IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 3, 2020, accepted June 11, 2020, date of publication June 18, 2020, date of current version June 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003286

Deep Learning Based Channel Estimation Schemes for IEEE 802.11p Standard

ABDUL KARIM GIZZINI[1], (Graduate Student Member, IEEE), MARWA CHAFII[1], (Member, IEEE), AHMAD NIMR[2], (Member, IEEE), AND GERHARD FETTWEIS[2], (Fellow, IEEE)

[1]ETIS, National Council for Scientific Research (UMR8051), ENSEA, CY Cergy Paris Université, 95000 Cergy, France
[2]Vodafone Chair Mobile Communication Systems, Technische Universitat Dresden, 01069 Dresden, Germany

Corresponding author: Abdul Karim Gizzini (abdulkarim.gizzini@ensea.fr)

This work was supported by CNRS (France).

ABSTRACT IEEE 802.11p standard is specially developed to define vehicular communications requirements and support cooperative intelligent transport systems. In such environment, reliable channel estimation is considered as a major critical challenge for ensuring the system performance due to the extremely time-varying characteristic of vehicular channels. The channel estimation of IEEE 802.11p is preamble based, which becomes inaccurate in high mobility scenarios. The major challenge is to track the channel variations over the course of packet length while adhering to the standard specifications. The motivation behind this paper is to overcome this issue by proposing a novel deep learning based channel estimation scheme for IEEE 802.11p that optimizes the use of deep neural networks (DNN) to accurately learn the statistics of the spectral temporal averaging (STA) channel estimates and to track their changes over time. Simulation results demonstrate that the proposed channel estimation scheme STA-DNN significantly outperforms classical channel estimators in terms of bit error rate. The proposed STA-DNN architectures also achieve better estimation performance than the recently proposed auto-encoder DNN based channel estimation with at least 55.74% of computational complexity decrease.

INDEX TERMS Channel estimation, deep learning, DNN, IEEE 802.11p standard, vehicular channels.

I. INTRODUCTION

The cooperative intelligent transportation system (C-ITS) has been developed to provide various traffic services such as road safety, route planning and congestion avoidance. To realize these applications, the IEEE 802.11p standard was developed to define the physical layer of the wireless access in vehicular environments [1]. In general, wireless access in vehicular environments contains mainly two distinct types of networking which are vehicle-to-vehicle (VTV) and roadside-to-vehicle (RTV). Ensuring communication reliability between different vehicular network units is very important for IEEE 802.11p critical applications.

Communication reliability can be ensured by precisely estimating the wireless channel response in vehicular environments. The design of channel estimation technique in such environment is challenging because of its double selectivity nature. In fact, the IEEE 802.11p is originally derived from the well-known standard IEEE 802.11a, which was initially designed for relatively stationary indoor environments, without considering the impact of high mobility. Indeed, at higher velocities, the wireless channel varies fast, so that its impulse response changes within a frame. This is because in vehicular environments both the transmitter and the receiver are in motion, and there are both mobile and stationary scatterers leading to multi-path channel with a short coherence time. The IEEE 802.11p standard employs orthogonal frequency-division multiplexing (OFDM) transmission scheme. The transmitted frame is composed of a preamble, signal, and a data field containing the useful transmitted data. The preamble field marks the beginning of the transmitted frame, and it is used for channel estimation. At the receiver, the channel is then estimated once for each frame using the predefined transmitted preamble. Then, the estimate is used to equalize all the received symbols within the frame. As a result of the time-varying nature of the channel, the estimate obtained at

The associate editor coordinating the review of this manuscript and approving it for publication was Razi Iqbal.

the beginning of the frame can quickly become outdated, resulting in poor performance. To track the channel changes, four pilot subcarriers are used within each OFDM symbol. Channel estimates are updated utilizing these pilots besides the channel estimate of previous OFDM symbol. Due to fast channel variations in vehicular communications, four pilots are not sufficient to accurately track these variations. Thus, proper channel estimation schemes are required, either by increasing the number of pilots within the transmitted OFDM symbols which reduces the throughput, or by adhering to the standard specifications employing the transmitted preamble with further operations. Therefore, the primary challenge is to design a more accurate scheme of estimating and tracking the channel estimate over the course of a frame length. Improving channel estimation accuracy decreases packet errors, and thus increases the reliability of vehicular communications.

Several classical channel estimation schemes have been proposed in the literature for IEEE 802.11p standard. The STA scheme [2] that uses both time and frequency correlation of the successive received OFDM symbols to update the channel estimates for the current symbol, achieves better performance than other classical channel estimation schemes at low signal-to-noise ratio (SNR) region but significant error floor occurs at high SNRs. Moreover, STA channel estimation parameters that determine the time and frequency correlation ratio between two successive received OFDM symbols requires a prior knowledge about the channel to be accurately estimated. However, it is hard to obtain such information in real case scenarios. The authors in [3] have proposed constructed data pilots (CDP) scheme to improve the accuracy of the channel estimates by exploiting the correlation characteristics between each two adjacent symbols through utilizing data subcarriers as pilots, such that the data subcarriers from previous OFDM symbol are used as preamble to estimate the channel for the current symbol. The CDP scheme is effective in estimating the channels. It outperforms STA scheme especially in high SNR regions, but it still can not provide acceptable error performance when higher modulation orders are employed. Inspired by the work presented in [3], the authors in [4] have proposed a time domain reliable test frequency domain interpolation (TRFI) scheme. Reliability test is the key element in this scheme, it is performed by equalizing the previous received OFDM symbol by the previous and current estimated channel respectively. Identical equalization results reveal that the channel estimates are reliable, these reliable channel estimates are then used to interpolate the channel estimates at the unreliable data subcarriers, i.e the equalization results are different. The minimum mean square error using virtual pilots (MMSE-VP) scheme proposed in [5], uses virtual pilot subcarriers that are constructed from the received data subcarriers to obtain the channel correlation matrices, and then frequency domain minimum mean squared error (MMSE) is applied to estimate the channel. MMSE-VP achieves worse performance than the STA in low SNR regions.

In all classical channel estimation schemes, the previous OFDM symbol, which is generated from estimated data symbols is used as a preamble of the current OFDM symbol. This process is influenced by the demapping error, which depends on the accuracy of previous estimation and the noise level. This error propagates and increases from one symbol to another all over the frame causing a considerable reliability degradation in realistic vehicular environments. A detailed discussion of the advantages and disadvantages of IEEE802.11p classical channel estimation schemes is presented in [3].

Recently, deep learning (DL) has drawn attentions for its great success in computer vision, automatic speech recognition, and natural language processing [6]. There are two main reasons promoting the applications of DL in various areas [7]. First, DL-based algorithms are more accurate in capturing imperfections of real-world systems. Second, DL-based algorithms have low computational complexity, as they involve layers of simple operations such as matrix-vector multiplications. Motivated by these advantages, DL has been introduced to physical layer in wireless communications and achieved superior performance in various applications [8]–[10]. A DL-based channel estimation scheme for IEEE 802.11p based on auto-encoder deep neural network (AE-DNN) is proposed in [11]. The key ingredient of AE-DNN scheme is to combine DL and data-pilot aided channel estimation schemes such that both are mutually beneficial. First, the AE-DNN performs initial channel estimation, where the previous received OFDM symbols are used to estimate the channel for the current one. Then, the AE-DNN is trained offline in order to learn the ability of correcting the initial channel estimation errors, besides learning the channel characteristics in the frequency domain. Then, the trained AE-DNN is used for updating the channel estimates for each received OFDM symbol. AE-DNN is capable of recovering possible estimation errors, but this ability becomes limited when higher modulation orders are employed. Moreover, AE-DNN suffers from considerable complexity and does not consider the time-frequency correlation of the channel at successive OFDM symbols. The authors in [12] propose channel estimation approach using DNN, which works in three phases: (i) pre-training, to acquire a desirable DNN weights initialization, (ii) training phase, where DNN is trained offline on known data to learn how to estimate the channel, and (iii) testing online phase, where DNN is tested over unknown data. The proposed scheme requires several inputs to the DNN, and suffers from considerable complexity while using huge number of neurons within the hidden layers.

In order to improve the aforementioned schemes in terms of higher accuracy and lower complexity, we propose in this work a hybrid approach. In this approach, an initial coarse estimation is obtained using STA, and then a fine estimation is achieved by means of DNN. The STA estimation considers the time and frequency correlation of the channel at the successive OFDM symbols. Applying additional DNN processing on top of STA allows improved learning of the correlation

and leads to a better tracking of channel changes. Moreover, the proposed STA-DNN architectures are optimized to reduce the computational complexity achieving much lower complexity compared with AE-DNN. The numerical results reveal that our STA-DNN schemes outperform the classical channel estimators as well as AE-DNN, especially in high mobility scenarios employing high modulation orders.

The remainder of this paper is organized as follows: in Section II, a description of the IEEE 802.11p standard and the system model is provided. Section III illustrates IEEE 802.11p classical channel estimations schemes. The proposed STA-DNN schemes are presented in Section IV. In Section V, the performance of the proposed schemes is evaluated through simulation results carried out using different vehicular channel models and modulation orders. A detailed computational complexity analysis is provided in Section VI. Finally, the paper is concluded in Section VII.

II. SYSTEM DESCRIPTION

In this section, the IEEE 802.11p standard specifications, frame structure, and transmitter-receiver design structure are first presented. Then, the system model considered in this paper is described.

A. IEEE 802.11p STANDARD SPECIFICATIONS

IEEE 802.11p is one of the recent approved amendments to the IEEE 802.11 standard to add wireless access in vehicular environments. It is based on IEEE 802.11a standard with some enhancements required to support C-ITS applications. This includes data exchange between high speed vehicles and between the vehicles and the roadside infrastructure.

In IEEE 802.11p, a 10MHz frequency bandwidth is used, instead of 20MHz bandwidth in IEEE 802.11a, and thus, all parameters in the time domain for IEEE 802.11p are doubled compared with the IEEE 802.11a. The doubled guard interval reduces the inter-symbol-interference (ISI) more than the guard interval in IEEE 802.11a. IEEE 802.11p also employs OFDM with 64 subcarriers, as shown in Fig. 1, only 52 of them are active in the range from -26 to 26 except the 0-th subcarrier which is null. Pilots are embedded into the subcarriers $\{-21, -7, 7, 21\}$ for channel tracking. The other active subcarriers are used for data.

IEEE 802.11p standard offers data exchange among vehicles VTV and between vehicles and roadside infrastructure RTV within a range of 1 km using a transmission rate of 3 Mbps to 27 Mbps with different modulation and coding schemes, and it supports high vehicles velocities. Table 1 illustrates the IEEE 802.11p main physical layer parameters.

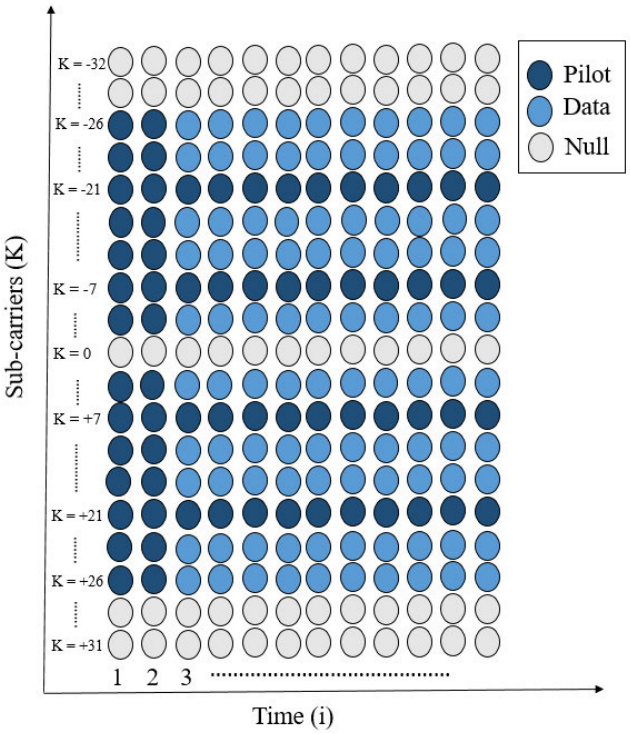


FIGURE 1. IEEE 802.11p subcarriers arrangement.

TABLE 1. IEEE 802.11p Physical layer specifications.

Parameter	Values
Bandwidth	10 MHz
FFT size	64
FFT Period	6.4 us
Guard interval duration	1.6 us
Symbol duration	8 us
Total subcarriers	64
Pilot subcarriers	4
Data subcarriers	48
Null subcarriers	12
Subcarrier spacing	156.25 KHz
Code rate	1/2, 2/3, 3/4
Modulation schemes	BPSK, QPSK, 16QAM, 64QAM
Data rate	3, 4.5, 6, 9, 12, 18, 24, 27 Mbps

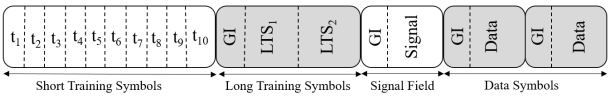


FIGURE 2. IEEE 802.11p transmitted frame structure.

1) IEEE 802.11p FRAME

As illustrated in Fig. 2, the frame starts with a preamble field that includes: (i) *short training sequence*: consists of ten short sequences each of duration 1.6 μs used by the receiver for signal detection, and time synchronization, and (ii) *long training symbols*: used for channel estimation, the guard interval GI2 is used for the long training sequence, whereas GI is used as cyclic prefix (CP) guard interval per OFDM symbol. These guard intervals are employed to reduce the ISI. After the preamble, the *signal field* consists of one CP-OFDM symbol that carries the physical layer convergence protocol, which provides information about the payload. Finally, the *data field* contains a sequence of OFDM symbols corresponding to the data payload.

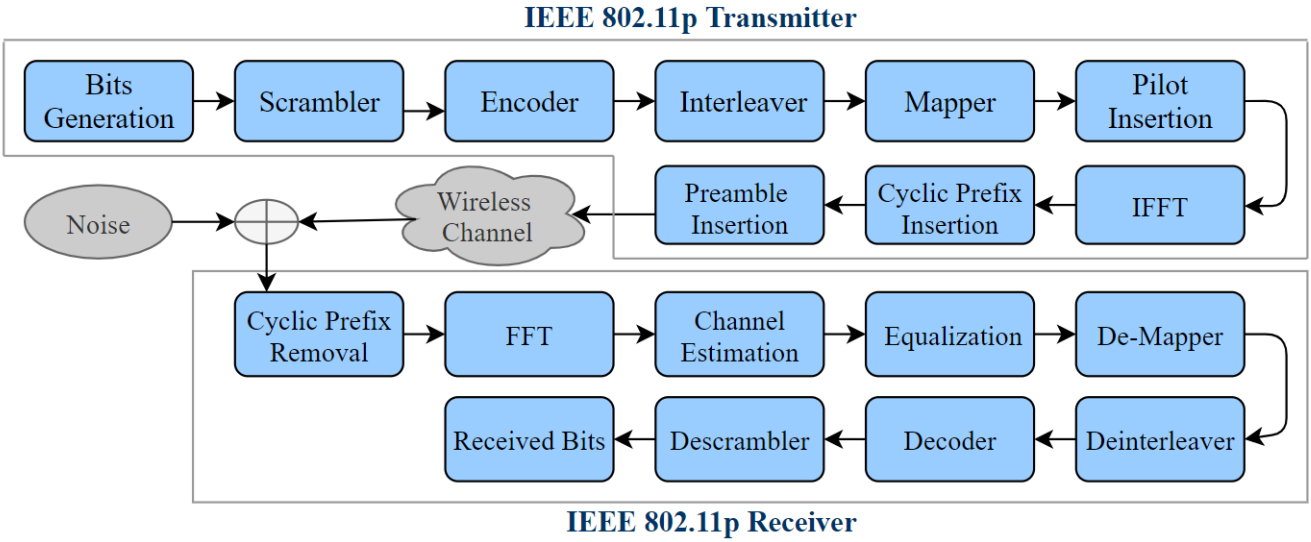


FIGURE 3. IEEE802.11p transmitter-receiver block diagram.

2) IEEE 802.11p TRANSCEIVER

As shown in Fig. 3, the first operation on the transmitter side is the binary bits generation. Generated bits are scrambled in order to randomize the bits pattern, which may contain long streams of 1s or 0s. Bits scrambling facilitates the work of a timing recovery circuit and eliminates the dependence of signal's power spectrum upon the actual transmitted data. The scrambled bits are then passed to a convolutional encoder, which introduces some redundancy into the bits stream. This redundancy is used for error correction that allows the receiver to combat the effects of the channel, hence reliable communications can be achieved.

Bits interleaving is used to cope with the channel noise such as burst errors or fading. The interleaver rearranges input bits such that consecutive bits are split among different blocks. This can be done using a permutation process that ensures that adjacent bits are modulated onto non-adjacent subcarriers and thus allows better error correction at the receiver.

After that, the interleaved bits are mapped according to the used modulation technique. IEEE 802.11p standard defines four modulation techniques: BPSK, QPSK, 16QAM and 64QAM. Moreover Gray code is used, where the mapped symbol corresponding to neighboring symbols differs by exactly one bit. Bits mapping operation is followed by constructing the OFDM symbols to be transmitted. The data symbols and pilots are mapped to the active subcarriers and passed to the IFFT block to generate the time-domain OFDM symbols and followed by inserting the CP. Finally, the IEEE 802.11p packet is formed by concatenating the constructed CP-OFDM symbols, and the predefined preamble symbols in one frame.

At the receiver side, the preamble is used for synchronization and initial channel estimation. The CP is removed followed by FFT. The pilot subcarriers are used for channel tracking, and the data symbols are forwarded to the equalizer. The equalized data are de-mapped to obtain the encoded bits. Afterwards, deinterleaving, decoding and descrambling are performed to obtain the detected bits.

B. CHANNEL ESTIMATION SIGNAL MODEL

Assuming perfect synchronization, and ignoring the signal field, we focus on a frame that consists of two long preambles at the beginning followed by I OFDM data symbols, as shown in Fig. 1. Let $\mathcal{K}_{\text{on}}$ be the set of $K_{\text{on}} = |\mathcal{K}_{\text{on}}|$ active subcarriers, the input-output relation between the transmitted and the received OFDM frame of size $(K_{\text{on}} \times I)$ can be expressed as follows:

$$\boldsymbol{Y}[k,i] = \tilde{\boldsymbol{H}}[k,i]\boldsymbol{X}[k,i] + \boldsymbol{N}[k,i], \quad k \in \mathcal{K}_{\text{on}}, \tag{1}$$

where $\boldsymbol{X}[k,i]$, $\boldsymbol{Y}[k,i]$, and $\boldsymbol{N}[k,i]$ denote the transmitted OFDM symbol, the received OFDM symbol, and the noise of the k-th sub-carrier in the i-th OFDM symbol, respectively. Here, $\tilde{\boldsymbol{H}}$ represents the time variant frequency response of the channel for all subcarriers within the transmitted OFDM frame. For simplicity, the transmitted data symbols and the transmitted preambles can be expressed in vector form as shown in (2), and (3) respectively.

$$\boldsymbol{y}_i[k] = \tilde{\boldsymbol{h}}_i[k]\boldsymbol{x}_i[k] + \boldsymbol{n}_i[k], \quad k \in \mathcal{K}_{\text{on}} \tag{2}$$

$$\boldsymbol{y}_i^{(p)}[k] = \tilde{\boldsymbol{h}}_i[k]\boldsymbol{p}[k] + \boldsymbol{n}_i^{(p)}[k], \quad k \in \mathcal{K}_{\text{on}}, \tag{3}$$

where $\boldsymbol{x}_i[k]$ and $\boldsymbol{p}[k]$ denote the i-th transmitted OFDM data symbol, and the transmitted preamble symbol respectively.

III. CLASSICAL CHANNEL ESTIMATION SCHEMES

The allocation of pilot subcarriers in the OFDM time-frequency grid is of crucial importance for channel estimation. As mentioned before, IEEE 802.11p allocates four pilots within each OFDM symbol. These pilots are not sufficient

for channel variations tracking, therefore proper channel estimation cannot be guaranteed employing only these pilots. To tackle this problem, two categories of channel estimation schemes have been proposed. The first category includes channel estimation methods that adhere to the IEEE 802.11p standard, where data subcarriers are used in addition to pilots to estimate the channel. While in the second category, channel estimation schemes perform some modifications on the IEEE 802.11p standard by inserting additional pilots within the transmitted frame and thus reducing the transmission rate. In this section, we present a detailed description of IEEE 802.11p classical channel estimation techniques that belongs to the first category. All the presented approaches will be compared with our proposed scheme.

A. LS ESTIMATION SCHEME

The least square (LS) channel estimation is a basic solution used in IEEE 802.11p. It employs two received long preambles denoted as $\boldsymbol{y}_1^{(p)}[k]$ and $\boldsymbol{y}_2^{(p)}[k]$ to estimate the channel gain as follows:

$$\hat{\hat{\boldsymbol{h}}}_{\mathrm{LS}}[k] = \frac{\boldsymbol{y}_1^{(p)}[k] + \boldsymbol{y}_2^{(p)}[k]}{2\boldsymbol{p}[k]}, \tag{4}$$

where $\boldsymbol{p}[k]$ is the predefined frequency domain symbol transmitted on the k-th subcarrier of the long training preamble, $\hat{\hat{\boldsymbol{h}}}_{\mathrm{LS}}[k]$ is the estimated channel at the k-th subcarrier. This estimate ignores the time variations of the channel and its accuracy decreases with the increase of the index i of the OFDM symbol. Therefore, employing the LS estimation without any channel tracking will significantly degrade the performance under mobility conditions.

B. INITIAL CHANNEL ESTIMATION SCHEME

Channel estimation in vehicular communications assumes high correlation between successive received OFDM symbols. First, LS channel estimation is performed. Then, an initial channel estimation is applied to update the channel estimates over all the received OFDM symbols expressed below:

1) The i-th OFDM symbol is equalized by the previously estimated channel as follows:

$$\boldsymbol{y}_{\mathrm{eq}_i}[k] = \frac{\boldsymbol{y}_i[k]}{\hat{\hat{\boldsymbol{h}}}_{\mathrm{Initial}_{i-1}}[k]}, \quad \hat{\hat{\boldsymbol{h}}}_{\mathrm{Initial}_0}[k] = \hat{\hat{\boldsymbol{h}}}_{\mathrm{LS}}[k]. \tag{5}$$

2) The data subcarrier $\boldsymbol{y}_{\mathrm{eq}_i}[k]$ is then demapped to the nearest constellation point to obtain $\boldsymbol{d}_i[k]$, whereas the pilot subcarriers are set to the predefined symbols in the standard.
3) The initial channel estimate is calculated:

$$\hat{\hat{\boldsymbol{h}}}_{\mathrm{Initial}_i}[k] = \frac{\boldsymbol{y}_i[k]}{\boldsymbol{d}_i[k]}. \tag{6}$$

$\hat{\hat{\boldsymbol{h}}}_{\mathrm{Initial}_i}[k]$ is considered as the key element of all IEEE 802.11p classical channel estimation schemes.

C. STA ESTIMATION SCHEME

The STA scheme [2] has been proposed as an enhanced equalization scheme used to tackle the problem of dynamic channel variations in vehicular communications. STA continuously updates the channel estimate by using data symbol decisions to estimate the channel at data subcarriers taking into consideration the time and frequency correlation between successive received OFDM symbols. STA consists of five steps, first it performs the initial channel estimation as expressed in (5), and (6). After that, frequency domain averaging is applied as follows:

$$\hat{\hat{\boldsymbol{h}}}_{\mathrm{FD}_i}[k'] = \sum_{\lambda=-\beta}^{\lambda=\beta} \omega_\lambda \hat{\hat{\boldsymbol{h}}}_{\mathrm{Initial}_i}[k+\lambda], \ \omega_\lambda = \frac{1}{2\beta+1}, \tag{7}$$

where $2\beta + 1$ represents the number of subcarriers that is averaged and ω_λ is a set of weighting coefficients with unit sum, and $k' = [3, .., K_{\mathrm{on}} - 2]$. Note that the subcarriers at both edges of the received OFDM symbols (positions = 1, 2, 51, 52) are not included in the frequency domain averaging process. Finally, STA performs time averaging in order to calculate the final STA channel estimate as following:

$$\hat{\hat{\boldsymbol{h}}}_{\mathrm{STA}_i}[k] = (1 - \frac{1}{\alpha})\hat{\hat{\boldsymbol{h}}}_{\mathrm{STA}_{i-1}}[k] + \frac{1}{\alpha}\hat{\hat{\boldsymbol{h}}}_{\mathrm{FD}_i}[k]. \tag{8}$$

The time and frequency averaging is used to alleviate the demapping error, and therefore to improve the estimation accuracy [2]. This final channel estimate is then used to equalize the next received OFDM symbol, and this procedure continues until all symbols in the packet are equalized. The values of α and β depend on the nature of vehicular channels. According to the analysis developed in [2], significant accuracy can be achieved by adjusting these values according to the knowledge of the vehicular channel environment. In general, a large value of α is more suitable to a slowly changing channel, and a smaller value of α is more suitable for a fast changing channel. Similarly, a large value of β is more suitable for a relatively flat fading channel, whereas a smaller value of β is more suitable for a frequency-selective channel. Larger values of α and β help reducing measurement noise since it adds redundancy to the measurements. Moreover, in vehicular communications where the channel varies very fast, if α and β are made very large, then too many subcarriers (in frequency) or symbols (in time) are averaged such that the estimated channel is blurred, thus the resulting channel estimate becomes inaccurate. However, such information is practically quite hard to obtain since in real case scenarios, since the behavior of the vehicular channel besides the noise impact cannot be accurately predicted. Therefore, fixed values of these parameters can be properly chosen to have an acceptable performance degradation.

D. CDP ESTIMATION SCHEME

The CDP estimation scheme [3] has been proposed to alleviate the effects of demapping errors when updating the channel estimate. The CDP scheme relies on the fact, that the time correlation of the channel response between two adjacent

OFDM symbols is high. This scheme shares the first three steps required by the initial channel estimation procedure. Thus, CDP channel estimation proceeds after (4), (5), and (6), but it only employs the data subcarriers within the received OFDM symbol.

Instead of the time and frequency averaging performed in STA, CDP proposes two different steps considering the high time correlation between successive received OFDM symbols. After calculating the initial channel estimation from (6), the previously received OFDM symbol is equalized by $\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[k]$ and $\hat{\tilde{\boldsymbol{h}}}_{\text{CDP}_{i-1}}[k]$ respectively as follows:

$$\begin{aligned} \boldsymbol{y'}_{\text{eq}_{i-1}}[k] &= \frac{\boldsymbol{y}_{i-1}[k]}{\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[k]}, \\ \boldsymbol{y''}_{\text{eq}_{i-1}}[k] &= \frac{\boldsymbol{y}_{i-1}[k]}{\hat{\tilde{\boldsymbol{h}}}_{\text{CDP}_{i-1}}[k]}. \end{aligned} \tag{9}$$

Then, the obtained $\boldsymbol{y'}_{\text{eq}_{i-1}}[k]$ and $\boldsymbol{y''}_{\text{eq}_{i-1}}[k]$ are demapped into $\boldsymbol{d'}_{i-1}[k]$ and $\boldsymbol{d''}_{i-1}[k]$ respectively according to the used constellation. Finally, CDP applies a comparison step where the final $\hat{\tilde{\boldsymbol{h}}}_{\text{CDP}_i}[k]$ is obtained as:

$$\hat{\tilde{\boldsymbol{h}}}_{\text{CDP}_i}[k] = \begin{cases} \hat{\tilde{\boldsymbol{h}}}_{\text{CDP}_{i-1}}[k], & \boldsymbol{d'}_{i-1}[k] \neq \boldsymbol{d''}_{i-1}[k] \\ \hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[k], & \boldsymbol{d'}_{i-1}[k] = \boldsymbol{d''}_{i-1}[k]. \end{cases} \tag{10}$$

The case where $\boldsymbol{d'}_{i-1}[k] \neq \boldsymbol{d''}_{i-1}[k]$, indicates that $\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[k]$ is unreliable, and thus, the previous channel estimate is used. In other words, in order for $\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[k]$ to be considered as a reliable channel estimate, the demapping results of the previous received OFDM symbol when it is equalized by $\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[k]$ and $\hat{\tilde{\boldsymbol{h}}}_{\text{CDP}_{i-1}}[k]$ should be identical due to the high time correlation between two received successive OFDM symbols. This channel update process suffers from performance degradation in the high mobility environments, and particularly, for the high order modulation techniques such as 16QAM and 64QAM.

E. TRFI ESTIMATION SCHEME

To improve the accuracy of the CDP estimation for higher order modulation, the TRFI scheme [4] has been proposed. It is mainly based on the frequency domain interpolation of the reliable constructed data pilots. Thus, instead of applying the comparison step as in CDP (10), TRFI applies channel estimation steps in (4), (5), (6), and (9). The demapping results $\boldsymbol{d'}_{i-1}[k]$ and $\boldsymbol{d''}_{i-1}[k]$ obtained after (9) are used to recover the channel estimate by defining two subcarriers sets: (i) Reliable subcarriers (RS) and (ii) Unreliable subcarriers (URS). The main idea of TRFI scheme is to use the RS channel estimates to interpolate the channel at the URS based on the reliability test described in Algorithm 1.

Here, $\mathcal{K}_p$ and $\mathcal{K}_d$ denote the set of pilot and data subcarriers with $K_d = |\mathcal{K}_d|$ and $k_p = |\mathcal{K}_p|$, respectively. Since the four comb pilot subcarriers are known, they are all considered as reliable subcarriers channel estimates. For the data subcarriers, a reliability test is applied, where the channel estimate at the k-th subcarrier is considered reliable when $\boldsymbol{d'}_{i-1}[k]$ and $\boldsymbol{d''}_{i-1}[k]$ are equal. Finally, frequency domain interpolation is applied by using the channel estimates at RS to recover the channel estimates at the URS. It is shown in [4] that cubic interpolation fits better than the linear one.

Algorithm 1 TRFI Reliability Test

Require: $\boldsymbol{d'}_{i-1}[k]$ and $\boldsymbol{d''}_{i-1}[k]$
  for all $k \in \mathcal{K}_p$ **do**
    RS $\leftarrow$ RS $+ k$
  end for
  for all $k \in \mathcal{K}_d$ **do**
    if $\boldsymbol{d'}_{i-1}[k] == \boldsymbol{d''}_{i-1}[k]$ **then**
      $\hat{\tilde{\boldsymbol{h}}}_{\text{TRFI}_i}[k] = \hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[k]$
      RS $\leftarrow$ RS $+ k$
    else
      URS $\leftarrow$ URS $+ k$
    end if
  end for
  $\hat{\tilde{\boldsymbol{h}}}_{\text{TRFI}_i}[\text{URS}] = \text{cubic Interpolation}(\hat{\tilde{\boldsymbol{h}}}_{\text{TRFI}_i}[\text{RS}])$

F. MMSE-VP ESTIMATION SCHEME

MMSE-VP channel estimation scheme is proposed to tackle the drawbacks caused by the channel variations in vehicular environments [5]. Similarly to all classical channel estimation schemes, MMSE-VP employs the initial channel estimation performed in (5), and (6) by using the frequency-domain MMSE channel estimation to compute the final estimate. First of all, MMSE-VP builds a virtual pilots vector by rearranging the initial channel estimates $\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[k]$ obtained from (6) as follows:

$$\hat{\tilde{\boldsymbol{h}}}_{\text{vp}_i}[k] = [\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[-21], \hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[k], \ldots, \hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[21]]; \quad k \in N_d. \tag{11}$$

Here MMSE-VP re-arrange $\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[k]$ by placing two pilots at the beginning ($k = -21, -7$) and two pilots at the end ($k = 7, 21$), this new arrangement is called virtual pilot vector and it is denoted by $\hat{\tilde{\boldsymbol{h}}}_{\text{vp}_i}[k]$. As noticed in (11), the virtual pilot vector $\hat{\tilde{\boldsymbol{h}}}_{\text{vp}_i}[k]$ contains pilots subcarriers on its boundaries, such that the first two pilots $\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[-21]$ and $\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[-7]$ are added at the beginning while the other two pilots $\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[7]$ and $\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[21]$ are concatenated at the end. The final channel estimate for each received OFDM symbol is calculated using the frequency domain MMSE scheme as illustrated below:

$$\hat{\tilde{\boldsymbol{h}}}_{\text{MMSE}_i} = \boldsymbol{R}_{\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}\hat{\tilde{\boldsymbol{h}}}_{\text{vp}_i}} \left(\boldsymbol{R}_{\hat{\tilde{\boldsymbol{h}}}_{\text{vp}_i}\hat{\tilde{\boldsymbol{h}}}_{\text{vp}_i}} + \sigma_i^2 \boldsymbol{I} \right)^{-1} \hat{\tilde{\boldsymbol{h}}}_{\text{vp}_i}, \tag{12}$$

where $\boldsymbol{R}_{\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}\hat{\tilde{\boldsymbol{h}}}_{\text{vp}_i}}$ is the cross-correlation matrix between the initial channel estimate vector, $\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}$ and the virtual pilots vector $\hat{\tilde{\boldsymbol{h}}}_{\text{vp}_i}$, $\boldsymbol{R}_{\hat{\tilde{\boldsymbol{h}}}_{\text{vp}_i}\hat{\tilde{\boldsymbol{h}}}_{\text{vp}_i}}$ is the auto-correlation matrix of the

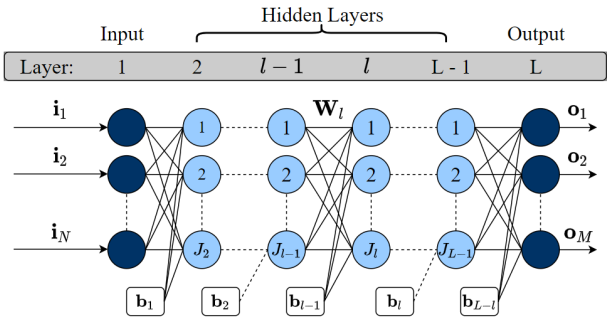


FIGURE 4. DNN architecture.

virtual pilots vector. $\boldsymbol{I}$ is the identity unit matrix, and σ_i^2 is the average noise power in the i-th received OFDM symbol. The MMSE-VP scheme uses the correlation characteristics between the initial channel estimates and the virtual pilots vectors to reduce the channel estimation errors. This scheme provides a better performance than the existing classical channel estimation techniques especially when QPSK modulation is utilized, but it still suffers from performance degradation in high mobility scenarios and it has lower performance than the STA in a low SNR region.

G. AE-DNN ESTIMATION SCHEME

Inspired by the fact that the auto-encoder (AE) neural networks are useful for solving diverse problems in the physical layer wireless communication systems. AE-DNN channel estimation scheme [11] has been proposed to recover possible estimation and demapping errors resulting from (6), and thus to help in mitigating the error propagation issue of the initial channel estimation. As shown in Fig. 5, the output of the initial channel estimation $\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[k]$ is fed as an input to a DNN with three hidden layers, and then the corrected AE-DNN channel estimates are used to equalize the received symbol. AE-DNN scheme outperforms classical channel estimation schemes. Moreover, our proposed scheme presented in Sections IV and V is able to outperform AE-DNN with less computational complexity.

IV. PROPOSED DL-BASED CHANNEL ESTIMATION SCHEME

In this section, the DNN main concepts are briefly described, and a detailed explanation of the proposed DL-based channel estimation scheme is presented.

A. DEEP NEURAL NETWORK (DNN)

Neural networks are among the most popular machine learning algorithms [13]. Initially, neural networks are inspired by the neural architecture of a human brain, and like in a human brain the basic building block is called a neuron. Its functionality is similar to a human neuron, i.e. it takes in some inputs and fires an output. In purely mathematical terms, a neuron represents a placeholder for a mathematical function, and its only job is to provide an output by applying the function on the inputs provided. Neurons are stacked together to form a layer. The neural network consists at least of one layer, and when multiple layers are used, the neural network is called DNN.

Consider a DNN that consists of L layers, including one input layer, $L-2$ hidden layers, and one output layer as shown in Fig. 4. The l-th hidden layer of the network consists of J neurons where $2 \leq l \leq L-1$, and $1 \leq j \leq J$. The DNN inputs $\boldsymbol{i}$ and outputs $\boldsymbol{o}$ are expressed as $\boldsymbol{i} = [\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_N] \in \mathbb{R}^{N\times 1}$ and $\boldsymbol{o} = [\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_M] \in \mathbb{R}^{M\times 1}$, where N and M denote the number of DNN inputs and outputs respectively. $\boldsymbol{W}_l \in \mathbb{R}^{J_{l-1}\times J_l}$, and $\boldsymbol{b}_l \in \mathbb{R}^{J_l\times 1}$ are used to denote the weight matrix and the bias vector of the l-th hidden layer respectively. Each neuron $n_{(l,j)}$ performs a nonlinear transform of a weighted summation of output values of the preceding layer. This nonlinear transformation is represented by the activation function $f_{(l,j)}$ on the neuron's input vector $\boldsymbol{i}_{(l)} \in \mathbb{R}^{J_{l-1}\times 1}$ using its weight vector $\boldsymbol{\omega}_{(l,j)} \in \mathbb{R}^{J_{l-1}\times 1}$, and bias $b_{(l,j)}$ respectively. The neuron's output $o_{(l,j)}$ is:

$$o_{(l,j)} = f_{(l,j)}\Big(b_{(l,j)} + \boldsymbol{\omega}_{(l,j)}^T \boldsymbol{i}_{(l)}\Big). \tag{13}$$

The DNN over all output of the l-th hidden layer is represented by the vector form:

$$\boldsymbol{o}_{(l)} = \boldsymbol{f}_{(l)}\Big(\boldsymbol{b}_{(l)} + \boldsymbol{W}_{(l)}\boldsymbol{i}_{(l)}\Big),\ \boldsymbol{i}_{(l+1)} = \boldsymbol{o}_{(l)}, \tag{14}$$

where $\boldsymbol{f}_{(l)}$ is a vector that results from the stacking of the n_l activation functions.

Once the DNN architecture has been chosen, the parameter $\theta = (\boldsymbol{W}, \boldsymbol{B})$ that represents the total DNN weights and biases have to be estimated through the learning procedure applied during the DNN training phase. As well known, θ estimation is obtained by minimizing a loss function Loss(θ) (15). A loss function measures how far apart the predicted DNN outputs ($\boldsymbol{o}_{(L)}^{(\mathrm{P})}$) from the true outputs ($\boldsymbol{o}_{(L)}^{(\mathrm{T})}$). Therefore, DNN training phase carried over N_{train} training samples can be described in two steps: (i) calculate the loss, and (ii) update θ. This process will be repeated until convergence, so that the loss becomes very small.

$$\text{Loss}(\theta) = \arg\min_{\theta}(\boldsymbol{o}_{(L)}^{(\mathrm{P})} - \boldsymbol{o}_{(L)}^{(\mathrm{T})}). \tag{15}$$

Various optimization algorithms can be used to minimize Loss(θ) by iteratively updating the parameter θ, i.e., stochastic gradient descent [13], root mean square prop [14], adaptive moment estimation (ADAM) [15]. DNN optimizers updates θ according to the magnitude of the loss derivative with respect to it as follows:

$$\theta_{\text{new}} = \theta - \rho\frac{\partial \text{Loss}(\theta)}{\partial \theta}, \tag{16}$$

where ρ represents the learning rate of the DNN, which controls how quickly θ is updated. Smaller learning rates require more training, given the smaller changes made to θ in each update, whereas larger learning rates result in rapid changes and require less training. The final step after DNN training, is to test the trained DNN on new data so that its

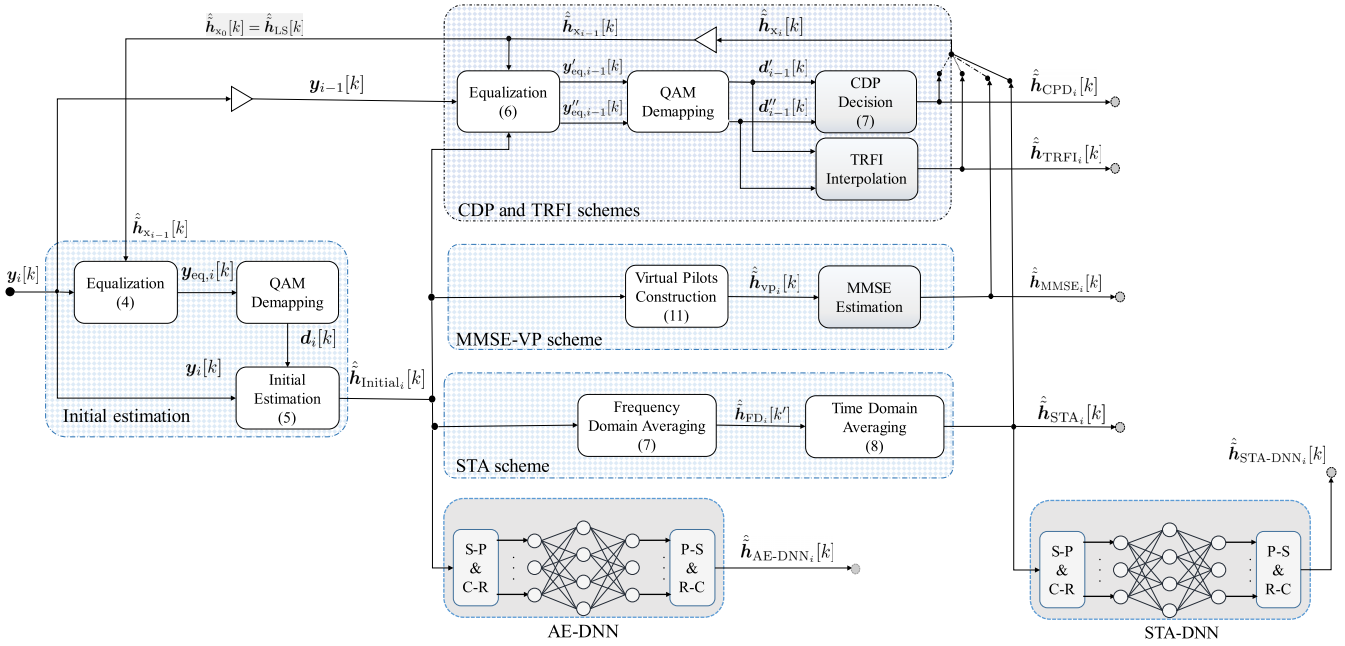


FIGURE 5. Proposed STA-DNN channel estimation scheme block diagram.

performance is evaluated. A detailed comprehensive analysis of DNN different principles is presented in [16].

B. PROPOSED DNN-BASED CHANNEL ESTIMATION SCHEME

The proposed STA-DNN channel estimation scheme is mainly based on applying a DNN based processing to the coarse STA channel estimates, in order to achieve a fine channel estimation. It is clearly shown from (8), that STA updates the current channel estimate using a linear combination of the previous channel and initially estimated channel. Therefore, by integrating DNN as an additional module in the STA scheme, we add a non-linear processing to capture the non-linear dependencies between the previous and the current channel.

As shown in Fig. 5, the proposed scheme first applies the STA channel estimation, and then, an additional non-linear processing using DNN is performed. The DNN captures more features of the time-frequency correlations of the channel samples. Thus, the DNN corrects the estimation error of the classical STA scheme. This is achieved by minimizing the mean squared error between the ideal channel $\tilde{\boldsymbol{h}}_i$ and STA estimated channel $\hat{\tilde{\boldsymbol{h}}}_{\text{STA}_i}$ as follows:

$$\text{MSE}_{\text{STA-DNN}} = \frac{1}{N_T}.\sum_{i=1}^{N_T} \tilde{\boldsymbol{h}}_i - \hat{\tilde{\boldsymbol{h}}}_{\text{STA}_i}, \tag{17}$$

where N_T represents the number of samples considered during the DNN training. Table 2 illustrates the different parameters of the proposed STA-DNN architectures. After performing STA scheme, STA channel estimates should be converted from complex to real valued domain, in order to introduce them to the DNN input. Therefore, $\hat{\tilde{\boldsymbol{h}}}_{\text{STA}_i}$ will be processed according to the vector realization function:

$$f_{\mathbb{R}}(V) = [\Re(V); \Im(V)]. \tag{18}$$

After that, $\hat{\tilde{\boldsymbol{h}}}^{(R)}_{\text{STA}_i} \in \mathbb{R}^{2\mathcal{K}_{\text{on}}\times 1}$ is fed as an input to STA-DNN. Finally, the corrected STA channel estimates $\hat{\tilde{\boldsymbol{h}}}_{\text{STA-DNN}_i}$, are processed again to get back $|\mathcal{K}_{\text{on}}|$ complex valued domain.

The decision to base our scheme on the STA channel estimation and not other classical schemes is justified by the fact that STA scheme takes into consideration the time and frequency correlation of successive received symbols. However, it suffers from performance degradation in real case scenarios due to fixing α and β coefficients. Moreover, accurate estimation of α and β requires the knowledge of the channel characteristics, which is hard to obtain in practice.

TABLE 2. Proposed DNN parameters.

Parameter	Values
STA-DNN (hidden layers; neurons per layer)	(3; 15-10-15)
STA-DNN (hidden layers; neurons per layer)	(3; 15-15-15)
Activation function	ReLU
Number of epochs	500
Training samples	800000
Testing samples	200000
Batch size	128
Optimizer	ADAM
Loss function	MSE
Learning rate	0.001

Our proposed STA-DNN first applies a basic STA scheme as a coarse channel estimation, and then a non-linear post processing based on DNN, for a fine channel estimation that capture more non-linear correlations between the channel samples in time and frequency domains. By doing so, we implicitly overcome the performance loss of STA, and the errors resulting from fixing α and β values. It is worth mentioning that even though initial channel estimation is less complex than STA scheme, but we have conducted several experiments to optimize the hyper parameters of the proposed DNN architectures, in order to achieve a better performance with less overall complexity than AE-DNN, as described in Section V.

In our research work [17], we have proved that the DNN performance highly depends on the SNR values used in the training phase. The training performed at the highest expected SNR value (which we consider here equal to 30 dB) provides the best performance. In fact, when the training is performed at a high SNR value, the DNN is able to learn better the channel, because in this SNR range the impact of the channel is higher than the impact of the noise. Thanks to the good generalization properties of DNN, it can still estimate the channel even if the noise is increased i.e. at low SNR values. Moreover, if exact SNR knowledge is available at the receiver, then we can train offline several DNNs each for a fixed SNRs value, and according to the estimated SNR value at the receiver, we can select the appropriate DNNs in an adaptive manner.

V. SIMULATION RESULTS

In this section, normalized mean-squared error (NMSE) and bit error rate (BER) are used to evaluate the performance of the proposed STA-DNN channel estimation schemes. First, we present the vehicular channel models employed in our simulations. After that, the simulation results of different scenarios are presented.

A. VEHICULAR CHANNEL MODELS

Vehicular channel models characteristics have been explored in the literature [18]–[20]. Six realistic channel models have been widely used for vehicular communication environments [21]. These models are obtained through a channel measurement campaign which has been performed in the metropolitan Atlanta, Georgia, USA. The campaign consists of six different scenarios, three VTV models and three RTV models as summarized below:

- **VTV Expressway (VTV-EX)**: the measurements here are performed between two vehicles entering the highway at the same time, and moving in opposite direction, then they are accelerated to reach 104 km/h. In this scenario there is no wall separating the two highway sides.
- **VTV Urban Canyon (VTV-UC)**: this scenario has been measured in Edge wood Avenue in Downtown Atlanta, where urban canyon characteristics exist. The vehicles move at 32-48 km/hr velocity range in a dense traffic environment.
- **VTV Expressway Same Direction with Wall (VTV-SDWW)**: The communication between the two vehicles is established on a highway having center wall between its lanes. The separation distance between both vehicles is 300–400 m, and vehicles speed was 104 km/hr.
- **RTV Suburban Street (RTV-SS)**: The road side unit (RSU) is placed at roads intersection in a sub urban environment. The vehicle is far away from the RSU by 100 m and moving at 32-48 km/hr velocity range.
- **RTV Expressway (RTV-EX)**: In this scenario, the RSU is placed on a highway, and the vehicle moves towards the RSU at a speed of 104 km/hr.
- **RTV Urban Canyon (RTV-UC)**: the measurements here are performed in a dense traffic environment. The RSU transmitting antenna is mounted on a pole near the urban intersection of Peach tree Street and Peach tree Circle. The vehicle moves at 32-48 km/hr speed and 100 m far away from the RSU.

TABLE 3. Vehicular channel models characteristics.

Channel model	Distance [m]	Velocity [kmph]	Doppler [Hz]	No. Paths
VTV-EX	300-400	104	1000-1200	11
VTV-UC	100	32-48	400-500	12
VTV-SDWW	300-400	104	900-1150	12
RTV-SS	100	32-48	300-500	12
RTV-EX	300-400	104	600-700	12
RTV-UC	100	32-48	300	12

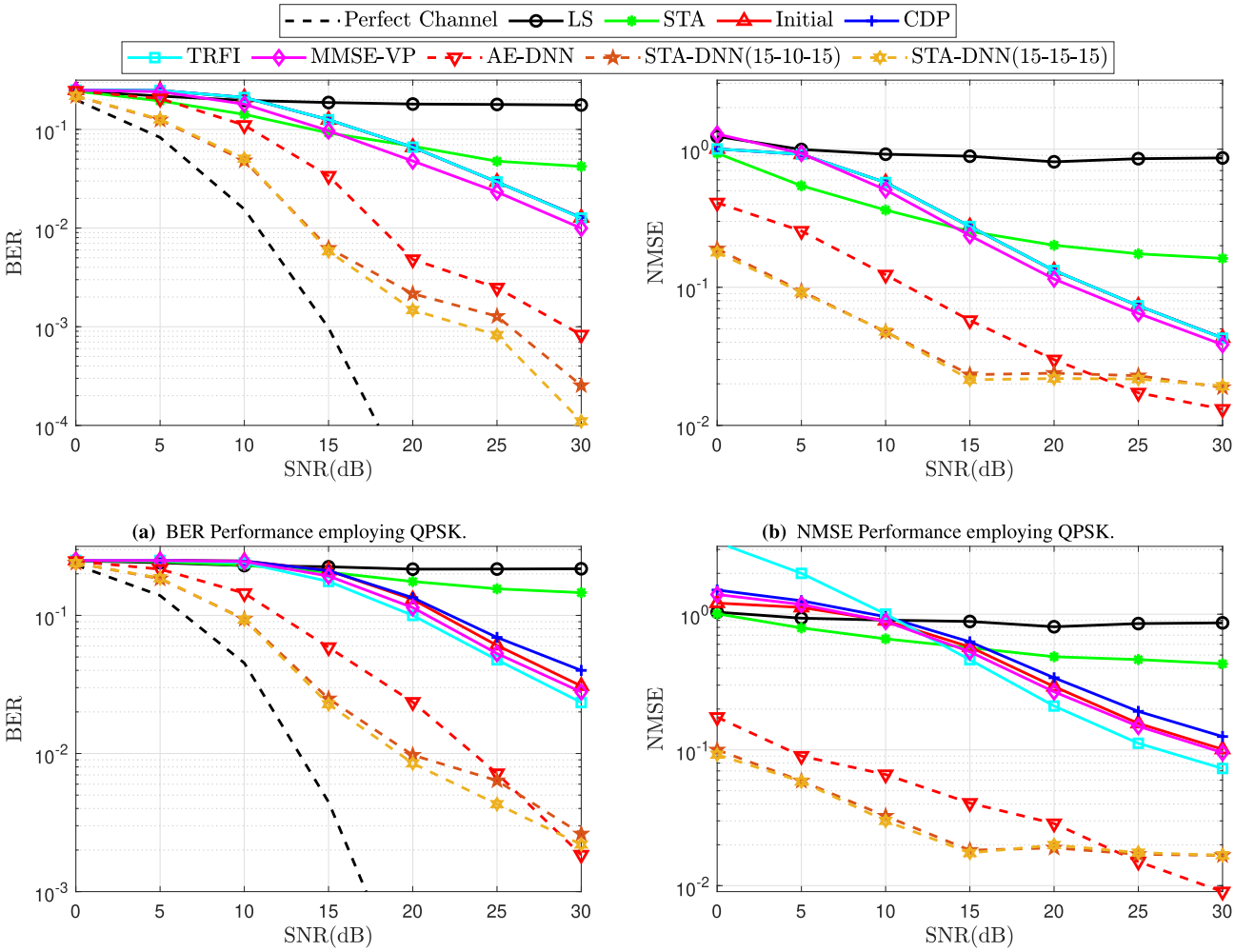
Table 3 presents a detailed overview about several vehicular channel models characteristics.

B. BER AND NMSE PERFORMANCE

To evaluate the BER and NMSE performance under different high and low mobility vehicular environments, two vehicular channel models are chosen: (i) VTV-SDWW that has high Doppler shifts (900-1150 Hz) and thus high time-varying channel variations. (ii) RTV-UC that features lower Doppler shifts (400-500 Hz). The comparison of the IEEE 802.11p classical channel estimation schemes, the recently proposed AE-DNN scheme [11], and our proposed STA-DNN schemes are performed over the chosen vehicular channel models in three different criteria: (i) Modulation order, (ii) Mobility, and (iii) DNN architectures.

1) MODULATION ORDER

For QPSK modulation order, we can notice from Fig. 6 (a), 6 (b), 7 (a), and 7 (b) that the STA scheme outperforms other classical schemes in low SNR region due to the frequency and time averaging operations used in STA (7), (8). While in high SNR regions, classical schemes express a



(a) BER Performance employing QPSK.

(b) NMSE Performance employing QPSK.

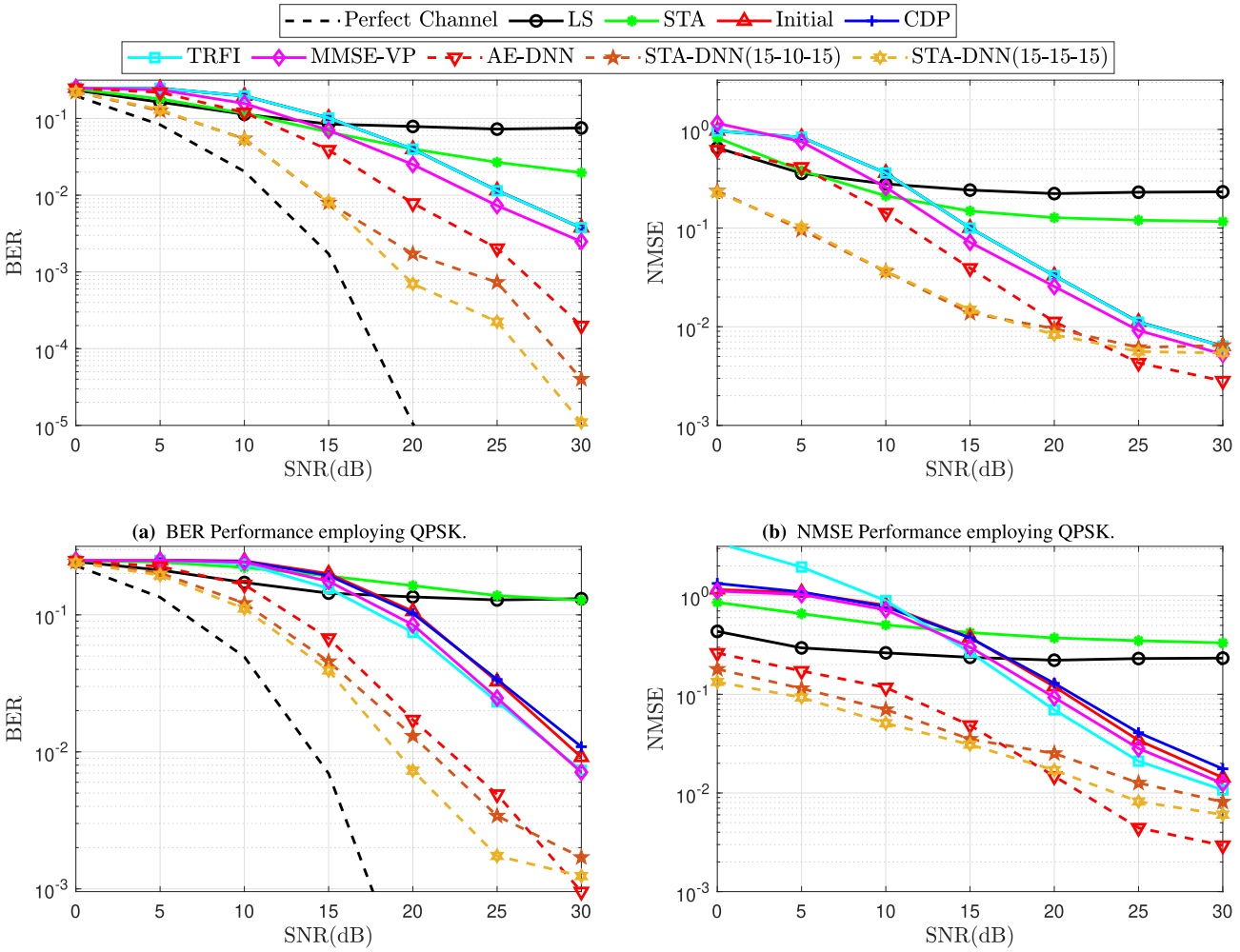
(c) BER Performance employing 16QAM.

(d) NMSE Performance employing 16QAM.

FIGURE 6. BER and NMSE simulation results for VTV-SDWW vehicular channel model employing QPSK and 16QAM modulation respectively.

significant improvement over the STA scheme, where CDP, TRFI, and equalizing by the initial channel estimates (6) reach the same performance. This is due to the fact that when the SNR is low, the impact of noise and interference are high and powerful enough to shift the equalized received OFDM symbol $\boldsymbol{y}_{\text{eq}_i}[k]$ to wrong regions and as a result, its demapping $\boldsymbol{d}_i[k]$ is shifted to incorrect constellation points. As the SNR increases, the aforementioned influence is reduced and thus the superiority of the classical scheme emerges over STA. It is worth mentioning that in order to achieve the optimal performance of the STA scheme, the frequency-domain averaging window β, and the time-domain averaging coefficient α must be equal to 2 as discussed in [2]. But, fixing these parameters makes the smoothing in the time and frequency domains not effective under vehicular environment. Thus, the gradually accumulated demapping error of $\boldsymbol{d}_i[k]$ cannot be well mitigated using fixed β and α. Hence the emergence of the error floor in the STA scheme performance. Moreover, AE-DNN outperforms the classical schemes, but it is not sufficient, especially in low SNR regions, since it just corrects the demapping error of (5) and neglects the frequency and time correlation of the received symbols. Employing DNN with STA scheme shows a considerable performance improvement over AE-DNN by which STA-DNN outperforms AE-DNN up to 7 dB gain in terms of SNR for a BER = 10^{-3}.

When adopting high modulation order (16QAM), TRFI scheme shows a slight performance improvement over all classical schemes as shown in Fig. 6 (c), and 7 (c). This reveals that the frequency-domain interpolation that is implemented by TRFI scheme is effective in estimating the channel at the unreliable subcarriers as explained in algorithm 1. On the other hand, the classical STA scheme suffers from a severe degradation as noticed in Fig. 6 (c). Moreover, STA-DNN can still outperforms the AE-DNN scheme by up to 4 dB gain in terms of SNR for a BER = 10^{-2}.



(a) BER Performance employing QPSK.

(b) NMSE Performance employing QPSK.

(c) BER Performance employing 16QAM.

(d) NMSE Performance employing 16QAM.

FIGURE 7. BER and NMSE simulation results for VTV-UC vehicular channel model employing QPSK and 16QAM modulation respectively.

2) MOBILITY

Simulations are conducted over two different vehicular scenarios where low and high mobility conditions are taken into consideration in **VTV-UC** and **VTV-SDWW** channel models, respectively. In both cases, the impact of mobility is clearly shown from the different figures, where in the high mobility scenario, performance degradation of all the benchmarked schemes can be noticed. It is worth mentioning that DNN based channel estimation schemes outperform classical schemes in both low and high mobility scenarios. Moreover, STA-DNN shows a great superiority over AE-DNN especially in low SNR region.

3) DNN ARCHITECTURES

An intensive investigation was performed on several DNN architectures in order to select the more suitable hyper parameters in terms of both performance and complexity.

STA-DNN(15-15-15) outperforms STA-DNN(15-10-15), especially with higher order modulation (16QAM), at the cost of a slightly higher computational complexity. Moreover, STA-DNN has better optimized DNN architectures which definitely makes it less complex than AE-DNN as we will discuss in Section VI.

Simulation results according to the studied criteria show that integrating DNN with classical estimators attains significant performance improvement. Correcting the estimation error of the initial channel estimates $\hat{\tilde{\boldsymbol{h}}}_{\text{Initial}_i}[k]$ (6), which is the main idea of the recently proposed AE-DNN scheme [11] is not sufficient, since it just corrects the demapping error of (5) and neglects the frequency and time correlation of the received symbols. Moreover, employing DNN as a post processing module after the classical STA scheme reflects the performance superiority among other channel estimation schemes, due to the fact that classical STA scheme takes into consideration both

frequency and time correlation between the received OFDM symbols.

VI. COMPUTATIONAL COMPLEXITY ANALYSIS

In this section, a computational complexity analysis of the proposed and the benchmark channel estimation schemes is presented. The computational complexity is computed in terms of real-valued mathematical operations including multiplication/division and summation/subtraction needed to estimate the channel for the received OFDM symbol.

The LS estimation is the simplest scheme, where the received preambles symbols are added to each others, resulting in 2 K_{on} summations. After that, the summation result will be divided by the predefined preamble, thus 2 K_{on} divisions are also required since the predefined preamble $\boldsymbol{p}[k]$ is BPSK modulated. Therefore, the total number of divisions and summations needed by the LS scheme is 2 K_{on} and 2 K_{on} respectively.

According to (5) and (6), the initial channel estimation requires 2 complex-valued divisions. We note that, each complex-valued division requires 6 real-valued multiplications, 2 real-valued divisions, 2 real-valued summations, and 1 real-valued subtraction. Therefore, the overall computational complexity of the initial channel estimation accumulated by the LS required operations is 18 K_{on} multiplications/divisions and 8 K_{on} summations/subtractions.

Concerning the STA scheme, it requires in addition to the initial channel estimation steps, the frequency domain averaging that needs 2 K_{d} multiplications, and 10 K_{d} summations. After that, 4 K_{on} multiplications and 2 K_{on} summations are needed to accomplish the time averaging operation. Hence, the accumulated computational complexity of STA scheme is 22 K_{on}+2 K_{d} multiplications/divisions and 10 K_{on} + 10 K_{d} summations/subtractions.

The CDP scheme is also based on the initial channel estimate (6), but only K_{d} subcarriers are processed in CDP. The additional computation required by CDP in (10) is 16 K_{d} multiplications and 6 K_{d}. Hence, the CDP scheme requires in total 34 K_{d} multiplications/divisions and 14 K_{d} summations/subtractions.

The TRFI scheme is considered as an improved version of the CDP scheme, but K_{on} subcarriers are considered in the estimation process. The channel estimation procedure of TRFI scheme is similar to CDP scheme except the final step. Instead of updating the channel estimates as described in (10), TRFI utilizes the frequency domain interpolation to update the channel estimates. The computational complexity of TRFI scheme relies mainly on the size of unreliable subcarriers set. In order to have a good approximation of the unreliable subcarriers set size, we ran our simulation 10000 times, and we found that an average of $K_{\text{int}} = 10$ subcarriers are considered as unreliable subcarriers in each received OFDM symbol. Each unreliable subcarrier is bounded by two reliable subcarriers. Theoretically, the cubic interpolation of points located within a known interval requires the calculation of the third degree polynomial coefficients. This polynomial (19) expresses the behaviour of the interpolated curve within the specified interval.

$$f(x) = a.x^3 + b.x^2 + c.x + d. \tag{19}$$

As shown in Appendix A, the cubic interpolation of one subcarrier between two reliable subcarriers requires 26 multiplications/divisions and 30 summations/subtractions. Thus, the computational complexity of TRFI is 34 K_{on} + 26K_{int} multiplications/divisions and 14 K_{on} + 30K_{int} summations/subtractions.

MMSE-VP scheme suffers from high computational complexity due to the correlation matrices manipulation in addition to the matrix inversion operation. $\boldsymbol{R}_{\hat{\boldsymbol{h}}_{\text{Initial}_i}\hat{\boldsymbol{h}}_{\text{vp}_i}}$ calculation requires $4K_{\text{on}} + 2K_{\text{on}}^2$ real-valued multiplications and $3K_{\text{on}} + 2K_{\text{on}}^2$ summations. The same calculation is performed for $\boldsymbol{R}_{\hat{\boldsymbol{h}}_{\text{vp}_i}\hat{\boldsymbol{h}}_{\text{vp}_i}}$ manipulation. Moreover, $\boldsymbol{R}_{\hat{\boldsymbol{h}}_{\text{vp}_i}\hat{\boldsymbol{h}}_{\text{vp}_i}}$ as presented in (12) is added to the noise power vector multiplied by the identity matrix. Thus, $2K_{\text{on}}^2$ multiplications and $2K_{\text{on}}^2$ summations are required in this step. Finally, matrix inversion followed by multiplication is applied to calculate the final MMSE-VP channel estimate. This requires K_{on}^3 complex multiplications, which is equivalent to 4 K_{on}^3 real-valued multiplications and 3 K_{on}^3 real-valued summations/subtractions. Therefore, the overall accumulated computational complexity of MMSE-VP is 26 K_{on}+6 K_{on}^2+4 K_{on}^3 multiplications/divisions and 14 K_{on}+6 K_{on}^2+3 K_{on}^3 summations/subtractions operations.

On the other hand, when working with DNN, we are interested in evaluating the online computational complexity that can be represented by the number of multiplications needed to compute the activation of all neurons (vector product) in all network layers. The transition between the l-th and $(l-1)$-th layers requires $J_l J_{l-1}$ multiplications for the linear transform. The additional operations in DNN are simple, which include the sum of bias and the vector product in the activation functions. Therefore, the total number of real-valued multiplications and summations in DNN network is given by:

$$\begin{aligned} N_{\text{mul/sum}} &= \sum_{l=2}^{L} J_{l-1}J_l + J_{l-1}J_l \\ &= 2\sum_{l=2}^{L} J_{l-1}J_l,\ J_1 = 2K_{\text{on}},\ J_L = 2K_{\text{on}}. \end{aligned} \tag{20}$$

All the DNNs considered in this paper are 3-hidden layers DNNs. Thus, the general DNN architecture is DNN (J_2, J_3, J_4), where J_2, J_3, J_4 are the number of neurons within the first, second, and third hidden layer respectively. The transition from the DNN input layer l_1 to l_2 requires J_1J_2 multiplications and J_1J_2 summations respectively, where J_1 =2 K_{on}. Similarly for the other successive layers, the DNN requires $2J_2J_3$, $2J_3J_4$, and $2J_4J_5$ operations respectively where J_5 =2 K_{on} denotes the DNN outputs.

As presented in [11], the AE-DNN architecture is composed of three hidden layers with $J_1 = J_5$ =2 K_{on},

TABLE 4. Computation complexity in terms of real-valued operations.

Estimation Step	Mul./Div.	Sum./Sub.
LS	$2K_{\text{on}}$	$2K_{\text{on}}$
Initial estimation	$16K_{\text{on}}$	$6K_{\text{on}}$
STA	$4K_{\text{on}} + 2K_{\text{d}}$	$2K_{\text{on}} + 10K_{\text{d}}$
CDP	$16K_{\text{d}}$	$6K_{\text{d}}$
TRFI	$26K_{\text{int}}$	$30K_{\text{int}}$
MMSE-VP	$8K_{\text{on}} + 6K_{\text{on}}^2 + 4K_{\text{on}}^3$	$6K_{\text{on}} + 6K_{\text{on}}^2 + 3K_{\text{on}}^3$
DNN(J_2-J_3-J_4)	$2K_{\text{on}}J_2 + J_2J_3 + J_3J_4 + 2K_{\text{on}}J_4$	$2K_{\text{on}}J_2 + J_2J_3 + J_3J_4 + 2K_{\text{on}}J_4$
Overall channel estimation		
STA-DNN(15-10-15)	$82K_{\text{on}} + 2K_d + 300$	$70K_{\text{on}} + 10K_d + 300$
STA-DNN(15-15-15)	$82K_{\text{on}} + 2K_d + 450$	$70K_{\text{on}} + 10K_d + 450$
AE-DNN(40-20-40)	$178K_{\text{on}} + 1600$	$168K_{\text{on}} + 1600$

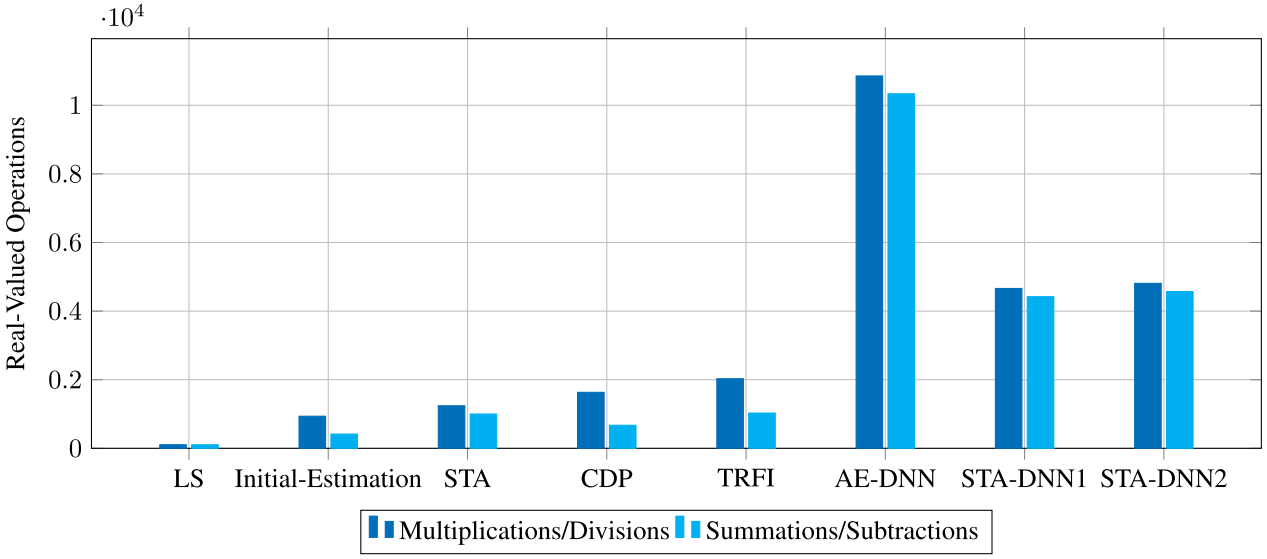


FIGURE 8. Computational complexity for different channel estimation schemes.

$J_2 = J_4 = 40$, and $J_3 = 20$ neurons respectively. Thus, AE-DNN requires 4 $K_{\text{on}}J_2 + 2J_2J_3$ multiplications, and 2 $K_{\text{on}} + 2J_2 + J_3$ summations. Moreover, the computational complexity of LS and the initial channel estimation are accumulated for AE-DNN computational complexity resulting in 178 $K_{\text{on}} + 1600$ multiplications and 168 $K_{\text{on}} + 1600$ summations/subtractions are required by the AE-DNN.

The proposed channel estimation scheme illustrates two DNN architectures, STA-DNN(15-10-15) and STA-DNN(15-15-15) denoted by STA-DNN1 and STA-DNN2 respectively. It can be observed that these architectures are less complex than the AE-DNN architecture. STA-DNN (15-10-15) requires 82 K_{on}+2 K_d + 300 multiplications, and 70 K_{on} + 10 K_d + 300 summations/subtractions, where $J_2 = J_4 = 15$, and $J_3 = 10$ neurons respectively. On the other hand, STA-DNN(15-15-15) has three hidden layers with equal neurons $J_2 = J_3 = J_4 = 15$. Hence, the number of multiplications required is 4 $K_{\text{on}}J_2 + 2J_2^2$, besides 2 $K_{\text{on}} + 3J_2$ summations. Therefore, the total number of operations required by STA-DNN(15-15-15) is 82 K_{on}+2 K_d +450 multiplications, and 70 K_{on}+10 K_d +450 summations/subtractions.

Table 4 shows a detailed summary of the computational complexities for benchmarked channel estimation schemes. It can be deduced from Fig. 8 that the proposed STA-DNN(15-15-15) scheme achieves 55.74% computational complexity decrease compared with that of AE-DNN, while achieving considerable BER performance gain.

VII. CONCLUSION

In this article, we have investigated the challenging channel estimation in vehicular communications, due to its double dispersive nature. We have studied the IEEE 802.11p specifications, and we have proposed a deep learning based scheme that adheres to the standard structure. The STA-DNN proposed schemes combine the use of the classical STA channel estimation and DNN to capture more features of the time and frequency correlations between the channel samples, in order to ensure a more advanced tracking of the channel variations in time and frequency domains. Simulation results for different channel models of vehicular communications have demonstrated that the proposed STA-DNN significantly outperforms state-of-the art methods. In addition, we have shown that the optimized STA-DNN architectures achieve at least 55.74% computational complexity decrease compared with the recently proposed AE-DNN. As a future work, we will investigate how advanced, yet more complex, DNN architectures such as recurrent and convolutional neural networks, can be applied to channel estimation operation in vehicular communications.

A. CUBIC INTERPOLATION

Let $h[k] = h_Q[k] + jh_I[k]$ be the channel function. Assume, the channel is known at the subcarriers k_1 and k_2. In order to find the channel at the unknown subcarrier $k_1 < k < k_2$, we consider cubic interpolation per real and imaginary component. First, we change the variable $x = \frac{k-k_1}{k_2-k_1}$ and considering the real part, we define $f(x) = h_Q[k]$ such that:

$$f(x) = a_3x^3 + a_2x^2 + a_1x + a_0, \quad x \in [0, 1]. \tag{21}$$

In order to compute the coefficients $\{a_k\}$, four equations are required. Two equations are obtained by:

$$\begin{aligned} h_Q[k_1] &= f(0) = a_0, \\ h_Q[k_2] &= f(1) = a_3 + a_2 + a_1 + a_0, \end{aligned} \tag{22}$$

and another two equations by using the derivative:

$$\begin{aligned} f'(0) &= a_1, \\ f'(1) &= 3a_3 + 2a_2 + a_1. \end{aligned} \tag{23}$$

As a result, $a_0 = f(0)$ and $a_1 = f'(0)$. In addition, a_2 and a_3 can be computed from the equations:

$$\begin{aligned} a_3 + a_2 &= f(1) - f(0) - f'(0), \\ 3\,a_3 + 2a_2 &= f'(1) - f'(0). \end{aligned}$$

The solution is given by:

$$\begin{aligned} a_2 &= 3[f(1) - f(0)] - f'(1) - 2f'(0), \\ a_3 &= -2[f(1) - f(0)] + f'(1) + f'(0). \end{aligned}$$

As assuming that the channel is known at $k_0 \leq k_1$ and $k_3 \geq k_2$, the derivatives can be computed as:

$$\begin{aligned} f'(0) &= \frac{h_Q[k_2] - h_Q[k_0]}{k_2 - k_0}, \\ f'(1) &= \frac{h_Q[k_3] - h_Q[k_1]}{k_3 - k_1}. \end{aligned} \tag{24}$$

This guarantees that the polynomials in different segments are continuous. In the case where k_1 is a boundary subcarrier, then $k_0 = k_1$, similar for boundary k_2; $k_3 = k_1$.

Based on that, in the worst case, the computation efforts required to interpolate the real part of the channel gain requires 4 subtractions and 4 divisions to compute the derivatives. To compute the polynomial coefficients, we need 3 multiplications and 6 summations/subtractions. To compute the gain, first x is computed by 2 subtractions and 1 division. Afterwards, computing $f(x)$ is achieved by 5 multiplications and 3 summations. In total, it requires 13 multiplications/divisions and 15 summations/subtractions. Similar process is repeated for the imaginary part, and therefore, to interpolate the channel gain at one subcarrier between two known subcarriers, it is required to perform 26 multiplications/divisions and 30 summations/subtractions.

REFERENCES

[1] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments," in *Proc. IEEE Veh. Technol. Conf. (VTC Spring)*, May 2008, pp. 2036–2040.
[2] J. A. Fernandez, K. Borries, L. Cheng, B. V. K. V. Kumar, D. D. Stancil, and F. Bai, "Performance of the 802.11p physical layer in vehicle-to-vehicle environments," *IEEE Trans. Veh. Technol.*, vol. 61, no. 1, pp. 3–14, Jan. 2012.
[3] Z. Zhao, X. Cheng, M. Wen, B. Jiao, and C.-X. Wang, "Channel estimation schemes for IEEE 802.11p standard," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 4, pp. 38–49, Oct. 2013.
[4] Y.-K. Kim, J.-M. Oh, J.-H. Shin, and C. Mun, "Time and frequency domain channel estimation scheme for IEEE 802.11p," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1085–1090.
[5] J.-Y. Choi, K.-H. Yoo, and C. Mun, "MMSE channel estimation scheme using virtual pilot signal for IEEE 802.11p," *J. Korean Inst. Inf. Technol.*, vol. 27, no. 1, pp. 27–32, Jan. 2016.
[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org
[7] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.
[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, *arXiv:1710.05312*. [Online]. Available: https://arxiv.org/abs/1710.05312
[9] H. He, S. Jin, C.-K. Wen, F. Gao, G. Y. Li, and Z. Xu, "Model-driven deep learning for physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 77–83, Oct. 2019.
[10] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 93–99, Apr. 2019.
[11] S. Han, Y. Oh, and C. Song, "A deep learning based channel estimation scheme for IEEE 802.11p systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.
[12] Y. Yang, F. Gao, X. Ma, and S. Zhang, "Deep learning-based channel estimation for doubly selective fading channels," *IEEE Access*, vol. 7, pp. 36579–36589, 2019.
[13] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.

[14] S.-I. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, nos. 4–5, pp. 185–196, Jun. 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/092523129390006O
[15] 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, *arXiv:1807.06766*. [Online]. Available: https://arxiv.org/abs/1807.06766
[16] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, *arXiv:1706.05098*. [Online]. Available: https://arxiv.org/abs/1706.05098
[17] A. Gizzini, M. Chafii, A. Nimr, and G. Fettweis, "Enhancing least square channel estimation using deep learning," in *Proc. IEEE Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–6.
[18] C.-X. Wang, X. Cheng, and D. Laurenson, "Vehicle-to-vehicle channel modeling and measurements: Recent advances and future challenges," *IEEE Commun. Mag.*, vol. 47, no. 11, pp. 96–103, Nov. 2009.
[19] X. Cheng, Q. Yao, M. Wen, C.-X. Wang, L.-Y. Song, and B.-L. Jiao, "Wideband channel modeling and intercarrier interference cancellation for vehicle-to-vehicle communication systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 434–448, Sep. 2013.
[20] I. Sen and D. W. Matolak, "Vehicle–vehicle channel models for the 5-GHz band," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 235–245, Jun. 2008.
[21] G. Acosta-Marum and M. A. Ingram, "Six time- and frequency-selective empirical channel models for vehicular wireless LANs," *IEEE Veh. Technol. Mag.*, vol. 2, no. 4, pp. 4–11, Dec. 2007.

ABDUL KARIM GIZZINI (Graduate Student Member, IEEE) received the bachelor's (B.E.) degree in computer and communication engineering from the IUL Islamic University of Lebanon, in 2015, and the M.E. degree, in 2017. He is currently pursuing the Ph.D. degree in wireless communication engineering with the ETIS Laboratory (Joint Research Unit), National Council for Scientific Research (CNRS) (UMR 8051), ENSEA, CY Cergy Paris Université, France. His master's thesis was hosted by the Lebanese CNRS. His Ph.D. thesis focuses mainly on deep learning-based channel estimation in high-mobility vehicular scenarios.

MARWA CHAFII (Member, IEEE) received the master's degree in advanced wireless communication systems (SAR) and the Ph.D. degree in telecommunications from the CentraleSupélec, France, in 2013 and 2016, respectively. From 2014 to 2016, she has been a Visiting Researcher with the Poznan University of Technology, Poland, University of York, U.K., Yokohama National University, Japan, and the University of Oxford, U.K. She joined the Vodafone Chair Mobile Communication Systems, Technische Universitat Dresden, Germany, in February 2018, as a Research Group Leader. Since September 2018, she has been a research projects lead with Women in AI and has been an Associate Professor with ENSEA, France, where she holds the Chair of Excellence from the Paris Seine Initiative. Her research interests include signal processing for digital communications, advanced waveform design for wireless communications, and machine learning for communications. She received the 2018 Ph.D. Prize in signal, image and vision, France. She is the Vice-Chair of the IEEE ComSoc Emerging Technology Initiative on Machine Learning for Communications. She is also managing the Gender Committee of the AI4EU Community. She serves as an Associate Editor for the IEEE COMMUNICATIONS LETTERS.

AHMAD NIMR (Member, IEEE) received the Diploma degree in communication engineering, in 2004, and the Master of Science (M.Sc.) degree in communications and signal processing from TU Ilmenau, in 2014. He was a Software and Hardware Developer, from 2005 to 2011. He has been a member of the Vodafone Chair with the Technische Universitat Dresden, since October 2015. He is also involved in the design and implementation of real-time communication systems. His research interests include multicarrier waveforms and multiple access techniques. He received the Best Graduate Student Award for his Excellent Master's Grades.

GERHARD FETTWEIS (Fellow, IEEE) received the Ph.D. degree under H. Meyr's supervision from RWTH Aachen, in 1990. He has been a Vodafone Chair Professor with the Technische Universitat Dresden, since 1994, and has been heading the Barkhausen Institute, since 2018. He was with IBM Research, San Jose, CA, USA, for a period of one year. He moved to TCSI Inc., Berkeley, CA, USA. He coordinates the 5G Laboratory, Germany, and the two German Science Foundation (DFG) Centers with the Technische Universitat Dresden, namely CFAED and HAEC. His team has spun-out 16 start-ups and setup funded projects in volume of close to EUR 1/2 billion, Dresden. His research interests include wireless transmission and chip design for the wireless/IoT platforms, with 20 companies from Asia/Europe/U.S. sponsoring his research. He is a member of the German Academy of Sciences (Leopoldina), the German Academy of Engineering (acatech), the multiple IEEE recognitions, and the VDE Ring of Honor. He is the Co-Chairs of the IEEE 5G Initiative and helped organizing the IEEE conferences, most notably as a TPC Chair of ICC, in 2009, and TTM, in 2012. He is the General Chair of VTC Spring 2013 and DATE 2014.