$ multiplications. Moreover, the time domain averaging step (15), requires $4K_{on}$ real valued divisions, and $2K_{on}$ real valued summations. Therefore, the overall computational complexity of the ADD-TT interpolation for the whole received OFDM frame requires $24K_{on} I + 4LK_{on} I$ real valued multiplications/divisions, and $18K_{on} I + 5K_{on} I L$ real valued summations/subtractions, and its computational complexity is expressed in terms of the overall operations applied in the input, forget, and output gates of the SR-ConvLSTM network, such that

$$CC_{\text{ConvLSTM}} = \sum_{j=1}^J h_j w_j d_j f_j (8v_j^2 + 30). \quad (31)$$

Based on (31), the SR-ConvLSTM network utilized in the TS-ChannelNet estimator requires $226880K_{on} I$ multiplications/divisions and $81472K_{on} I$ summations/subtractions. TS-ChannelNet estimator is less complex than the ChannelNet estimator, since it employs only one CNN on top of the ADD-TT interpolation, unlike the ChannelNet estimator where both SR-CNN and DN-CNN are used.

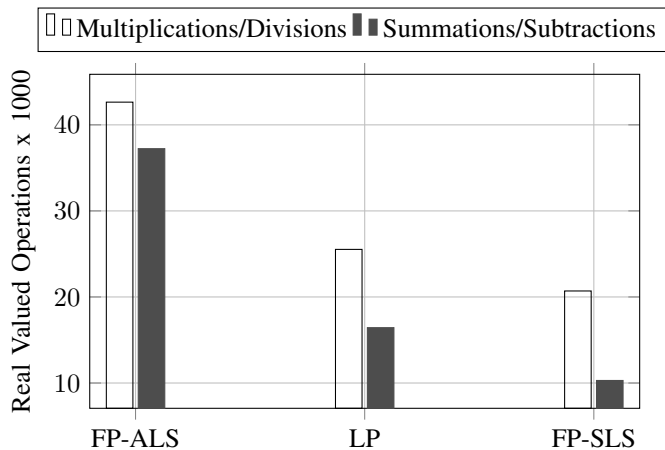


Fig. 9: Computational complexities of the proposed WI estimators employing $P = 2$.

4) *Proposed WI estimators*: WI estimators computational complexity depend mainly on the selected frame structure, the pilot allocation scheme, and the selected optimized CNN. The overall computational complexity of the proposed WI estimators can be expressed as follows

$$CC_{WI} = CC_{\hat{H}_{WI}} + CC_{O-CNN}. \quad (32)$$

In case of inserting full pilot symbols, there are two options, SLS estimator that requires only $2K_{on}P + 2K_{on}$ divisions, and $2K_{on}$ summations. The second option is employing ALS, where $2K_{on}P + 2K_{on}$ divisions, followed by $4K_{on}^2P$ multiplications, and $5K_{on}^2P$ summations are required. On the other hand, when $K_p = L$ pilots are inserted with each pilot symbol, then the LS estimation requires $2LP + 2K_{on}$ divisions, $4K_{on}LP$ multiplications, and $5K_{on}LP$ summations. After selecting the required frame structure and pilot allocation scheme, the proposed estimators apply the weighted interpolation as shown in (22), where the channel estimation for each received OFDM frame requires $4K_{on}I_d$ divisions and $2K_{on}I_d$ summations. Finally, the optimized SR-CNN is utilized in low mobility scenario and it requires $7008K_{on}I_d$ multiplications/divisions and $1120K_{on}I_d$ summations/subtractions, while the optimized DN-CNN is exploited in high and very high mobility scenarios requires $84096K_{on}I_d$ multiplications/divisions and $9856K_{on}I_d$ summations/subtractions

The proposed WI-FP-ALS records the higher computational complexity among the other proposed estimators in all mobility scenarios, due to the \mathbf{W}_{ALS} calculation in (18). Moreover, the proposed WI-FP-SLS estimator is the simplest one. Fig. 9 shows the computational complexity of the proposed WI estimators employing $P = 2$ pilot symbols within the frame.

Table V shows the overall computational complexity of the benchmarked estimators in terms of real valued operations. It is worth mentioning that the proposed WI estimators achieve a significant computational complexities decrease compared to ChannelNet and TS-ChannelNet estimators, where ChannelNet and TS-ChannelNet estimators are 70 and 39 times more complex than the proposed FP-ALS-SR-CNN estimator respectively. Moreover, the proposed estima-

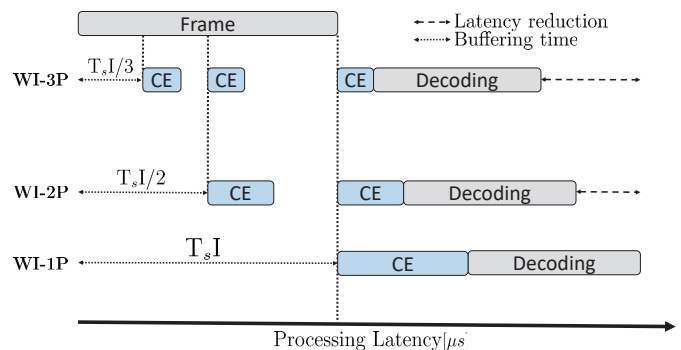


Fig. 10: Processing latency of the proposed WI estimators.

tors achieve at least 7027.35 times less complexity than the 2D LMMSE estimator, with an acceptable BER performance, thus making them a good alternative to the 2D LMMSE. We note that FP-ALS-DN-CNN is 12 times more complex than FP-ALS-SR-CNN since the optimized DN-CNN architecture complexity that is employed in high and very high scenarios is higher than the optimized SR-CNN architecture which is used in low mobility scenario.

B. Transmission data rate and latency analysis

The TDR and the latency introduced at the receiver in order to recover the transmitted bits are important issues in vehicular communications, especially for real time applications [35]. The transmission data rate is influenced by the number of allocated data subcarriers within the transmitted frame:

$$TDR = \frac{K_{DF} \log_2(M) \rho}{T_s I}, \quad (33)$$

where K_{DF} and ρ denote the total allocated data subcarriers within the transmitted frame, and the employed code rate respectively, and M represents the utilized modulation order. Moreover, the buffering time φ can be expressed by the total duration that the receiver should wait before starting the channel estimation, such that $\varphi = T_s I$, where T_s represents the received OFDM data symbol duration. ChannelNet and TS-ChannelNet estimators employ $K_p = 4$ pilot subcarriers within each transmitted OFDM symbol, however ChannelNet estimator requires sparsely pilot allocation within the transmitted OFDM frame, and thereby, increasing the complexity of pilots extraction at the receiver, since the allocated pilots subcarriers indices differs between the OFDM symbols within the frame. Therefore, ChannelNet and TS-ChannelNet estimators have similar transmission data rate as defined in the IEEE 802.11p standard. However, they suffer from high buffering time at the receiver, since the full frame should be received before the channel estimation starts leading to high latency.

As shown in Table VI, the proposed WI estimators record different TDRs gains according to the selected frame structure. Moreover, the proposed WI-2P and WI-3P estimators require lower buffering time than the proposed WI-1P, ChannelNet, and TS-ChannelNet estimators, since it divides the frame into several sub frames as shown in Fig. 10. Hence, the channel estimation process starts earlier. Therefore, the proposed WI estimators contribute in reducing the total required latency.

Table VI: Transmission data rate and buffering time analysis for the benchmarked schemes.

| Scheme | ChannelNet | TS-ChannelNet | Proposed 2D WI | | | | | |
|----------------------|------------|---------------|----------------|-------|-------|-------|-------|-------|
| | | | WI-1P | | WI-2P | | WI-3P | |
| | | | FP | LP | FP | LP | FP | LP |
| TDR gain | 0 % | 0% | 7.25% | 8.08% | 6.16% | 7.83% | 5.08% | 7.58% |
| φ [μ s] | 800 | 800 | 800 | | 400 | | 265 | |

Finally, we note that the transmission parameters and the chosen frame structure should be adapted according to the mobility condition, required data rate, and the acceptable latency by the vehicular application.

VIII. CONCLUSION

In this paper, FBF channel estimation in vehicular communication is studied, where the limitations of the conventional 2D LMMSE estimator and the motivation behind employing CNN processing in the channel estimation are presented. Moreover, The recently proposed CNN-based channel estimators have been extensively surveyed. In this context, we have proposed a hybrid, adaptive, and robust WI channel estimators for the IEEE 802.11p standard, where pilot symbols are inserted within the transmitted frame, with several pilot allocation schemes adapted according to the mobility condition. Unlike the recently proposed ChannelNet and TS-ChannelNet estimators that suffer from high computational complexity, performance degradation in high mobility vehicular scenarios, and high latency at the receiver, the proposed WI estimators have reduced computational complexity and robustness in high mobility scenarios. Moreover, they require low buffering time at the receiver and more TDR gain is achieved since all the OFDM symbols within the transmitted frame are fully allocated to data. Additionally, the employed SR-CNN and DN-CNN architectures are optimized through intensive experiments in order to alleviate the high complexity problem. Simulation results have shown the performance superiority of the proposed WI estimators over ChannelNet and TS-ChannelNet estimators in all vehicular scenarios with a substantial reduction in computational complexity, where ChannelNet and TS-ChannelNet are more complex than the proposed WI-FP-ALS-SR-CNN by 70 and 39 times respectively. On the other hand, the proposed estimators are less complex than the conventional 2D LMMSE estimator by at least 7027.35 times while recording a convenient BER performance especially in high mobility vehicular scenarios, which makes them good alternatives to the conventional 2D LMMSE estimator.

REFERENCES

- [1] C. F. Mecklenbrauker, A. F. Molisch, J. Karedal, F. Tufvesson, A. Paier, L. Bernado, T. Zemen, O. Klemp, and N. Czink, "Vehicular Channel Characterization and Its Implications for Wireless System Design and Performance," *Proceedings of the IEEE*, pp. 1189–1212, 2011.
- [2] R. Bomfin, M. Chafii, A. Nimr, and G. Fettweis, "A Robust Baseband Transceiver Design for Doubly-Dispersive Channels," *IEEE Transactions on Wireless Communications*, 2021.
- [3] A. Abdelgader and L. Wu, "The Physical Layer of the IEEE 802.11 p WAVE Communication Standard: The Specifications and Challenges," in *The Physical Layer of the IEEE 802.11 p WAVE Communication Standard: The Specifications and Challenges*, vol. 2, 10 2014.
- [4] J. A. Fernandez, K. Borries, L. Cheng, B. V. K. Vijaya Kumar, D. D. Stancil, and F. Bai, "Performance of the 802.11p Physical Layer in Vehicle-to-Vehicle Environments," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 3–14, 2012.
- [5] Z. Zhao, X. Cheng, M. Wen, B. Jiao, and C. Wang, "Channel Estimation Schemes for IEEE 802.11p Standard," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 4, pp. 38–49, 2013.
- [6] Yoon-Kyeong Kim, Jang-Mi Oh, Yoo-Ho Shin, and Cheol Mun, "Time and Frequency Domain Channel Estimation Scheme for IEEE 802.11p," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 1085–1090.
- [7] S. Ehsanfar, M. Chafii, and G. P. Fettweis, "On UW-based Transmission for MIMO Multi-carriers with Spatial Multiplexing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, pp. 5875–5890, 2020.
- [8] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep Learning for Wireless Physical Layer: Opportunities and Challenges," 2017.
- [9] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.
- [10] Y. Yang, F. Gao, X. Ma, and S. Zhang, "Deep Learning-Based Channel Estimation for Doubly Selective Fading Channels," *IEEE Access*, vol. 7, pp. 36 579–36 589, 2019.
- [11] X. Ma, H. Ye, and Y. Li, "Learning Assisted Estimation for Time-Varying Channels," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, 2018, pp. 1–5.
- [12] H. Ye, G. Y. Li, and B. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.
- [13] S. Han, Y. Oh, and C. Song, "A Deep Learning Based Channel Estimation Scheme for IEEE 802.11p Systems," in *IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.
- [14] A. K. Gizzini, M. Chafii, A. Nimr, and G. Fettweis, "Deep Learning Based Channel Estimation Schemes for IEEE 802.11p Standard," *IEEE Access*, vol. 8, pp. 113 751–113 765, 2020.
- [15] A. K. Gizzini, M. Chafii, A. Nimr, and G. Fettweis, "Joint TRFI and Deep Learning for Vehicular Channel Estimation," in *IEEE GLOBE-COM 2020*, Taipei, Taiwan, Dec. 2020.
- [16] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep Learning-Based Channel Estimation," *IEEE Communications Letters*, vol. 23, no. 4, pp. 652–655, 2019.
- [17] X. Zhu, Z. Sheng, Y. Fang, and D. Guo, "A Deep Learning-aided Temporal Spectral ChannelNet for IEEE 802.11p-based Channel Estimation in Vehicular Communications," *EURASIP Journal on Wireless Communications and Networking*, vol. 1, no. 94, 2020.
- [18] S. I. Kim, H. S. Oh, and H. K. Choi, "Mid-Ambled Aided OFDM Performance Analysis in High Mobility Vehicular Channel," in *2008 IEEE Intelligent Vehicles Symposium*, 2008, pp. 751–754.
- [19] W. Cho, S. I. Kim, H. k. Choi, H. S. Oh, and D. Y. Kwak, "Performance Evaluation of V2V/V2I Communications: The Effect of Midamble Insertion," in *2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology*, 2009, pp. 793–797.
- [20] X. Weidong, G. Javier, N. Zhisheng, A. Onur, and E. Eylem, "Wireless Access in Vehicular Environments," *EURASIP Journal on Wireless Communications and Networking*, vol. 1, pp. 1687–1499, 2009.
- [21] W. Toit, "Radial Basis Function Interpolation," *Master Thesis dissertation, Department of Mathematical Sciences University of Stellenbosch, Matieland, South Africa*, 2008.
- [22] C. Chatfield, *Time-Series Forecasting*. Chapman and Hall/CRC, 2000.
- [23] Y. R. Zheng and C. Xiao, "Channel Estimation for Frequency-Domain Equalization of Single-Carrier Broadband Wireless Communications," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 815–823, 2009.

- [24] 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.
- [25] C. Dong, C. C. Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.
- [26] F. Pontes, G. Amorim, P. Balestrassi, A. Paiva, and J. Ferreira, "Design of Experiments and Focused Grid Search for Neural Network Parameter Optimization," *Neurocomputing*, vol. 186, pp. 22 – 34, 2016.
- [27] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, p. 85–117, Jan 2015.
- [28] S. ichi Amari, "Backpropagation and Stochastic Gradient Descent Method," *Neurocomputing*, vol. 5, no. 4, pp. 185 – 196, 1993.
- [29] S. De, A. Mukherjee, and E. Ullah, "Convergence Guarantees for RMSProp and ADAM in Non-Convex Optimization and an Empirical Comparison to Nesterov Acceleration," 2018.
- [30] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015.
- [32] A. K. Gizzini, M. Chafii, A. Nimr, and G. Fettweis, "Enhancing Least Square Channel Estimation Using Deep Learning," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.
- [33] G. Acosta-Marum and M. A. Ingram, "Six Time and Frequency Selective Empirical Channel Models for Vehicular Wireless LANs," *IEEE Vehicular Technology Magazine*, vol. 2, no. 4, pp. 4–11, 2007.
- [34] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. kin Wong, and W. chun Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," 2015.
- [35] M. I. Ashraf, Chen-Feng Liu, M. Bennis, and W. Saad, "Towards Low-Latency and Ultra-Reliable Vehicle-to-Vehicle Communication," in *2017 European Conference on Networks and Communications (EuCNC)*, 2017, pp. 1–5.



Abdul Karim Gizzini (Student Member, IEEE) received the bachelor degree in computer and communication engineering (B.E) from the IUL university of Lebanon, in 2015 followed by the M.E degree in 2017. His master thesis was hosted by the lebanese national council for scientific research (CNRS). Since 2019 he is pursuing the Ph.D. degree in wireless communication engineering in ETIS laboratory which is a joint research unit at CNRS (UMR 8051), ENSEA, and CY Cergy Paris University in France. His Ph.D. thesis focuses mainly on deep

learning based channel estimation in high mobility vehicular scenarios.



Marwa Chafii (Member, IEEE) received her Ph.D. degree in electrical engineering in 2016 from CentraleSupélec, France. She joined the TU Dresden, Germany, in 2018 as a group leader, and ENSEA, France, in 2019 as an associate professor where she held a Chair of Excellence on Artificial Intelligence from CY Initiative. Since September 2021, she is an associate professor at New York University (NYU) Abu Dhabi, and a global network associate professor at NYU WIRELESS, Tandon School of Engineering, NYU. She received the prize of the best Ph.D. in

France in the fields of Signal, Image & Vision, and she has been nominated in the top 10 Rising Stars in Computer Networking and Communications by N2Women in 2020. She served as Associate Editor at IEEE Communications Letters 2019-2021, where she received the Best Editor Award in 2020. She is currently vice-chair of the IEEE ComSoc ETI on Machine Learning for Communications and leading the Education working group of the ETI on Integrated Sensing and Communications.



Ahmad Nimr (Member, IEEE) received the Ph.D. degree in electrical engineering from TU Dresden, Germany in 2021, M.Sc degree from TU Ilmenau, Germany in 2014, and the Diploma from HIAST, Syria in 2004. From 2005 to 2011, he pursued industrial carrier in software and hardware development. Since 2020 he has been a research group leader at Vodafone Chair Mobile Communications, TU Dresden. He has worked on several EU and German funded projects with publications in journals and conferences proceedings. His current research

focuses on signal processing for communications and sensing from theory to implementation.



Raed M. Shubair (Senior Member, IEEE) received his Ph.D. degree in electrical engineering from the University of Waterloo, Canada, in 1993. He is a Full Professor affiliated to New York University (NYU) Abu Dhabi. He is recipient of the Distinguished Service Award from ACES Society and from MIT Electromagnetics Academy. He is a Board Member of the European School of Antennas, Regional Director for the IEEE Signal Processing Society in Middle East and served as the founding chair of the IEEE Antennas and Propagation Society Educational

Initiatives Program. He is a Fellow of MIT Electromagnetics Academy and a Founding Member of MIT Scholars of the Emirates. He is Editor for the IEEE Journal of Electromagnetics, RF, and Microwaves in Medicine and Biology, and Editor for the IEEE Open Journal of Antennas and Propagation. He is a Founding Member of five IEEE society chapters in UAE. He is the Founder and Chair of IEEE at NYUAD. He is an officer for IEEE ComSoc ETI Machine Learning for Communications. He is the founding director of IEEE UAE Distinguished Seminar Series Program.



Gerhard Fettweis (Fellow, IEEE) is Vodafone Chair Professor at TU Dresden since 1994, and heads the Barkhausen Institute since 2018, respectively. He earned his Ph.D. from RWTH Aachen in 1990. After one year at IBM Research in San Jose, CA, he moved to TCSI Inc., Berkeley, CA. He coordinates the 5G Lab Germany, and 2 German Science Foundation (DFG) centers at TU Dresden, namely cfaed and HAEC. His research focusses on wireless transmission and chip design for wireless/IoT platforms, with 20 companies from Asia/Europe/US sponsoring

his research. Gerhard is IEEE Fellow, member of the German Academy of Sciences (Leopoldina), the German Academy of Engineering (acatech), and received multiple IEEE recognitions as well as the VDE ring of honor. In Dresden his team has spun-out sixteen start-ups, and setup funded projects in volume of close to EUR 1/2 billion. He co-chairs the IEEE 5G Initiative, and has helped organizing IEEE conferences, most notably as TPC Chair of ICC 2009 and of TTM 2012, and as General Chair of VTC Spring 2013 and DATE 2014.