electronics

MDPI

Article

Semi-Supervised Extreme Learning Machine Channel Estimator and Equalizer for Vehicle to Vehicle Communications

Eduardo Salazar [1], Cesar A. Azurdia-Meza [1,*], David Zabala-Blanco [2,3], Sandy Bolufé [1] and Ismael Soto [4]

[1] Department of Electrical Engineering, Universidad de Chile, Santiago 8370451, Chile; eduardo.salazar@ing.uchile.cl (E.S.); sbolufe@ing.uchile.cl (S.B.)
[2] Centro de Investigación de Estudios Avanzados del Maule (CIEAM), Vicerrectoría de Investigación y Postgrado, Universidad Católica del Maule, Talca 3466706, Chile; dzabala@ucm.cl
[3] Department of Computer Science and Industry, Faculty of Engineering Science, Universidad Católica del Maule, Talca 3480112, Chile
[4] Department of Electrical Engineering, University of Santiago de Chile, Santiago 91701234, Chile; ismael.soto@usach.cl
* Correspondence: cazurdia@ing.uchile.cl

Abstract: Wireless vehicular communications are a promising technology. Most applications related to vehicular communications aim to improve road safety and have special requirements concerning latency and reliability. The traditional channel estimation techniques used in the IEEE 802.11 standard do not properly perform over vehicular channels. This is because vehicular communications are subject to non-stationary, time-varying, frequency-selective wireless channels. Therefore, the main goal of this work is the introduction of a new channel estimation and equalization technique based on a Semi-supervised Extreme Learning Machine (SS-ELM) in order to address the harsh characteristics of the vehicular channel and improve the performance of the communication link. The performance of the proposed technique is compared with traditional estimators, as well as state-of-the-art machine-learning-based algorithms over an urban scenario setup in terms of bit error rate. The proposed SS-ELM scheme outperformed the extreme learning machine and the fully complex extreme learning machine algorithms for the evaluated scenarios. Compared to traditional techniques, the proposed SS-ELM scheme has a very similar performance. It is also observed that, although the SS-ELM scheme requires the largest operation time among the evaluated techniques, its execution time is still far away from the latency requirements specified by the standard for safety applications.

Keywords: channel estimation and equalizer; extreme learning machine; IEEE 802.11p amendment; semi-supervised learning; vehicular communications

check for updates

Citation: Salazar, E.; Azurdia-Meza, C.A.; Zabala-Blanco, D.; Bolufé, S.; Soto, I. Semi-Supervised Extreme Learning Machine Channel Estimator and Equalizer for Vehicle to Vehicle Communications. *Electronics* **2021**, *10*, 968. https://doi.org/10.3390/electronics10080968

Academic Editors: Michele Segata, Shankar Kathiresan, Deepak Gupta, Gyanendra Prasad Joshi, Chi-Hua Chen and Vicente García-Díaz

Received: 12 February 2021
Accepted: 9 April 2021
Published: 19 April 2021

1. Introduction

Vehicular communications are a promising technology for deploying next-generation intelligent transportation systems [1,2]. Vehicular Communication Systems (VCSs) rely on Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications subject to a time-varying, frequency-selective wireless channel for enabling road safety and traffic efficiency applications. Important efforts are being made towards the identification, definition, and characterization of applications and use cases for VCSs. For instance, the European Telecommunication Standards Institute (ETSI) has defined a basic set of vehicular applications to be considered as a reference for deployment and standardization [3]. ETSI has also specified the operation mode and communication requirements of vehicular applications, such as pre-crash sensing, lane change warning, and cooperative forward-collision warning [4,5]. These safety applications aim to decrease the probability of an accident by using early warnings, or even by taking control of the vehicle. On the contrary, vehicular applications such as adaptive cruise control and intersection management aim to decrease road congestion and improve traffic efficiency [3]. A challenge in VCSs is in satisfying the strict communication requirements of safety applications, which are vital for preserving human lives. Road safety applications have specified requirements concerning reliability and latency [4,5]; therefore, it is necessary for the the underlying structure of the vehicle network to guarantee these standards.

In VCSs, the particular propagation conditions of the vehicular environment make channel estimation and equalization a relevant challenge, especially because the high velocity of vehicles leads to variations in time and frequency [6]. In order to adapt to the harsh vehicular channel, the effective data rates of the IEEE 802.11p amendment were reduced to half in comparison to IEEE 802.11-based standards [7]. Traditional IEEE 802.11-based standards are the basis of Wireless Fidelity (WiFi) systems, which are commonly used in indoor scenarios and characterized by their low mobility [6,7]. Despite the particular propagation conditions of the vehicular environment and the high mobility of the vehicles, no modifications were made to the IEEE 802.11p amendment regarding the channel estimation techniques adopted by traditional IEEE 802.11-based standards. A well-known channel estimation technique is the Least Squares (LS), which estimates the impulse response of the channel at the start of a frame [8]. LS works well for typical indoor Orthogonal Frequency Division Multiplexing (OFDM) systems; however, its performance is degraded in vehicular environments because the channel changes rapidly over time, which is more critical for higher velocities [6]. Another channel estimation technique corresponds to the Spectral Temporal Averaging (STA) [9], which uses time and frequency dependencies to properly perform the estimation stage. STA unfortunately suffers a great performance impact in non Wide-Sense Stationary Uncorrelated Scattering (WSSUS) channels. Unlike STA, the Constructed Data Pilots (CDP) technique [8] does not suffer significant performance degradation on non-WSSUS channels, but it has poor performance at low Signal to Noise Ratios (SNRs).

Machine Learning (ML) provides a versatile set of tools to address the challenges of vehicular communications due to its ability to exploit available data and perform difficult tasks; for example, to predict channel parameters, to augment network throughput, and to help with congestion control or even autonomous driving [10]. There are several proposals [11–13] that use ML schemes to perform channel estimation and equalization, but these proposals do not consider the 802.11p amendment. In other words, these studies can not be extrapolated for VCSs because they do not consider the characteristics of the non-WSSUS mobile-to-mobile channel or the particular frame structure of the communication packet; therefore, the channel estimation must be performed differently to the 802.11p amendment mandates. In [11], Ye and Li used a Deep Learning (DL) approach characterized by off-line training to perform channel equalization and estimation for an OFDM-based system, by showing a performance improvement over the LS and the Minimum Mean-Square Error (MMSE) schemes. The work presented in [11] shows that DL has the potential to improve the estimation equalization of communication channel in OFDM-based schemes. However, when considering that V2V safety applications have latency limitations, and since the channel has rapid variations in the time and frequency domain, an off-line approach should be discarded, as the channel models used in the training stage might differ from the actual channel. In [12], Sattiraju et al. proposed an artificial neural network (ANN) scheme based on a convolutional neural network to perform channel estimation on a Cellular Vehicle-to-Everything (C-V2X) communication system. This technique performs well but it again requires an off-line learning stage. In [13], Liu et al. introduces the use of a fully Complex Extreme Learning Machine (C-ELM) neural network to perform channel estimation and equalization on OFDM signals, where an exponential channel model with eight stochastic fading delay paths based on the IEEE 802.11g standard (mainly, a frequency-selective channel) is considered. The results show that the proposed C-ELM algorithm outperforms other state-of-the-art techniques, such as MMSE, LS, and a deep neural network-based scheme, and has a better generalization ability by considering different subcarrier modulation formats and numbers of subcarriers. It is worth noting that only the hyper-parameters (mapping function and number of hidden neurons) of the C-ELM are reported; no theoretical or simulation justifications are presented.

In this work, as an initial exploration, we propose the use of an Extreme Learning Machine (ELM) channel estimator and equalizer technique, which is characterized by Semi-Supervised (SS) learning for vehicular communications. The proposed algorithm considers the special propagation conditions in the time and frequency domains of vehicular communication channels. The main contributions of this work can be summarized as follows:

- We propose a regularized ELM subject to SS learning as channel estimator and equalizer to enhance the performance of a representative IEEE 802.11p OFDM-based system in terms of Bit Error Rate (BER). To this end, we add a novel parameter denoted by δ in the Semi-supervised Extreme Learning Machine (SS-ELM) to address the time-domain fluctuations of the channel. Furthermore, a frequency-domain localized mapping is used to properly recover the OFDM signal, namely to address the frequency-selective channel;
- Taking the simulation framework of the evaluated system into account, we compute the sub-optimal SS-ELM hyper-parameters to diminish BER via extensive simulations. We also show that a supervised ELM does not improve the BER performance of a vehicular IEEE 802.11p system;
- We compare the proposed technique with current state-of-the-art machine-learning-based channel estimation schemes as well as traditional techniques in an urban environment for several values of Energy per Bit to Noise Power Spectral Density Ratios (E_b/N_0). The addressed techniques are also contrasted in terms of the required processing time.

The remainder of this article is organized as follows: Section 2 depicts the foundations and background of this work, namely the physical layer of the IEEE 802.11p amendment, details of the vehicular channel model used in this work, and the ELM neural network subject to supervised and SS learning. Section 3 carefully presents the novel and improved SS-ELM algorithm used to estimate and equalize the communication channel of an IEEE 802.11p-based V2V system. Section 4 depicts the evaluation scenario, describes the optimization process of the SS-ELM parameters, and compares the BER and execution time of the proposed scheme with other state-of-the art and traditional IEEE 802.11 channel estimation techniques. Finally, Section 5 contains concluding remarks.

2. Background

2.1. The IEEE 802.11p Standard

IEEE 802.11p is an amendment of the IEEE 802.11a standard [7]. Its purpose is to adapt existing indoor wireless communication systems to vehicular environments, where it is important to address the high mobility of the users. IEEE 802.11p uses OFDM as modulation format in the Physical Layer (PHY) and it operates at the 5.9 GHz radio frequency band. The transmitter sends several parallel data streams through orthogonal subcarriers, by improving the spectral efficiency and mitigating the severity of multi-path fading [7].

Table 1 shows the most important parameters defined on the IEEE 802.11p amendment. In general, the time domain parameters of this amendment are two times larger with respect to the IEEE 802.11a standard. For instance, the bandwidth is reduced from 20 MHz to 10 MHz, and the subcarrier frequency spacing is reduced by a factor of 2. As mentioned, these changes are made in order to increase the reliability of the transmission by increasing the duration of the OFDM symbols.

Table 1. IEEE 802.11p system parameters for a 10 MHz channel.

Parameter	Value
Number of data subcarriers (N_{SD})	48
Number of pilot subcarriers (N_{SP})	4
Number of subcarriers total (N_{ST})	52
Subcarrier frequency spacing (Δf)	0.15625 MHz
IFFT/FFT periods (T_{FFT})	6.4 µs $(1/\Delta f)$
PHY preamble duration $(T_{PREAMBLE})$	32 µs
Duration of the Signal BPSK-OFDM symbol (T_{SIGNAL})	8 µs
Training symbol guard interval duration (T_{GI})	3.2 µs
Symbol interval (T_{sym})	8 µs
Short training sequence duration (T_{SHORT})	16 µs
Long training sequence duration (T_{LONG})	16 µs

Figures 1 and 2 depict a general IEEE 802.11p transceiver. The transceiver is based on traditional OFDM architectures. For simplicity, the block diagrams of the transceiver are depicted in its base-band representation. At the transmitter side, the data are modulated and then passed through an Inverse Fast Fourier Transform (IFFT) block to get orthogonal signals. Afterwards, a Cyclic Prefix (CP) is added to diminish inter-symbol interference, and finally a temporal preamble is inserted [7]. The purpose of this preamble is to synchronize the signal (short training symbols) and estimate the channel (long training symbols). Nevertheless, the structure of the preamble is detailed at the end of this Subsection. Figure 2 shows that the transmitted signal is distorted by the wireless channel, which is time-varying and frequency-selective for V2V communications (refer to the next Section), and by Gaussian additive noise, which is typical to all communication system. At the receiver side, the first signal processing task consists of removing the CP, and then applying a Fast Fourier Transform (FFT) to the signal. Then, the channel is estimated, and the signal is equalized by exploiting the subcarriers used as pilot signals. Finally, the signal is demodulated, and the data can be recovered.

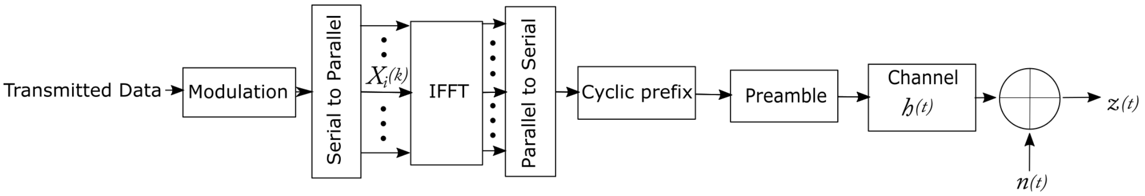

Figure 1. Block diagram of the OFDM transmitter.

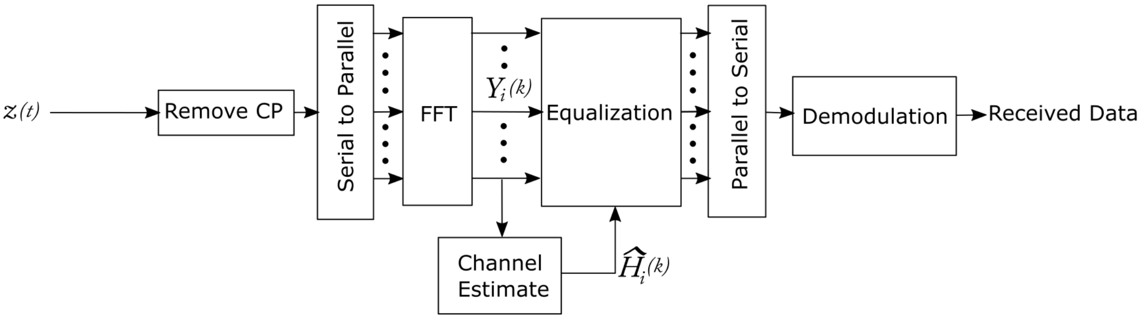

Figure 2. Block diagram of the OFDM receiver.

The signal received after the FFT operation is given as follows [6]

$$Y_i(k) = H_i(k)X_i(k) + N_i(k), i = \{1, ..., N_{symbol}\}, k = \{1, ..., M\}, \quad (1)$$

where i corresponds to the respective OFDM symbol, N_{symbol} denotes the total number of OFDM symbols, k is the corresponding subcarrier, and M represents the total number of subcarriers. Furthermore, $X_i(k)$ represents the transmitted signal that is obtained after the modulation in the transmitter block diagram, $Y_i(k)$ is the received signal after the FFT module at the receiver side, $H_i(k)$ is the Channel Frequency Response (CFR), and $N_i(k)$ is the Additive White Gaussian Noise (AWGN).

Based on [6], Figure 3 depicts the subcarrier distribution of the IEEE 802.11p standard. As mentioned previously, the bandwidth of the OFDM signal is given by 10 MHz and the radio frequency matches to 5.9 GHz. The data subcarriers are shown in green, while the pilots are shown in red and are uniformly distributed between the data subcarriers. Additionally, the null and Direct Current (DC) subcarriers are represented by black and blue colors, respectively. For the sake of simplicity, in our work, null and DC subcarriers are discarded.

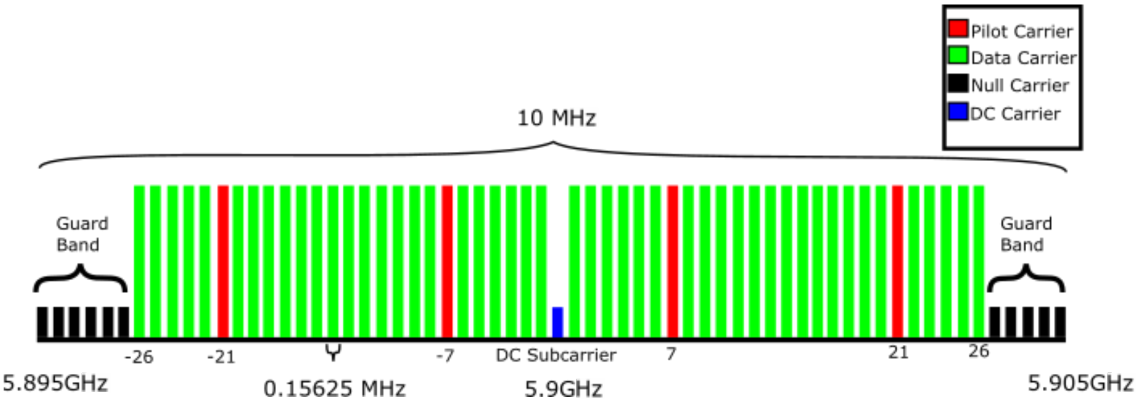

Figure 3. IEEE 802.11p subcarrier distribution.

In Figure 4, the IEEE 802.11p packet preamble structure is illustrated. It can be seen that the preamble has 10 short training symbols of 1.6 µs located at the beginning of the packet. These symbols are used to perform fine synchronization. In this work, the use of these symbols is discarded, since we assume a perfect synchronization to focus on the estimation/equalization issue. Furthermore, two long training symbols of 6.4 µs can be distinguished, which are used to perform the channel estimation. Since these symbols are found for each subcarrier of the OFDM signal, they are better suited to calculating the channel estimation than the pilot signals [7]. Note that a guard interval to avoid out-of-band interference also exists, whose duration is 3.2 µs.

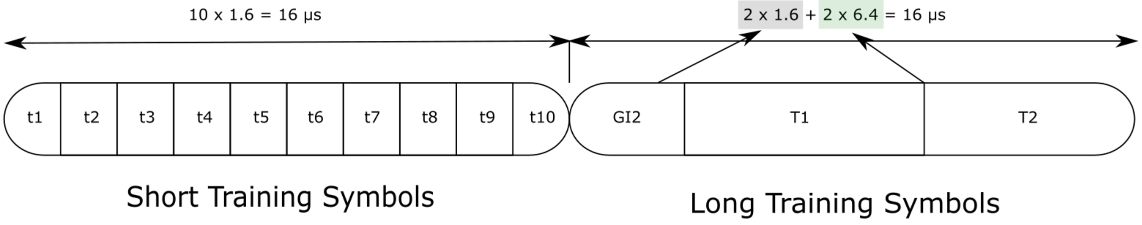

Figure 4. IEEE 802.11p packet preamble structure.

2.2. *Single Ring Geometrical Scattering Channel Model*

To test the channel estimation techniques, we need to define how to represent the communication channel. This representation needs to take most critical physical effects of the vehicular environment into consideration, and it also needs to be as close to reality as possible in order to properly evaluate the system performance.

Considering that the vehicular channel does not fulfill the WSSUS condition, the channel representation proposed in [14–16] is used in this work. The channel introduced in [14–16] is a statistical geometrical model for mobile-to-mobile systems. The scenario has a mobile transmitter with variable velocity and acceleration, as well as a receptor moving at a variable velocity and acceleration. The receiver is surrounded by interfering objects positioned in a circle whose center is the receptor itself. Therefore, the model depicts a realistic representation of a mobile to mobile system scenario. Furthermore, the model proposed in [14–16] is physically complete but mathematically simple in comparison with other non-WSSUS geometry-based statistical channel models.

In the model presented in Figure 5, V_T corresponds to the vector describing the velocity of the transmitter, V_R is the vector describing the velocity of the receptor, a_T represents the vector describing the acceleration of the transmitter, a_R denotes the vector describing the acceleration of the receptor, d comes to be the radius of the ring, and $l\text{-}th > IO$ corresponds to the $l\text{-}th$ interfering object. As a result, the channel transfer function is given as follows [14–16],

$$h(\tau,t) = \sum_{l=1}^{\mathcal{L}} g_l exp\{j[\theta_l - 2\pi f_c \tau_l(t)]\}\delta[t-\tau_l(t)]\Omega_{T_0}(t-t_0), \tag{2}$$

where f_c corresponds to the central frequency of the OFDM signal, whereas g_l, θ_l, and α_l^P are random variables. Specifically, g_l is a complex attenuation factor and follows a uniform distribution between $[0, \sqrt{\frac{1}{2\mathcal{L}}}]$, θ_l is the phase shift that follows a uniform distribution between $[0, 2\pi]$, and α_l^p is the angle between the $l\text{-}th > IO$ and the transmitter or receiver and follows a uniform distribution between $[0, 2\pi]$. However, $\Omega_{T_0}(t-t_0)$ is a windowing function equal to 1 if $t \in [t_0, t_0 + T_0]$ and 0 otherwise, where t_0 is the reference point where the observation begins and T_0 is the length of the observation window. This windowing function is used in order to ignore the large scale effects. The time-varying propagation delays of the multipath components are given as [14–16],

$$\tau_l(t) = \frac{d_l^T + d_l^R}{c} - \frac{t}{f_c}\left[f_l^S + \frac{\dot{f}_l^S(t)}{2}\right], \tag{3}$$

where f_l^S and $\dot{f}_l^S$ are the Doppler frequency shift caused by the vehicles' velocity and acceleration, respectively, and are defined in the following form,

$$f_l^S = f_{max}^T cos(\alpha_l^T - \gamma_T) + f_{max}^R cos(\alpha_l^R - \gamma_R), \tag{4}$$

$$\dot{f}_l^S(t) = \dot{f}_{max}^T(t) cos(\alpha_l^T - \beta_T) + \dot{f}_{max}^R(t) cos(\alpha_l^R - \beta_R), \tag{5}$$

where d_l^T, d_l^R are the distances of the transmitter and the receiver to the $l - th\ IO$, c is the speed of light, whereas γ_T and γ_R are the velocity vector angles of the transmitter and receiver velocity vector, respectively. At the same time, β_T and β_R are the acceleration vector angles of the transmitter and receiver, respectively. Finally, $f_{max}^P = V^P/\lambda_c$ and $\dot{f}_{max}^P(t) = a^P(t)/\lambda_c$ characterize the maximum Doppler shift caused by the velocity and acceleration of the mobiles, respectively, where λ_c stands for the wavelength of the transmitted signal and P must be replaced by R or T, which describes the receptor or the transmitter, respectively [6,17].

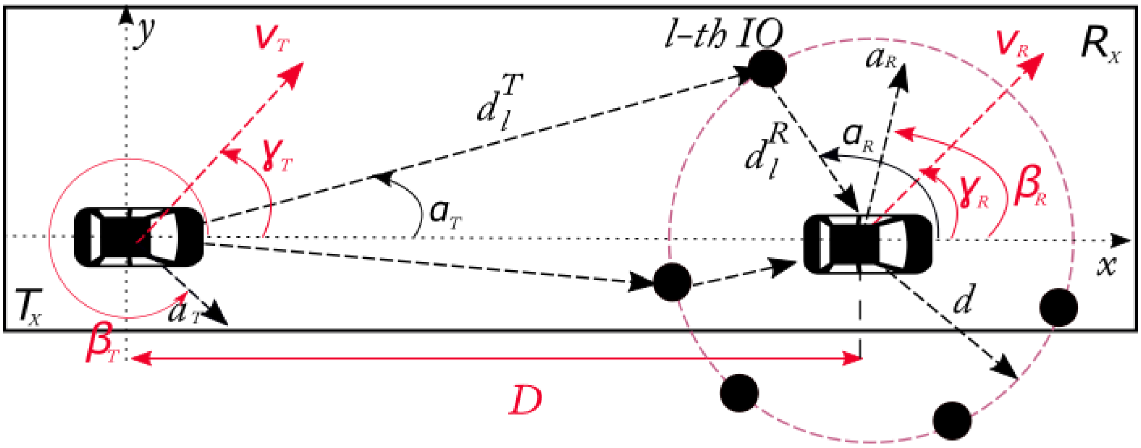

Figure 5. Single ring geometrical channel scattering model. The vehicle on the left side is the transmitter, while the vehicle on the right side represents the receiver.

2.3. Extreme Learning Machine

The ELM is a learning algorithm for feed-forward artificial neural networks characterized by a single hidden layer [18,19]. The ELM tends to minimize the training error thanks to the adoption of the smallest norm of the output weights. In addition, the ELM has a fast learning speed, as the only parameters that need to be optimized are the output weights between the hidden neurons and the output layer. The computational time required for training can be considered negligible in comparison with traditional feed-forward neural networks and support vector machines [20].

Generally, for a given training set with N_m instances $\{U, T\} = \{u_i, t_i\}_{i=1}^{N_m}$, where $u_i = [u_{i1}, u_{i2}, ..., u_{iD_i}] \in \mathbb{R}^{D_i}$ is an input sample and $t_i = [t_{i1}, t_{i2}, ..., t_{iD_o}] \in \mathbb{R}^{D_o}$ is its corresponding output (the input dimension D_i and output dimension D_o are not necessarily equal), the output of an standard single hidden layer feed forward network can be written in the following form [18,19],

$$\hat{t}_k = \sum_{i=1}^{n_h} \beta^i{}_{ELM} G[w_i \cdot u(k) + b_i], k = \{1, N_m\}, \tag{6}$$

where n_h is the number of hidden neurons, $\beta^i{}_{ELM} \in \mathbb{R}^{1xD_0}$ is the weight vector connecting the i-th neuron in the hidden layer with the output neurons, G is the activation function, $w_i \in \mathbb{R}^{N_m x1}$ is the weight vector describing connections of the i-th hidden neuron with the input neurons, and b_i is the i-th hidden neuron's threshold. In the ELM algorithm, input weights w_i and bias b_i are randomly generated according to a probability distribution and are related by the inner product operator $(\cdot)$.

The training process of the ANN can be expressed as a linear regression problem for the ELM algorithm, where a zero training error between the target output and the actual output $(\sum_{i=1}^{N_m} t_k - \hat{t}_k)$ can be determined as follows [18,19]

$$T = F\beta_{ELM}, \tag{7}$$

where the output matrix of the hidden layer acquires the form of

$$F = \begin{pmatrix} G(w_1 \cdot u_1 + b_1) & . & . & . & G(w_{n_h} \cdot u_1 + b_{n_h}) \\ . & . & & & . \\ . & & G(w_i \cdot u_j + b_i) & & . \\ . & & & . & . \\ G(w_1 \cdot u_{N_m} + b_1) & . & . & . & G(w_{n_h} \cdot u_{N_m} + b_{n_h}) \end{pmatrix}. \tag{8}$$

The optimal solution to the ELM can be computed as

$$\beta^*_{ELM} = F^{\dagger}T, \tag{9}$$

where $F^{\dagger}$ is the Moore-Penrose generalized inverse of F. If we consider zero training error, over-fitting could appear into the ELM algorithm. In order to avoid this problem, calculations are done by minimizing the final weights β_{ELM} with the inclusion or a regularization parameter. In other words, the mean square error is given by

$$\min_{\beta_{ELM},e} \frac{1}{2}||\beta_{ELM}||^2 + \frac{C}{2}\sum_{i=1}^{N_m}||e_i||^2,$$

$$\text{s.t } f(u_i)\beta_{ELM} = t_i^T - e_i^T, i = 1, ..., N_m, \tag{10}$$

where $f(u)$ represents an $output(row)$ vector of matrix in expression (8), $e_i \in \mathbb{R}^{D_o}$ is the error vector with respect to the *i-th* input sample, and C corresponds to the penalty coefficient, must be any real positive number. Finally, its solution when F has more rows than columns ($n_h > N_m$), can be written as follows [18,19]

$$\beta^*_{ELM} = \left(F^T F + \frac{I_{n_h}}{C}\right)^{-1} F^T T, \tag{11}$$

where I_{n_h} is an identity matrix with dimension n_h. On the other hand, in the case that F has less rows than columns ($n_h < N_M$), the previous solution can be written as follows

$$\beta^*_{ELM} = F^T \left(FF^T + \frac{I_{N_m}}{C}\right)^{-1} T, \tag{12}$$

where I_{N_m} is an identity matrix with dimension N_m. In summary, the training algorithm of the ELM is given in Algorithm 1.

Algorithm 1: ELM algorithm.

```
Inputs: The training set $\{U, T\} = \{u_i, t_i\}_{i=1}^{N_m}$
Output: The output weights $\beta_{ELM}$
1: Randomly generate the real value input weights and bias $w_i$, $b_i$
2: Model the hidden layer neurons using expression (6)
3: Based on expression (8), calculate the output matrix of the hidden layer F of
   ELM
4: Determine the output weights

if $n_h <= N_m$ then
    $\beta^*_{ELM} = (F^T F + \frac{I_{n_h}}{C})^{-1} F^T T$
else
    $\beta^*_{ELM} = F^T (FF^T + \frac{I_{N_m}}{C})^{-1} T$
end
```

2.4. *Semi-Supervised Extreme Learning Machine*

In a semi supervised setting, we have few labeled data and plenty of unlabeled data, which can be used to increase the perform of the system [18]. For this approach, we assume that the distribution of the label data is similar to the distribution of the unlabeled data and try to extract that the beneficial information. The SS-ELM algorithm is presented in Algorithm 2.

Algorithm 2: SS-ELM algorithm.

```
Inputs: Labeled data $\{U_l, T_l\} = \{u_i, t_i\}_{i=1}^{l}$, where $u_i = [u_{i1}, u_{i2}, ..., u_{iD_i}] \in \mathbb{R}^{D_i}$
  and $t_i = [t_{i1}, t_{i2}, ..., t_{iD_o}] \in \mathbb{R}^{D_o}$, and unlabeled data $U_\mu = \{u_i\}_{i=1}^{\mu}$,
where $l$ and $u$ are the numbers of labeled and unlabeled data, respectively
Output: The output weights $\beta_{ELM}$
1: Construct the graph Laplacian $L$ from both $U_l$ and $U_\mu$
2: Initiate an ELM network of $n_h$ hidden neurons with random input weights and
  biases, calculate the output matrix of the hidden neurons $F$
3: Select the hyper-parameters $C$ and $\lambda$
4: Find the output weights
if $n_h <= N_m$ then
    $\beta_{ELM} = (I_{n_h} + F^T CF + \lambda F^T LF)^{-1} F^T C\widetilde{T}$, where $\widetilde{T} \in I\!R^{(l+\mu)xD_o}$ is the
      augmented training target with its first $l$ rows equal to $T_l$ and the rest equal to
      0, $C \in I\!R^{(l+\mu)x(l+\mu)}$ is a diagonal matrix with $[C]_{ii} = C, i = 1, ..., l$ and the rest
      of values equal to 0
else
    $\beta_{ELM} = F^T (I_{l+\mu} + CFF^T + \lambda LFF^T)^{-1} C\widetilde{T}$, where $I_{l+\mu}$ is an identity matrix of
      dimension $l$ + $\mu$
end
```

It is important to notice that λ is known as the trade-off parameter, and indicates how important the unlabeled data are in the training stage. Obviously, if we set this parameter to zero, then the SS-ELM matches the standard ELM. The Laplacian graph is used to extract the geometry distribution information contained in the available data. As previously indicated, one of the assumptions of SS learning is that both labeled data U_l and unlabeled data U_μ are drawn from the same marginal distribution P_U. Another assumption is that the conditional probabilities of $P(t|u_1)$ and $P(t|u_2)$ should be similar if the two points u_1 and u_2 are close. With these assumptions, the manifold regularization term is usually minimized with the following function [20]

$$L_m = \frac{1}{2} \sum_{i,j} r_{i,j} ||P(t|u_i) - P(t|u_j)||^2, \tag{13}$$

where $r_{i,j}$ is the pair-wise similarity between two inputs and can be computed with the next expression

$$r_{i,j} = \begin{cases} \exp(-||u_i - u_j||^2/2\sigma^2) \text{ if } u_i, u_j \text{ are neighbors} \\ 0 \text{ otherwise} \end{cases}, \tag{14}$$

Expression (13) can be simplified as $L_m = Tr(\hat{T}^T L \hat{T})$, where $Tr(\cdot)$ is the trace of a matrix. The Laplacian graph can be calculated as $L = D - R$, where D is a diagonal matrix with its diagonal elements $D_{ii} = \sum_{j=1}^{l+\mu} r_{i,j}$, and the similarity matrix $R = [r_{i,j}]$. As mentioned, the SS-ELM setting incorporates the manifold regularization to incorporate the use of unlabeled data, thus improving the accuracy of the predictions. The expressions for β_{ELM} shown on Algorithm 2 are obtained when modifying the problem on Equation (10) as follows,

$$\min_{\beta} \frac{1}{2} ||\beta_{ELM}||^2 + \frac{1}{2} \sum_{i=1}^{N_m} C_i ||e_i||^2 + \frac{\lambda}{2} Tr(\beta_{ELM}^T F^T LF \beta_{ELM}),$$

$$\text{s.t } f(u_i)\beta_{ELM} = t_i^T - e_i^T, i = 1, ..., N_m. \tag{15}$$

In the context of V2V communications, this algorithm might offer a better performance, as it could, for example, learn from the time and frequency variations present in the propagation scenario. Even though it may learn from these variations, other algorithms that use these dependencies have a large performance degradation [6,8] in the presence of rougher propagation scenarios, like the STA algorithm.

3. Proposed SS-ELM Equalizer

Initially, we used the original ELM with supervised learning (as a reference case). To this end, we performed the training step by employing the two long training symbols as input training data and by following the steps of the Algorithm 1, then the frame was equalized by feeding each data subcarrier to the ELM. As any equalizer, this procedure is done frame by frame. Based on the poor BER resulting from this type of ELM (as will be depicted in the next Section), for this manuscript, we explored the SS learning paradigm for the ELM algorithm along with localized mapping. In the following paragraphs, we will define and explain the proposed SS-ELM algorithm.

The proposed SS-ELM algorithm is based on Algorithm 2 and, as a common machine learning approach, has two stages: training and testing. The first one consists of the training phase, in which we use the pilots and the two long training symbols to calculate the internal parameters of the ELM (β_{ELM}). Notice that the weights and biases between the input and hidden layers are arbitrarily generated (a main characteristic of the ELM algorithm as mentioned in Section 2.3) based on the uniform distribution on $[-1{,}1]$ [21,22]. Using these parameters, we then perform the evaluation phase. Here, we equalize the constellation symbols and, afterwards, calculate the BER metric by using the standard demodulation.

The time-frequency fluctuations in the channel can be seen in Equations (2) and (3) where $\tau(t)$ represents the general time delay variations. Instead, the several parameters on Equations (4) and (5) show the frequency oscillations associated with the Doppler effect. Consequently, if we feed all the data directly to the model, we will not properly address this problem (as observed in the next Section). Consequently, we perform the training and equalization by considering the frequency and time domains for the SS-ELM algorithm.

The basic ELM architecture consists of two real inputs $\mathbb{R}\{Y_i(k)\}$ and $\mathbb{I}\{Y_i(k)\}$, one for the real part of the constellation symbol and other for its imaginary part, n_h hidden neurons, and two neurons on the output, one for each part of the constellation symbol. For the sake of simplicity, we will refer to the inputs as $Y_i(k)$ and the outputs as $\hat{Y}_i(k)$. First of all, by considering that the channel changes rapidly in the frequency domain, the information from the pilot carriers is used in a localized manner, as shown in Figure 6.

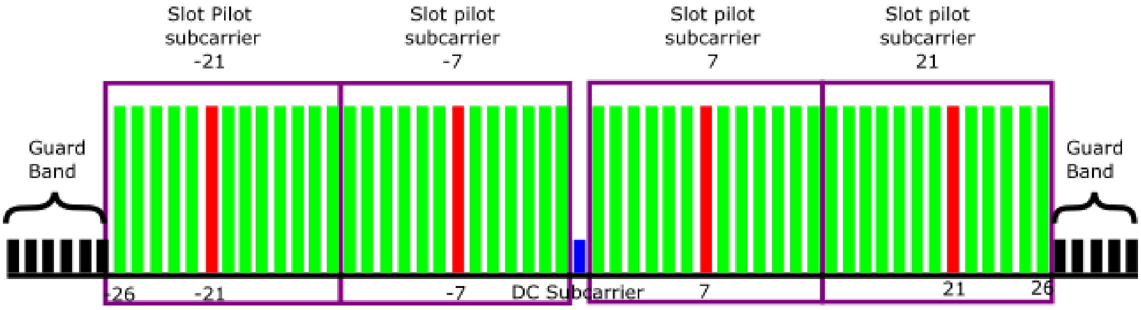

Figure 6. Subcarrier distribution slots.

Therefore, four localized equalization processes following the IEEE 802.11p standard are done at the same time, i.e., one for each pilot subcarrier. We define the slot signal as follows

$$X_i^p(k), Y_i^p(k) = X_i(k), Y_i(k)_{k=p}^{p+q}, \tag{16}$$

where i is the number of OFDM symbol, k denotes the subcarrier number, p represents the pilot corresponding to the slot being process, and q comes to be the length of the frequency slot.

With the localized mapping, there are 12 data carriers for each pilot. This number of data carriers was adopted because it corresponds to the relationship N_{SD}/N_{SP}, where N_{SD} and N_{SP} are given by the parameters exposed in Table 1. Consequently, we take advantage of the frequency dependency of the channel in order to diminish the impact of the frequency variations in the channel in the system performance, since the term λ in Equations (4) and (5) is the closest to the pilot subcarrier term. Then, for each slot, the process is performed symbol by symbol in the time domain by considering the definition given in Equation (1). This is done in order to track the time fluctuations in the channel. For illustration purposes, Figure 6 also represents how the subcarriers are distributed for each slot.

In addition, the block diagram of the equalization process is presented in Figure 7 for clarification purposes. Obviously, it begins when the FFT module is performed and ends with the constellation symbols to be demapped. As can be seen, a pre-equalization with the LS method is done right after the Fourier-based block. The reason behind this stage is because the channel estimation which resulted from the LS scheme for the time domain OFDM symbols is fairly accurate, and results in a good initial mitigation of the channel effects with relatively low computational impact [13].

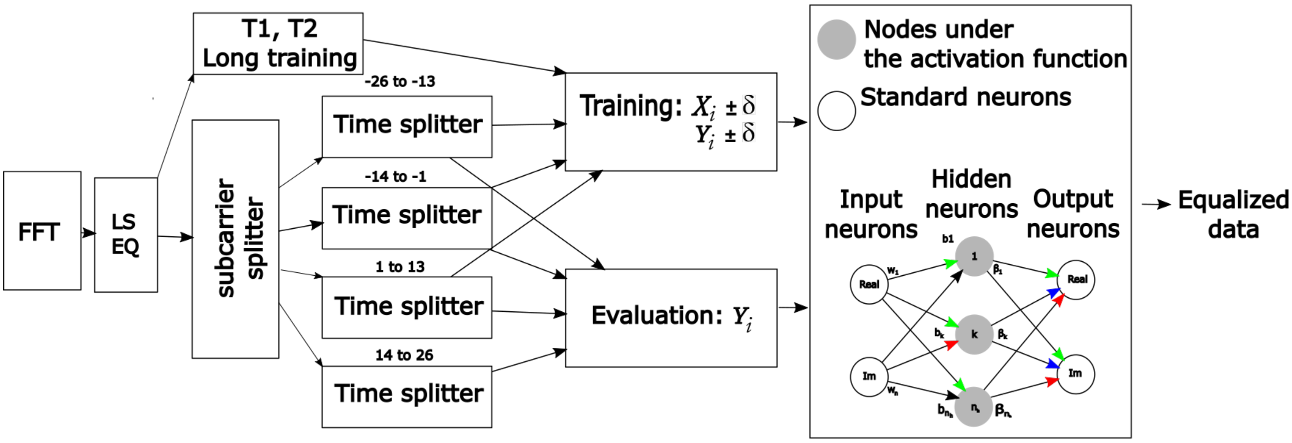

Figure 7. Block diagram of the equalization process.

In addition, an extra design parameter (δ) is introduced to strengthen the learning phase in the time domain. This determines how many pilots in the time domain are used for each training step of each slot. This parameter is important as it uses the time dependency of the channel to improve the estimation/equalization processes by considering that the vehicular environment changes rapidly over the time domain. According to our observations, and as is expected in this scenario, a high value of δ might have a negative impact on the system performance. Evidently, the proposed scheme (Figure 7) replaces the channel estimation and equalization blocks seen in Figure 2, by combining them into a single operation after the FFT module. To depict the general operation of the novel SS-ELM equalizer step by step, the SS-ELM training and testing is presented in Algorithm 3.

It is important to consider that the actual value of unlabeled data used for the training of each data frame is the multiplication of μ with the number OFDM symbols $N_{symbols}$ and the number of slots. Even though the values that are shown in the next Section are small, these values do not directly represent the total amount of data used for the unsupervised learning step of the SS-ELM scheme. Finally, each slot can be processed parallel to each other, by allowing the reduction in the execution time needed to process a full data frame. In fact, this procedure is done in Section 4.4 to estimate the complexity of the proposal.

Algorithm 3: SS-ELM training and equalization.

```
$N_{symbol}$ is the same in Equation (1) and depicts the total number of OFDM symbols
Input:
Training stage: Labeled data, $U_l = \{Y_{T1}, Y_{T2}, P^R\}$, $T_l = \{X_{T1}, X_{T2}, P^T\}$ where
  $X_{T1}, Y_{T1}$ and $X_{T2}, Y_{T2}$ are the two long training symbols and $P^T, P^R$, are the pilot
  vector transmitted and received respectively constructed as:
  $\{P^T, P^R\} = \{Y_i(p), X_i(p)\}_{t=i-\delta/2}^{i+\delta/2}$
Unlabeled data, $U_\mu = \{Y_i^p(k)\}_{k=1}^{\mu}$
Testing stage: Corrupted data, $Y_i(k)$
Output:
Training stage: $\beta_{ELM}$
Testing stage: $Y_i\hat{(}k)$
for $i = 1$ to $N_{symbol}$ do
    Training stage: train the SS-ELM computing $\beta_{ELM}$ based on Algorithm 2:
    1: Construct the graph Laplacian $L$ from both $U_l$ and $U_\mu$
    2: Initiate an ELM network of $n_h$ hidden neurons with random input weights
      and biases, calculate the output matrix of the hidden neurons $F$
    3: Select the hyper-parameters $C$ and $\lambda$
    4: Find the output weights
    if $n_h <= N_m$ then
        $\beta_{ELM} = (I_{n_h} + F^T CF + \lambda F^T LF)^{-1} F^T C\widetilde{T}$
    else
        $\beta_{ELM} = F^T (I_{l+\mu} + CFF^T + \lambda LFF^T)^{-1} C\widetilde{T}$
    end
    Testing: The equalization step for the $i^{th}$ OFDM symbol is perform by feeding
      the slot signal $Y_i^p$ to the model.
    1: Compute the output of the hidden layer $F_{signal}$
    2: Compute the output of the ELM with $F_{signal}\beta_{ELM}$
end
```

4. Simulation Results and Discussions

To further validate our proposal (SS-ELM), we evaluated other state-of-the-art schemes; these techniques are LS, STA, CDP, and C-ELM [13]. For the case of the LS, STA, and CDP schemes, Zero Forcing (ZF) is used as the channel equalize [23]. For comparison results, note that among the several ML-based approaches reported in the literature, we included the C-ELM method, as it has shown the best BER performances without computational cost, not only for communications through optical fiber [21,22] but also for advanced wireless communication systems [13,24–27]. Of course, these works are not focused on IEEE 802.11p-based V2V communication systems, which present a harsh non-stationary, time-varying, frequency-selective channel; refer to Section 2. On the other hand, it is worth noting that our proposal works with strictly real data (a single constellation symbol is divided into its quadrature and in-phase parts) and consists of a semi-supervised machine learning (a combination of supervised and unsupervised learning). While the C-ELM [13] represents fully complex neural networks under only supervised learning. Furthermore, the C-ELM neural network needs a bounded and differentiable activation function defined in the complex plane and cannot follow semi-supervised training, which limits its generalization ability. Each one of these models is tested in two different system setups. Further details regarding the system setups are given in the subsequent lines. A Monte Carlo simulation of 21 iterations is used to ensure statistical regularity. A total of 10.000 data frames are transmitted in each iteration, and E_b/N_0 ranging between 0 to 20 dB that correspond to real-life scenarios [28] are used in the evaluations. The parameters used to perform the simulations are shown in Tables 1 and 2. Specifically, Table 1 illustrates the IEEE 802.11p system parameters for a 10 MHz channel, whereas Table 2 shows the evaluated system configurations.

Table 2. Simulation parameters.

Parameter	Configuration 1	Configuration 2
Carrier Frequency (f_c)	5.9 GHz	5.9 GHz
Bandwidth (B)	10 MHz	10 MHz
Modulation	BPSK	BPSK
Number of OFDM symbols per package (L)	128	128
Transmitter velocity (V_T)	**40 km/h**	**20 km/h**
Receiver velocity (V_R)	**40 km/h**	**20 km/h**
Transmitter movement angle (γ_T)	**105°**	**10°**
Receiver movement angle (γ_R)	**70°**	**70°**
Transmitter acceleration angle (β_T)	**105°**	**15°**
Receiver acceleration angle (β_R)	**250°**	**70°**
Initial distance (D)	**300 m**	**100 m**
Radius of the ring (d)	30 m	30 m

Both system configurations simulate an urban environment, with speeds lower than 50 km/h. The first configuration depicts a scenario in which the transmitter is accelerating and changing lanes in preparation to overtake the receiver [29], whereas the second configuration is a scenario in which the receiver is taking an intersection while the transmitter remains in the same lane. For clarification purposes, the parameters that originate the overtake and intersection scenarios (the movement angles, acceleration angles, and velocities of the transmitter and receiver, and initial distance) are shown in bold in Table 2. In this sense, these parameters are highlighted with a red color in Figure 5, where they can also be easily distinguished. Furthermore, the simulations are done with Binary Phase Shift Keying (BPSK). This is because, with BPSK, we have the slowest data rates, as well as the most robust transmissions, which are critical in safety applications. To conclude this part, in order to find the sub-optimal hyper-parameters of the SS-ELM and compare the performance of the different models, we will present the results as a function of the BER and E_b/N_0.

4.1. Numerical Optimization of the SS-ELM Hyper-Parameters

The proposed scheme has several parameters that need to be optimized, and some of these parameters have a correlation between them, so they need to be optimized together. Throughout the paper, we adopt the sigmoid mapping function, since it has reported an excellent generalization ability [18,22]. We initially optimized the regularization parameter C and the number of hidden neurons n_h, by considering $\lambda = 0$ and $\mu = 0$. This means that the first optimization is basically the same for a supervised ELM. To select the best parameters, we use contour plots, which are obtained for four E_b/N_0 values: 5, 10, 15, and 20 dB for the two channel configurations presented in Table 2. In Figures 8 and 9, each point of the contour plot corresponds to the representative BER value obtained by following the Monte Carlo simulation explained at the beginning of this Section. Then, to select the sub-optimal values of the regularization parameter and number of hidden neurons that minimize the system performance, we use visual inspection. Namely, the areas with the best performance (blue zones) and overlap among most of the figures are chosen. All sub-figures in Figures 8 and 9 are considered in order to obtain a BER invariant to the movement angles, acceleration angles, and velocities of the transmitter and receiver. As expected, these plots also verify that the BER decreases as the relationship E_b/N_0 increases.

In Figures 8 and 9, we can see that there is a trade-off between the optimized parameters and the ELM predictability. When the number of hidden neurons is increased, it is possible to decrease the value of the regularization parameter C without compromising the BER performance. In general, the system performance severally decreases if the number of hidden neurons does not exceed the value of 24, specially for regularization parameters lower than 10^0. Furthermore, if both simulation configurations are compared, it can be observed that the BER metric decreases from configuration 1 to 2 and that the area of the optimal parameters increases from configuration 1 to 2. Based on the previous observations, we selected $C = 10^1$ and $n_h = 64$, namely, a pair of hyper-parameters that are inside the optimal area of both configurations. This adoption also considers that the ELM architecture is kept simple, namely, a minimum number of hidden neurons is selected. It is important to notice that the apparent optimal area increases from configuration 2 to 1, which indicates that, for bigger velocity values, the optimal area may be bigger and will probably contain the ones shown above.

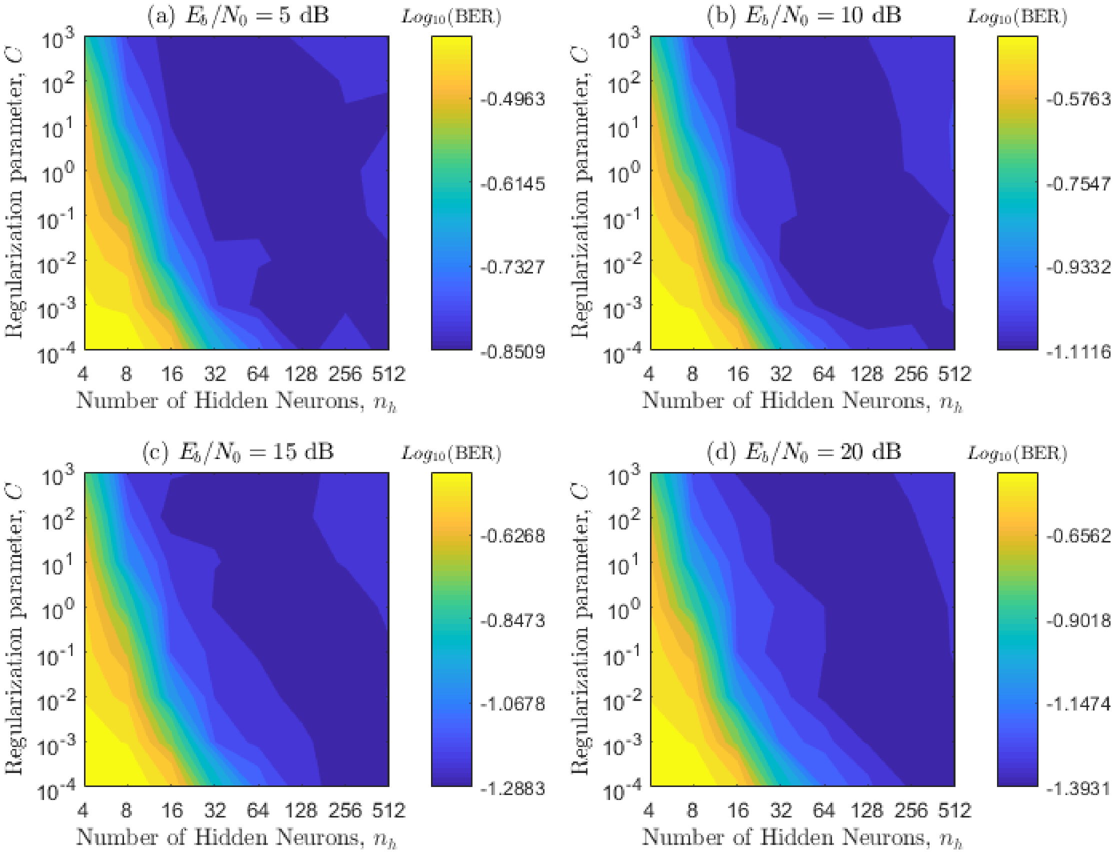

Figure 8. BER countour plot in terms of the regularization parameter C, number of hidden neurons, and different E_b/N_0 values for system configuration 1.

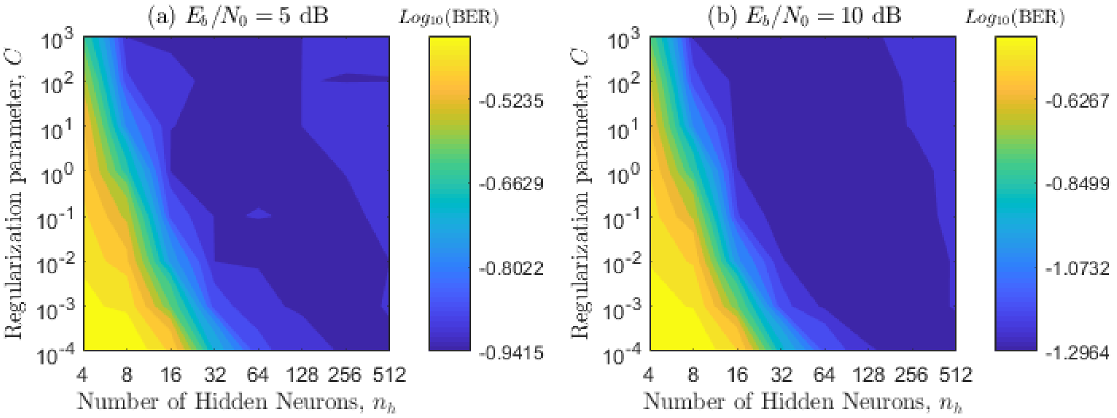

Figure 9. *Cont.*

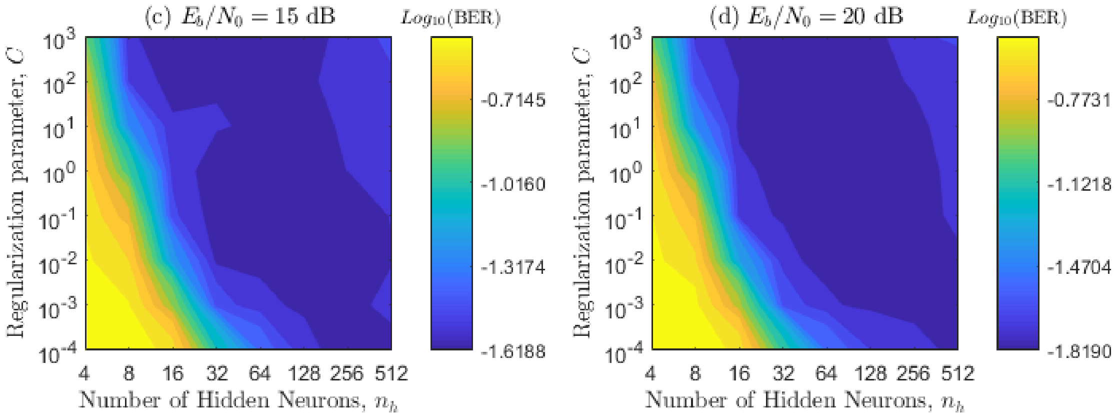

Figure 9. BER countour plot in terms of the regularization parameter C, number of hidden neurons, and different E_b/N_0 values for system configuration 2.

Once the values of C and n_h were optimized, we proceed to optimize the parameters λ and μ simultaneously for the SS-ELM training/testing, in the same way as the previous ones for the standard ELM. Remember that λ represents the trade-off parameter between the supervised and no-supervised learning and μ proportionally represents the number of unlabeled samples; refer to the end of Section 3. Note that, for these results, we use a linear representation of the BER instead of a logarithmic one because the results are closer to each other and the resulting areas are smaller. Looking at the results of Figures 10 and 11, it can bee seen that the optimal areas are relatively smaller when compared with the ones shown in Figures 8 and 9. This means that there is a smaller range of values that can be used without compromising the BER performance. Furthermore, the values obtained are more sensitive to different channel configurations. It can also be seen in Figures 10 and 11 that, when the value of the parameter μ is higher than 7, the system performs poorly. This result indicates that the proposed SS-ELM algorithm suffers over-fitting, or that there is too much noise added to the training process with the unlabeled data inclusion. When considering the parameter λ, it can be seen that the better values tend to be bigger than 10^0 for the first configuration, and bigger than 10^1 for the second configuration. This behavior means that SS-ELM benefits from bigger values of λ, meaning that the non-labeled data have a big impact on a one-to-one comparison with the labeled data. Then, the selected parameters are $\lambda = 100$ and $\mu = 6$, because this combination shows the best overall results for the simulated configurations. Finally, considering that the total value of unlabeled data used to train each data frame is $\mu = 6$ times $N_{symbols} = 128$ times 4, the total value used is 3072 symbols; when we compare this number with the total number of labeled data available for the training (616 symbols), we can observe that we use approximately five times more unlabeled data than labeled data for the training procedure. Hence, we take advantage of the semi-supervised paradigm, where we have few labeled data and plenty of unlabeled data. Transforming this to the studied communication scenario, we have only four pilot subcarriers to equalize 52 data subcarriers, but we use unlabeled data to improve the performance.

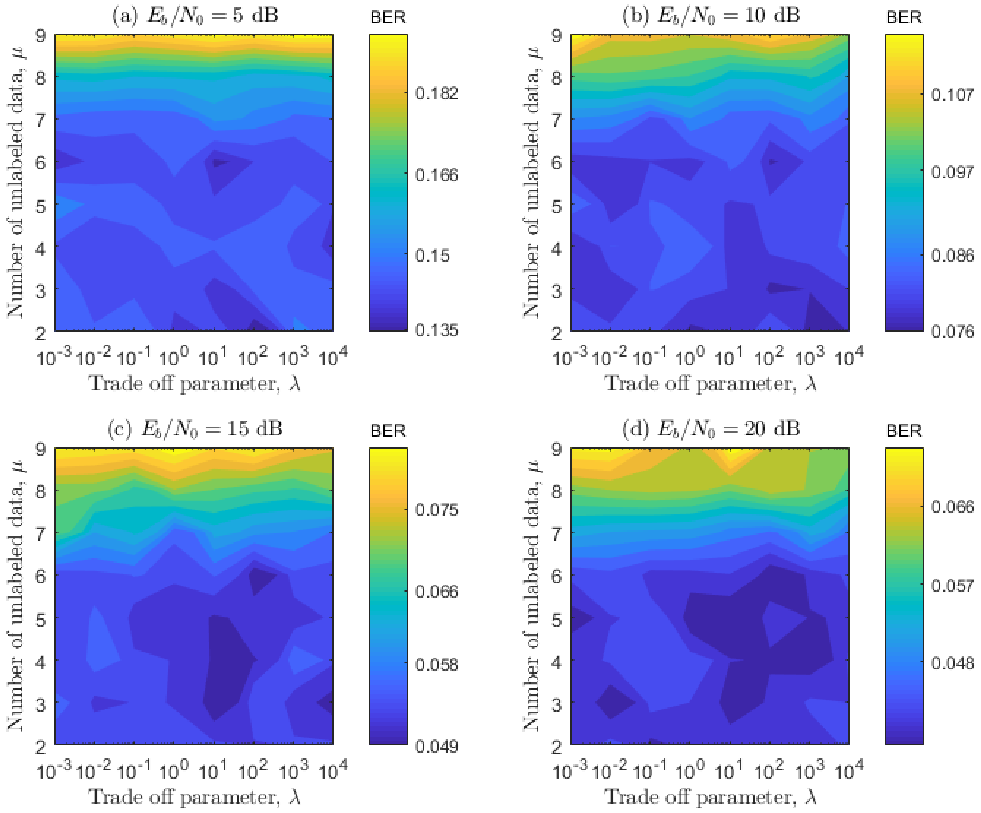

Figure 10. BER contour plot of λ and μ for different E_b/N_0 values and system configuration 1.

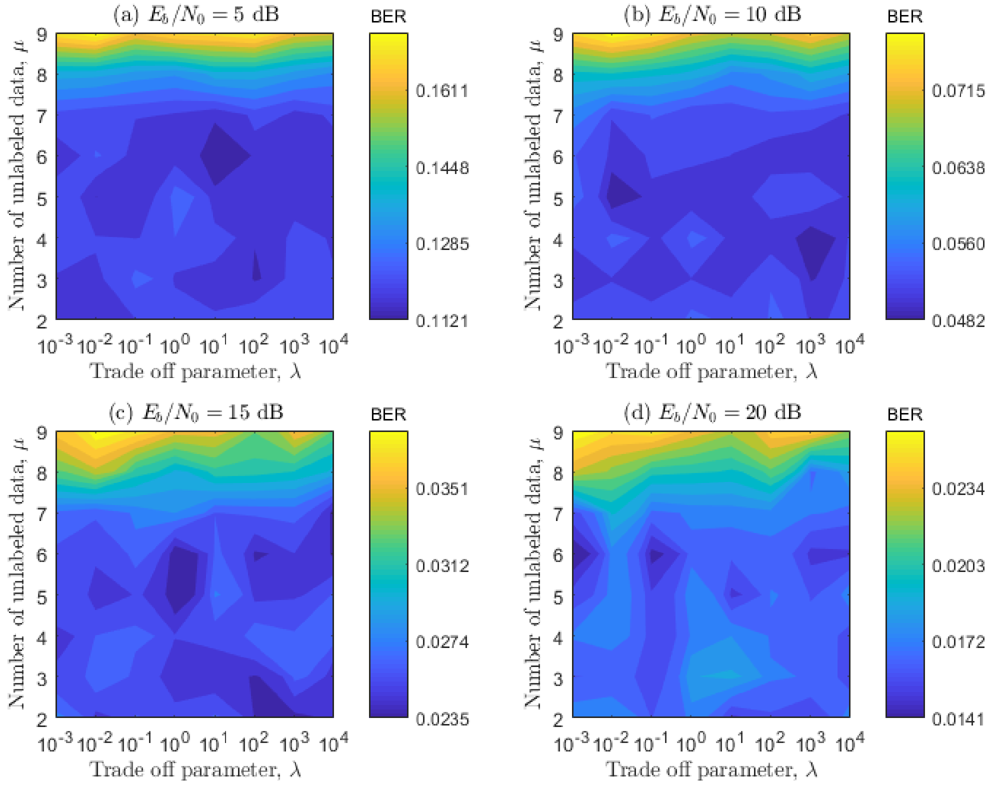

Figure 11. BER contour plot of λ and μ for different E_b/N_0 values and system configuration 2.

4.2. Impact of the δ Parameter on the BER Metric

In this Subsection, we analyze the results of the proposed SS-ELM for different values of the δ parameter. As mentioned in Section 3, the δ parameter is oriented to decrease the time-variant channel effects in the system performance. To this end, it relates to the number of pilot subcarriers employed in the learning phase of each time slot.

In Figures 12 and 13, it can be seen the BER resulted from the SS-ELM scheme against E_b/N_0 for diverse δ parameters by considering configurations 1 and 2, respectively. The LS results are also displayed as a reference BER curve. In the following Section, our proposal will be carefully compared with the bench-marking estimators/equalizers. There is a direct relation between the value of the parameter δ and the BE performance. We can also see that, for system configuration 1, there is a clear improvement, with values of δ near to 12; this enhancement is not present for configuration 2. Therefore, it is possible to say that the impact of the parameter δ in the performance of the systems is higher for a rougher system setting, such as the configuration 2. Finally, there is a clear performance improvement for the close interval between 7 and 12 of the parameter δ for both of the evaluated configurations. In conclusion, the best overall results are obtained with a SS-ELM characterized by $\delta = 16$, by relaxing the E_b/N_0 requirement with respect to the LS technique.

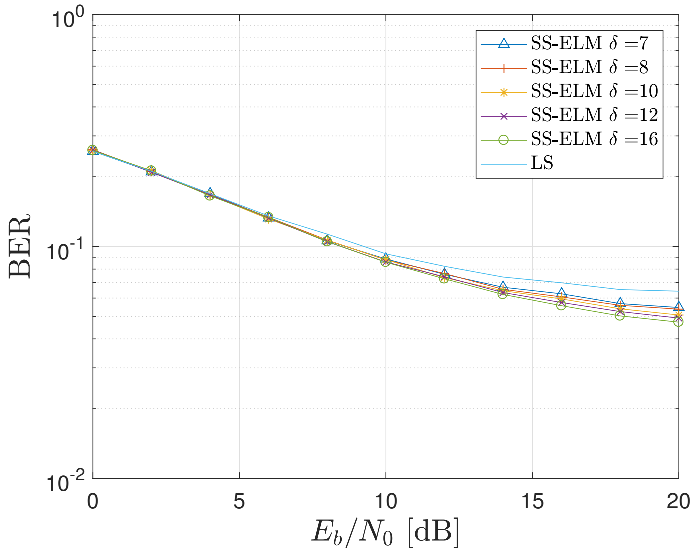

Figure 12. BER of the proposed SS-ELM scheme with δ as parameter and the LS algorithm for system configuration 1.

4.3. Performance Comparison

In this Subsection, we compare the results between the LS, STA, CDP, traditional ELM, C-ELM [13] techniques, and our proposal, which is based on a SS-ELM neural network. For the standard ELM, the regularization parameter and the number of hidden neurons are the same as the SS-ELM, namely $C = 10$ and $n_h = 64$. Furthermore, in our proposal, the rest of the hyper-parameters are the following: $\lambda = 100$, $\mu = 6$, and 16. Remember that these adoptions are done in order to minimize the system performance (see Section 4.2). According to [13], we adopt 512 hidden neurons and the arcsinh mapping function for the C-ELM.

For the first and second configuration, Figures 14 and 15 depict the BER as a function of E_b/N_0 for the different techniques evaluated. It can be seen that ELM, C-ELM, SS-ELM, and LS methods tend to have the same BER, regardless of the scenario. On the other hand, for the first and second configurations, the STA method has the worst and best BER values, respectively. The CDP method outperforms the other techniques for high SNRs only for the first configuration. These observations mean that STA and CDP do not perform uniformly for different channel configurations; consequently, these techniques are not feasible for V2V communications. Among the rest of the evaluated techniques, our proposal shows a uniform and slightly better performance for the evaluated scenarios, especially for higher SNRs. For instance, given a threshold of $BER = 10^{-1}$ in the first configuration, the SS-ELM allows a E_b/N_0 greater than 8 dB. Meanwhile, the E_b/N_0 requirement corresponds to 9, 10, and 14 dB in the case where the LS, ELM, and C-ELM approaches are used, respectively. It is worth noting that observations for SNRs greater than 20 dB can be discarded, since these values rarely occur in real settings [28]. Furthermore, it is worth noting that the vehicular communication channel evaluated in this manuscript is extremely harsh; therefore, none of the evaluated schemes achieved the BER threshold given by the standard and equal to 10^{-3} [30,31].

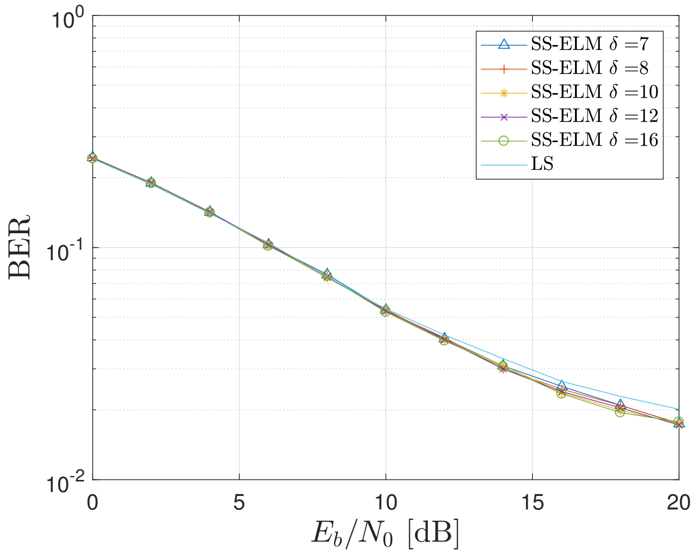

Figure 13. BER of the proposed SS-ELM scheme with δ as parameter and the LS algorithm for system configuration 2.

The superiority reported in our proposal can be explained by the use of the ELM neural network along with the exploitation of the semi-supervised machine learning scheme. With respect to conventional neural network learning algorithms, the ELM algorithm has demonstrated a better generalization performance in the presence of linear and no-linear distortions [13]. Its generalization ability comes from the parameter neurons between adjacent layers are found by reaching both the smallest training error and the smallest norm of output weights. On the other hand, semi-supervised learning has been applied to several classification and regression tasks with excellent results, in which both the labeled and unlabeled data are used to improve accuracy over supervised approaches. It occurs especially when insufficient training information is available [20]. The reason behind this is that unlabeled data naturally provide valuable information for exploring the data structure in the input space. To conclude this explanation, it is well known that the interactions among different types of impairments (such as the harsh time-varying, frequency-selective channel considered in this manuscript) are not effectively considered and handled for non-based ML-based schemes [6]. For instance, STA and CDP techniques were not designed considering the characteristics of the vehicular channel. In regular channels, STA uses frequency dependencies to improve the performance, but in vehicular communications, the channel varies rapidly in frequency, explaining the poor results for rougher channels. Meanwhile, CDP suffers greatly from low E_b/N_0 because it uses the channel estimation iteratively to equalize the channel, so, in high noise environments, this estimation is contaminated, decreasing the performance.

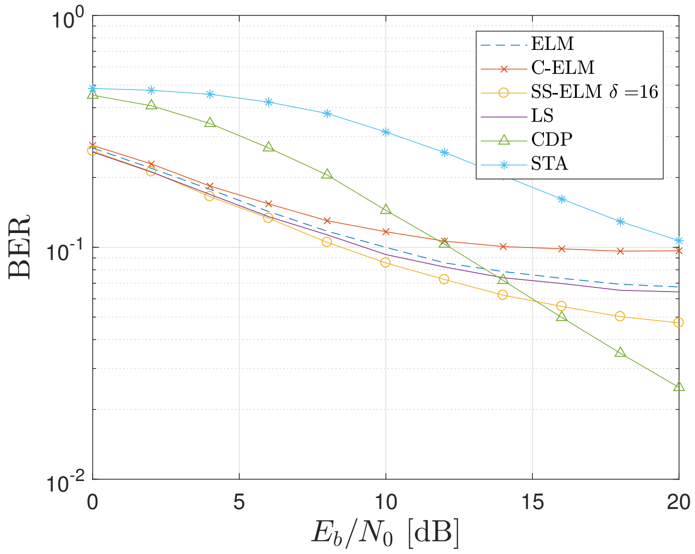

Figure 14. BER as a function of E_b/N_0 for different evaluated techniques and system configuration 1.

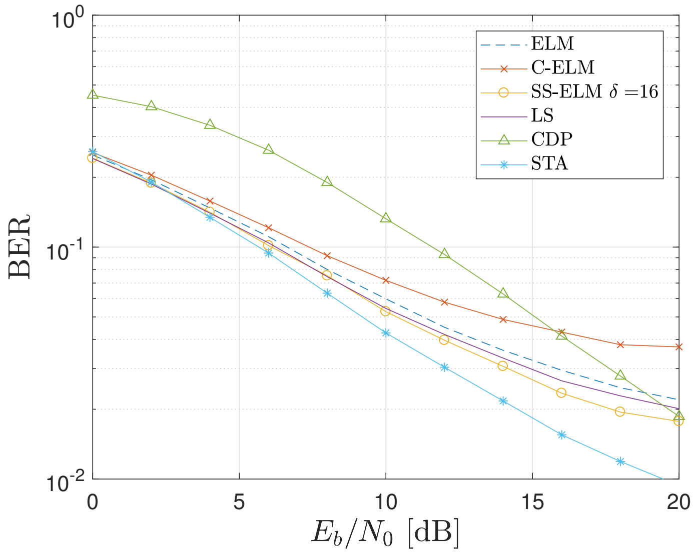

Figure 15. BER as a function of E_b/N_0 for different evaluated techniques and system configuration 2.

4.4. Execution Time Analysis

In this Subsection, we evaluate the performance of the addressed techniques in terms of execution time, which is the time required by the algorithms to complete the processing (from the FFT module to the parallel to serial converter, see Figure 2). As the execution time is hardware/software dependent, Table 3 is presented to show the hardware configuration used in the simulations. Regarding software features, Matlab R2017a has been used to perform the evaluation. To measure the representative interval times, we follow the Monte Carlo simulation explained at the beginning of Section 4 by fixing the channel model parameters at arbitrary values, since these do not interfere in the estimation of the processing times. Because each of the frequency slots shown in Figure 6 has a localized submapping estimation and equalization, the slots can be processed independently from each other, thus decreasing the execution time of the proposed SS-ELM scheme by a factor of 4 due to the number of pilots in an OFDM symbol (refer to Figure 6). The results of this experiment are depicted in Table 4 under the name of parallel SS-ELM.

Table 3. Computational system specifications.

Component	Model
Central Processing Unit (CPU)	Intel i5 10400F 2.9 GHz–4.1 GHz
Random Access Memory (RAM)	16 GB 2133 MHz
Graphics Processing Unit (GPU)	GTX1060 6 GB

Table 4. Algorithm execution times.

Algorithm	Time [ms]
LS	0.0269 ± 0.0095
STA	13.5 ± 0.344
CDP	18.4 ± 0.471
ELM	40.5 ± 5.1
C-ELM [13]	127 ± 9.74
SS-ELM	658 ± 10.4
Parallel SS-ELM	167 ± 2.6

The results of Table 4 depict that the proposed SS-ELM increases the execution time by around 19 times compared to the CDP estimator for instance, and when optimized with parallel computing, the execution time is reduced to 167 ± 2.6 ms, increasing the time to approximately nine times. Note that although the SS-ELM scheme requires the largest operation time among the evaluated techniques, its execution time is still far from the upper limit of latency equal to 300 ms [5] specified by ETSI for safety applications. It is important to consider that these results are software/hardware-dependent, so the results may vary from device to device. Finally, considering that the computer processing units that modern cars have on board are powerful, the overall results of the proposed algorithm are sufficient to justify an increase in the execution time. This is especially important for safety applications in vehicular communications, where a low BER is vital to ensure the reliability of the information. Considering the execution time. the proposed algorithm is suitable for safety applications such as collision risk warning or road hazard signaling [5]. However, its use for critical safety applications such as pre-crash sensing should be investigated further.

Note that although the proposed SS-ELM scheme has the longest execution time with respect to the compared models, we conclude that it is possible to further reduce the learning and/or operation time of the scheme by having only one layer forward, e.g., on an Field-Programmable Gate Array (FPGA) [32]. In future work, we will explore the use of FPGAs to implement the proposed scheme, analyse the impact of a larger sample size of δ, the possibility of applying unsupervised configurations as an alternative to semi-supervised ones, and analyse the impact of using channel coding schemes in different system configurations.

5. Conclusions

This work demonstrates the feasibility of using ML algorithms as an alternative to classical techniques, used in wireless communication systems, to estimate and equalize the vehicular communication channel. A channel estimation and equalization algorithm based on extreme learning machines was designed and tested over two different channel system configurations: overtake and intersection scenarios.

An improved version of an SS-ELM channel estimator and equalizer was introduced for vehicular communications considering the 802.11p amendment. The proposed SS-ELM scheme outperformed the extreme learning machine (ELM) and the fully complex extreme learning machine (C-ELM) algorithms for the evaluated scenarios. Furthermore, the proposed SS-ELM scheme performed similarly compared to other traditional techniques.

The proposed scheme had a better performance for lower E_b/N_0 values compared to STA and CDP for the first and second system configurations, respectively, considering that E_b/N_0 values in wireless environments are usually lower than 20 dB.

We analyzed the impact that δ has on the equalization process, and concluded that there is still room for improvement when considering rougher channel configurations than the ones shown in this work. As the value of δ increases, the algorithm requires a longer execution time, but a larger δ enhances the performance of the algorithm, it becomes suitable for optimization.

Finally, the execution time of the proposed model was simulated on a general-purpose computer with an interpreted language such as Matlab, for a fair comparison with the other models. Even though the SS-ELM scheme requires the largest execution time among the evaluated techniques, it still falls within the latency window specified by the standard.

Author Contributions: Conceptualization, E.S., C.A.A.-M. and D.Z.-B.; methodology, E.S. and D.Z.-B.; software, E.S. and D.Z.-B.; formal analysis, E.S.; investigation, E.S.; writing—original draft preparation, D.Z.-B., C.A.A.-M., S.B., E.S. and I.S.; writing—review and editing, D.Z.-B., C.A.A.-M., S.B., E.S. and I.S.; supervision, C.A.A.-M.; project administration, C.A.A.-M.; funding acquisition, C.A.A.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Vicerrectoría de Investigación y Desarrollo (VID) de la Universidad de Chile Proyecto ENL 01/20.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural Network
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CDP	Constructed Data Pilots
CFR	Channel Frequency Response
CP	Cyclic Prefix
CPU	Central Process Unit
C-ELM	Complex Extreme Learning Machine
C-V2X	Cellular Vehicular to Anything
DC	Direct Current
DL	Deep Learning
ELM	Extreme Learning Machine
ETSI	European Telecommunication Standards Institute
FFT	Fast Fourier Transform
FPGA	Field-Programmable Gate Array
GPU	Graphics Processing Unit
IFFT	Inverse Fast Fourier Transform
LS	Least Squares
ML	Machine Learning
MMSE	Minimum Mean-Square Error
OFDM	Orthogonal Frequency Division Multiplexing
PHY	Physical Layer
RAM	Random Access Memory
SS	Semi-Supervised
SS-ELM	Semi Supervised Extreme Learning Machine
STA	Spectral Temporal Averaging
SNR	Signal to Noise Ratio
VCS	Vehicular Communication Systems
V2V	Vehicle to Vehicle
V2I	Vehicle to Infrastructure
WiFi	Wireless Fidelity

WSSUS	Wide-Sense Stationary Uncorrelated Scattering
ZF	Zero Forcing

References

1. Ahmed, E.; Gharavi, H. Cooperative Vehicular Networking: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 996–1014. [CrossRef] [PubMed]
2. Wang, J.; Liu, J.; Kato, N. Networking and Communications in Autonomous Driving: A Survey. *IEEE Commun. Surv. Tutor. I* **2019**, *21*, 1243–1274. [CrossRef]
3. ETSI TR 102 638 V1.1.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions. Available online: https://www.etsi.org/deliver/etsi_tr/102600_102699/102638/01.01.01_60/tr_102638v010101p.pdf (accessed on 9 January 2015).
4. ETSI. TS 101 539-2 V1.1.1: Intelligent Transport Systems (ITS); V2X Applications; Part 2: Intersection Collision Risk Warning (ICRW) Application Requirements Specification 1. Available online: https://www.etsi.org/deliver/etsi_ts/101500_101599/10153902/01.01.01_60/ts_10153902v010101p.pdf (accessed on 30 June 2018).
5. ETSI. TS 101 539-3 V1.1.1: Intelligent Transport Systems (ITS); V2X Applications; Part 3: Longitudinal Collision Risk Warning (LCRW) Application Requirements Specification. Available online: https://www.etsi.org/deliver/etsi_ts/101500_101599/10153903/01.01.01_60/ts_10153903v010101p.pdf (accessed on 30 November 2013).
6. Ortega, N.M.; Azurdia-Meza, C.A.; Gutierrez, C.A.; Maciel-Barboza, F.M. On the Influence of the non-WSSUS Condition in the Performance of IEEE 802.11-Based Channel Estimators for Vehicular Communications. In Proceedings of the IEEE 10th Latin-American Conference on Communications (LATINCOM), Guadalajara, Mexico, 14–16 November 2018; pp. 1–6. [CrossRef]
7. *802.11-2016 - IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*; IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007); IEEE: Miami, FL, USA, 2016; pp. 1–3534. [CrossRef]
8. Zhao, Z.; Cheng, X.; Wen, M.; Jiao, B.; Wang, C. Channel Estimation Schemes for IEEE 802.11p Standard. *IEEE Intell. Transp. Syst.* **2009**, *5*, 38–49. [CrossRef]
9. Fernandez, J.; Borries, K.; Cheng, L.; Kumar, B.V.K.V.; Stancil, D.D.; Bai, F. Performance of the 802.11p physical layer in vehicle-to-vehicle environments. *IEEE Trans. Veh. Tech.* **2018**, *61*, 3–14. [CrossRef]
10. Ye, H.; Liang, L. Machine Learning for Vehicular Networks. *arXiv* **2018**, arXiv:1712.07143.
11. Ye, H.; Li, G.; Juang, B. Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems. *IEEE Wirel. Commun. Lett.* **2018**, *7*, 114–117. [CrossRef]
12. Sattiraju, R.; Weinand, A.; Schotten, H. Channel Estimation in C-V2X using Deep Learning. In Proceedings of the 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Goa, India, 16–19 December 2019.
13. Liu, J.; Mei, K.; Zhang, X.; Ma, D.; Wei, J. Online Extreme Learning Machine-Based Channel Estimation and Equalization for OFDM Systems. *IEEE Commun. Lett.* **2019**, *23*, 1276–1279. [CrossRef]
14. Gutiérrez, C.; Pätzold, M.; Dahech, W.; Youssef, N. A Non-WSSUS Mobile-to-Mobile Channel Model Assuming Velocity Variations of the Mobile Stations. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017. [CrossRef]
15. Gutierrez, C.; Gutierrez-Mena, J.; Luna-Rivera, J.; Campos-Delgado, D.; Velazquez, R.; Patzold, M. Geometry-Based Statistical Modeling of Non-WSSUS Mobile-To-Mobile Rayleigh Fading Channels. *IEEE Trans. Veh. Technol.* **2018**, *1*, 362–377. [CrossRef]
16. Jaime-Rodríguez, J.J.; Gómez-Vega, C.A.; Gutiérrez, C.A.; Luna-Rivera, J.M.; Campos-Delgado, D.U.; Velázquez, R. A non-WSSUS channel simulator for V2X communication systems. *Electronics* **2020**, *9*, 1190. [CrossRef]
17. Gutiérrez, C.A.; Pätzold, M.; Ortega, N.M.; Azurdia-Meza, C.; Maciel-Barboza, F.M. Doppler shift characterization of wideband mobile radio channels. *IEEE Trans. Veh. Technol.* **2019**, *68*, 12375–1238. doi:10.1109/TVT.2019.2945936. [CrossRef]
18. Huang, G.; Song, S.; Gupta, J. Semi-supervised and unsupervised extreme learning machines. *IEEE Trans. Cybern.* **2014**, *12*, 2405–2417. [CrossRef]
19. Souza, P.V.d.C.; Torres, L.C.B.; Silva, G.R.L.; Braga, A.d.P.; Lughofer, E. An advanced pruning method in the architecture of extreme learning machines using l1-regularization and bootstrapping. *Electronics* **2020**, *9*, 811. [CrossRef]
20. Li, S.; Song, S.; Wan, Y. Laplacian twin extreme learning machine for semi-supervised classification. *Neurocomputing* **2018**, *321*, 17–27. [CrossRef]
21. Zabala-Blanco, D.; Mora, M.; Azurdia-Meza, C.A.; Dehghan Firoozabadi, A.; Palacios Játiva, P.; Soto, I. Relaxation of the Radio-Frequency Linewidth for Coherent-Optical Orthogonal Frequency-Division Multiplexing Schemes by Employing the Improved Extreme Learning Machine. *Symmetry* **2018**, *8*, 632. [CrossRef]
22. Zabala-Blanco, D.; Mora, M.; Azurdia-Meza, C.A. Dehghan Firoozabadi, Extreme Learning Machines to Combat Phase Noise in RoF-OFDM Schemes. *Electronics* **2019**, *12*, 921. [CrossRef]
23. Ijiga, O.E.; Ogundile, O.O.; Familua, A.D.; Versfeld, D.J. Review of channel estimation for candidate waveforms of next generation networks. *Electronics* **2019**, *8*, 956. [CrossRef]
24. Liu, N.; Wang, H. Ensemble based extreme learning machine. *IEEE Signal Process. Lett.* **2010**, *17*, 754–757. [CrossRef]
25. Carrera, D.F.; Zabala-Blanco, D.; Vargas-Rosales, C.; Azurdia-Meza, C.A. Extreme Learning Machine-Based458Receiver for Multi-User Massive MIMO Systems. *IEEE Commun. Lett.* **2021**, *25*, 484–488. [CrossRef]

26. Yang, L.; Zhao, Q.; Jing, Y. Channel Equalization and Detection with ELM-Based Regressors for OFDM Systems. *IEEE Commun. Lett.* **2019**, *24*, 86–89. [CrossRef]
27. Carrera, D.F.; Vargas-Rosales, C.; Yungaicela-Naula, N.M.; Azpilicueta, L. Comparative Study of Artificial Neural Network Based Channel Equalization Methods for MmWave Communications. *IEEE Access* **2021**, *9*, 41678–41687. [CrossRef]
28. Shivaldova, V.; Winkelbauer, A.; Mecklenbrauker, C. Signal-to-noise ratio modeling for vehicle-to-infrastructure communications. In Proceedings of the 2014 IEEE 6th International Symposium on Wireless Vehicular Communications (WiVeC 2014), Vancouver, BC, Canada, 14–15 September 2014. [CrossRef]
29. Ortega, N.M.; Azurdia-Meza, C.A.; Gutierrez, C.A.; Gómez-Vega, C.A. Second Order Statistics and BER Performance Analysis of a non-WSSUS V2X Channel Model that Considers Velocity Variations. In Proceedings of the 2019 IEEE Latin-American Conference on Communications (LATINCOM), Salvador, Brazil, 11–13 November 2019; pp. 1–6. [CrossRef]
30. U.S. Department of Transportation. *Vehicle Safety Communications Project V Final Report*; Report DOT HS 810 591; National Highway Traffic Safety Administration: Washington, DC, USA, 2006.
31. Feukeu, E.; Djouani, K.; Kurien, A. An MCS Adaptation Technique for Doppler Effect in IEEE 802.11p Vehicular Networks. *Procedia Comput. Sci.* **2013**, *19*, 570–577. [CrossRef]
32. Frances-Villora, J.V.; Rosado-Muñoz, A.; Bataller-Mompean, M.; Barrios-Aviles, J.; Guerrero-Martinez, J.F. Moving learning machine towards fast real-time applications: A high-speed FPGA-based implementation of the OS-ELM training algorithm. *Electronics* **2018**, *7*, 308. [CrossRef]