Underwater Acoustic OFDM Receiver Using a Regression-based Deep Neural Network

Sabna Hassan*, Peng Chen*, Yue Rong*, Kit Yan Chan*
*School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Australia

OCEANS 2022, Hampton Roads | 978-1-6654-6809-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/OCEANS47191.2022.9976959

***Abstract*—The orthogonal frequency-division multiplexing (OFDM) technology is a promising technology for many scenarios in underwater acoustic (UA) communications. This paper presents the design of a UA OFDM communication system, which explains the design of a traditional transmitter and a modified receiver integrated with a deep neural network (DNN). The DNN is proposed to replace the channel estimation, channel equalization and demodulation in the traditional receiver design and recover transmitted bits directly. The regression-based DNN consists of a long short-term memory (LSTM) layer. The training stage of the DNN can be either offline or online. During the testing stage, the trained network is used to recover online transmitted data directly. The offline training method is performed with maximum possible channel scenarios with a large data set. Meanwhile, the online training uses a small data set with short training time. The designed regression-based DNN receiver achieves a better performance compared to previously developed DNN receivers and the traditional receiver which is implemented with the least-squares (LS) estimator.**

I. Introduction

Reliable communication through an underwater acoustic (UA) channel is significantly challenging due to its limited bandwidth, significant multipath interference and rapid time variation caused by Doppler shifts [1]. Orthogonal frequency-division multiplexing (OFDM) is an attractive technique for UA communication, since it has the capability for multipath channel with a long delay spread.

In the past years, machine learning has evolved as an effective way to perform modeling and pattern recognitions for complex systems. Emotion recognition using human brain-activity sensors [2], vehicle type classification [3] and speech separation methods [4] are some of the examples. A neural network is a collection of layers, where each layer has a number of neurons. Each layer performs a weighted sum of the inputs followed by a nonlinear activation and the output is fed as an input to the next layer [5]. Through the training process, the weights and biases are tuned according to the provided training data. A loss function is used for the parameter estimation of a supervised deep learning. During the training the loss function is used to optimize the optimal weights and biases [6].

The scope of deep learning in emerging communication applications is a recent research topic. Deep learning-based vehicular communication applications [7] and smart cities [8] are some of the examples. In this paper, a UA communication system is proposed by integrating a regression-based deep learning technique [9] with a long short-term memory (LSTM) layer in the architecture. Channel state information (CSI) plays a crucial part in UA communication systems. Accurate estimation of CSI provides a reliable OFDM system performance [10] [11]. In the traditional methods, CSI is estimated by using pilot symbols prior to detecting the transmitted data symbols. When the deep neural network (DNN) is implemented, explicit channel estimation and equalization are replaced, and the transmitted data can be recovered without using the conventional demodulation techniques. The DNN based systems utilise the data during its training period to understand the channel. If the DNN is trained offline, the training data will contain maximum possible channel scenarios. This is achieved by varying the channel during the generation of each data set. During the online training, a real time data is used to tune the DNN just before the testing. From these training data the DNN will learn the characteristics of the channel and prepare itself to directly recover transmitted data from received signal. Hence, quality of data transmission can be improved in terms of increasing data rate and decreasing error probability.

The DNN parameters are determined according to the channel condition during the training, and the DNN predicts the transmitted data with those channel information [12]. This paper presents a novel technology for UA communication systems which uses the DNN to replace the channel estimation, equalization and demodulation. Instead of using a fully connected layer after the input layer, an LSTM layer is employed in the proposed design. The simulation results show that the DNN performance is comparable to the traditional least-squares (LS) method. The implementation of an LSTM layer improves the performance in terms of bit-error-rate (BER).

The reminder of this paper is as follows. In Section 2, review of a conventional UA communication system, DNN, LSTM architecture, and the proposed system model are presented. The proposed system design is explained in Section 3 with two training methods for a DNN. In Section 4, experiment results are presented in details. Finally, conclusions are drawn in Section 5.

II. System Models

1) Conventional UA communication system: A traditional UA communication system is illustrated in Fig. 1. In the transmitter, the input data is modulated using the quadrature phase-shift keying (QPSK) modulation and converted into time domain through inverse fast Fourier transform (IFFT) (1). Cyclic prefix (CP) is used to reduce the effect of intersymbol interference (ISI) and maintain the orthogonality of subcarriers [13].

978-1-6654-6809-1/22/$31.00 ©2022 IEEE

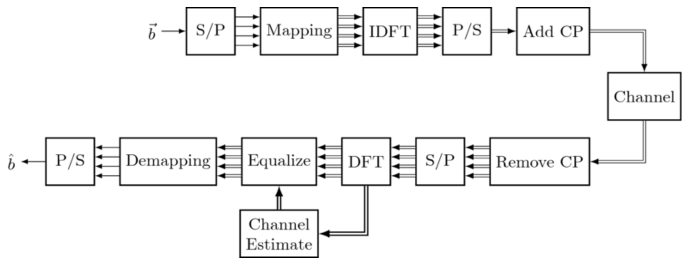

Fig. 1. Block diagram of a traditional UA communication system [14].

$$s(t) = \sum_{k=0}^{N-1} X_k exp(j2\pi kft) \tag{1}$$

where X_k is the data symbol, N is the FFT length, and $f = 1/T$ is the frequency spacing.

In the receiver side of a conventional UA communication system, the CP is removed and the fast Fourier transform (FFT) is performed to convert the received signals into frequency domain. Then, the channel estimation is performed by either LS or minimum mean-squared error (MMSE) method with the pilot sequences. The estimated channel value is used to recover the transmitted bits from the received sequences.

2) Deep neural network: DNN is a multi-layered artificial neural network inspired from the function and structure of a human brain [15]. As one of the most powerful classification tools, DNN is widely applied in computer vision, natural language processing, speech recognition and signal processing applications such as modulation recognition, signal detection and channel estimation [16]. A general configuration of the DNN is shown in Fig. 2.

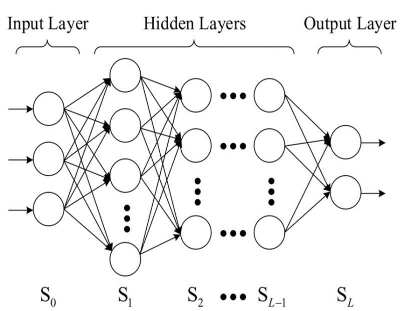

Fig. 2. Structure of a neural network [17].

The DNN consists of an input layer, an output layer and at least one hidden layer for processing the nonlinear elements. Each layer consists of neurons, bias, weights and activation functions. The number of hidden layers and the number of neurons in each layer can be determined by the trial and error process. The activation function is an important element to determine the relationship between the inputs and the outputs [18]. There are a variety of neural networks available such as CNN (convolutional neural network) and RNN (recurrent neural network). The DNN has a multi-layered structure with multiple neurons in each layer. Each neuron has a nonlinear activation function (3), which is activated by the inputs and

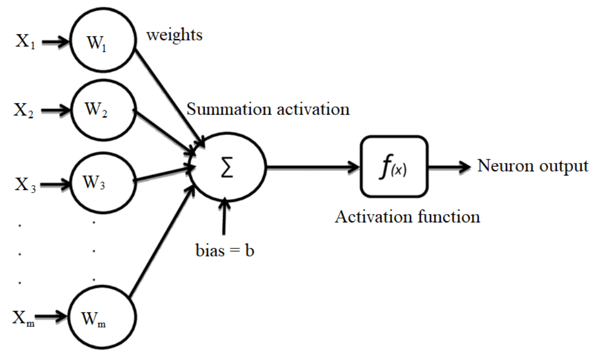

Fig. 3. Structure of a neuron [18].

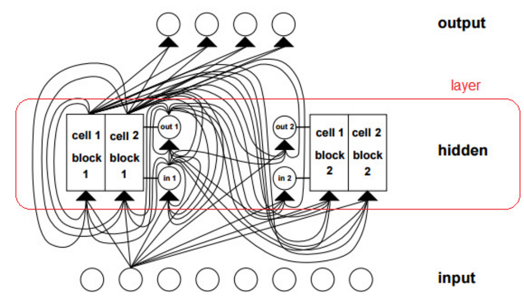

Fig. 4. LSTM architecture [21].

has been adjusted by adaptable weights. Therefore, the neural network can approximate any nonlinear function arbitrarily [19].

A neuron shown in Fig. 3 is a sum of inputs $X_1, X_2, \ldots, X_m$, corresponding weights $W_1, W_2, \ldots, W_m$ of that inputs and bias value b of that neuron.

$$x = \sum_{i=1}^{m} W_i X_i + b \tag{2}$$

Output of neuron is given as,

$$a = f(x) = f(\sum_{i=1}^{m} W_i X_i + b) \tag{3}$$

where $f(x)$ is the activation function [20].

3) LSTM architecture: LSTM neural network is an extension of RNN, designed to learn sequence data. The program employs a structure based on short-term memory processes to build longer-term memory. The primary components of an LSTM architecture are the cell state and its regulators as shown in Fig. 4. The cell state is the memory unit of the network. Through open and close gates, the cell state stores information that can be written to, read from, or stored in a previous cell state. Information from the previous step can also enter the cell state and carry relevant information throughout the data processing. Using a recurrent neural network learning process, the gates will decide between the data to remember and to discard throughout training [21]. A single classic LSTM machine learning model consists of a cell state and three gates: a forget gate, an input, and an output gate as shown in Fig. 5.

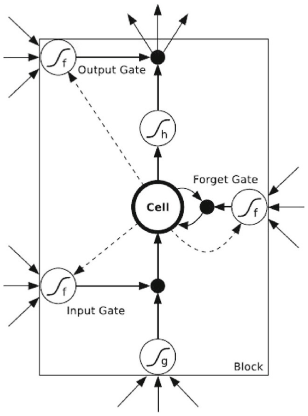

Fig. 5. LSTM memory block with one cell [22].

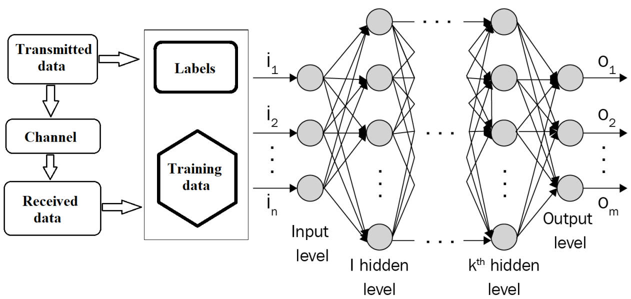

Fig. 6. Training process of the DNN.

4) *UA communication receiver using regression-based DNN:*

In this paper, we implement a frame-based UA OFDM communication system. Each OFDM frame contains a pilot block and a data block. In the data block, a randomly generated binary data stream of bits is encoded to form a sequence of encoded bits. The coded sequence is mapped into data symbols drawn from the QPSK constellation. Each OFDM symbol is converted to the time domain by the IFFT, where a CP with a length T_{cp}, which is longer than the channel delay spread, is then added to the time domain symbol. At the receiver end, the frame is organized as the input to the DNN, which has an LSTM layer, after downshifting and removing the CP in (4).

$$y(t) = x(t) * h(t) + w(t) \tag{4}$$

where $y(t)$ is the received signal, $x(t)$ is the transmitted signal, * denotes the convolution operation, $h(t)$ is the impulse response of the channel, and $w(t)$ is the channel noise.

This received data and the transmitted data are processed to generate training data as shown in Fig. 6. The features in the received data provide information about the channel and transmitted bits to the DNN. The DNN modifies its parameters according to these information in the training data. Then, the DNN predicts the transmitted data directly from $y(t)$ without the explicit channel estimation, equalization and demodulation.

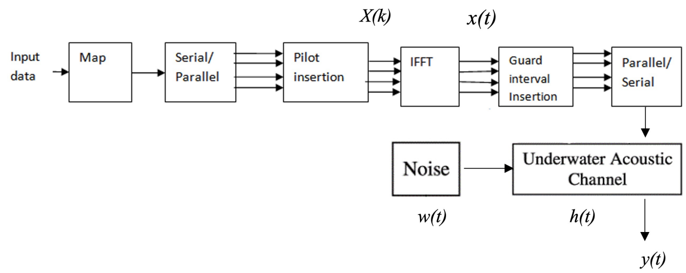

Fig. 7. Block diagram of the transmitter.

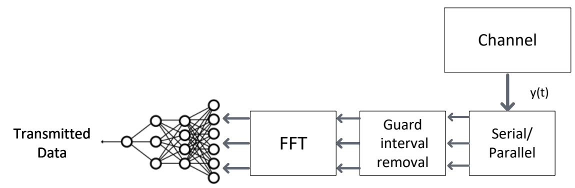

Fig. 8. Block diagram of the receiver.

III. System Design

In this section, the system design for the traditional transmitter and the modified receiver is presented. The transmitter and receiver designs are shown in Fig. 7 and Fig. 8 respectively.

1) Transmitter: The system design contains one OFDM frame. Each frame contains one OFDM data block and one pilot block. In each data block, there are 64 subcarriers and the pilot block consists of 64 pilot subcarriers. The data symbols are modulated by QPSK constellations. Hence one symbol is encoded by two bits. The CP is chosen as T_{cp} =20 ms with 15 channel paths.

2) Receiver: The baseband signal processing at the receiver is shown in Fig. 8. The figure shows that the signals received from the transmitter are undergone FFT and then the baseband signals are fed into the DNN. The DNN consists of several layers including an input layer, an output layer and some hidden layers. Behind the input layer, an LSTM layer is placed in our network design with a specific number of neurons. The LSTM layer architecture is a set of recurrently connected subnets, known as memory blocks as illustrated in Fig. 5. These blocks are similar to a differentiable version of the memory chips in a computer [22]. The output layer is a regression layer, which gives the demodulated transmitted bits.

Two different ways of training a neural network

a) Offline training: Training is essential to develop a DNN based system. Generally, a large data set is used for training a neural network which is capable of providing all possible features of the problem to the network. By learning these features, the weights and biases of the DNN are modified accordingly. In offline training, a large data set is used with maximum possible CSI. The channel requirement for offline training is that it remains quasi-stationary as below.

$$h[n] = h' + \epsilon[n] \tag{5}$$

where $h' = [h'_1, h'_2, ..., h'_G]^T$ is the original randomly generated channel with Gaussian distribution, G is the number of channel paths, n is the number of OFDM packets and $\epsilon[n] = [\epsilon_1[n], \epsilon_2[n], ..., \epsilon_G[n]]^T$ is a random vector with normal distribution.

$$\epsilon_k[n] \sim \mathcal{N}(0, \sigma^2) \tag{6}$$

where σ^2 is the variance and $k = 1, 2, ..., G$.

b) Online training: Online training updates the DNN with real time channel properties. The important factor is the training time in online training. Shorter training time provides better results in real time as the channel variation between the training and the testing is short. Hence, we use a small data set, which is generated from randomly selected received subcarriers.

Retraining from checkpoint improves the performance of the online training method. The trained network is retrained with a new data set in this method. However, we can start from a reference point which is saved as the checkpoint, instead of retraining the network from zero knowledge. This will improve the prediction capability of the DNN, since the noise and channels are both time-varying.

Both transmitted and received data are used for the training data generation as illustrated in Fig. 6. During the training process information of transmitted and received data is utilised for tuning the parameters of the DNN, such as the weights and biases. The DNN parameters are determined using the stochastic gradient descent and the back-propagation algorithms. The loss function L in (7) attempts to reduce the difference between transmitted sequences and outputs of the DNN.

$$L = 1/N \sum_{k=0}^{N-1} (\hat{x}(k) - x(k))^2 \tag{7}$$

where $\hat{x}(k)$ is the prediction and $x(k)$ is the supervision data [23].
For testing, the test data and its supervision labels are generated by another set of data. The trained DNN is used to predict the transmitted data from the given test data. By comparing the predicted data with the labels, the accuracy of the DNN can be evaluated. The proposed approach uses the DNN to replace the channel estimation, equalization and demodulation in the traditional receiver.

IV. Experiment Results

In this section, we evaluate the performance of our proposed UA OFDM system through simulations by comparing the performance of the DNN with an LSTM layer, the DNN designed in [23] and the traditional LS method. The designed DNN consists of an input layer with a number of neurons, which depends on the numbers of subcarriers used to generate the training data, followed by an LSTM layer, a fully connected layer with a number of neurons according to the numbers of bits recovered, and a regression layer as the output layer. We chose 15 channel paths with 20 samples delay spread.

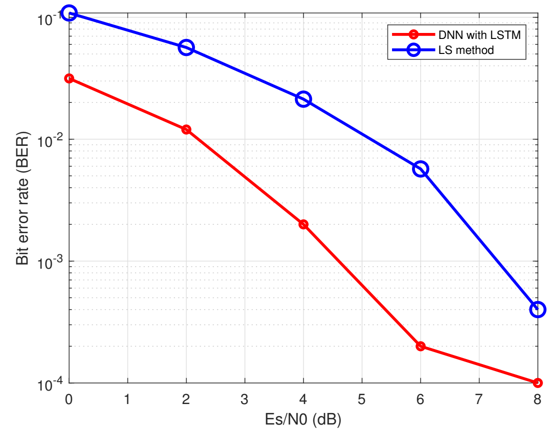

Fig. 9. BER performance; Same SNR for training (offline) and testing with time-invariant channel.

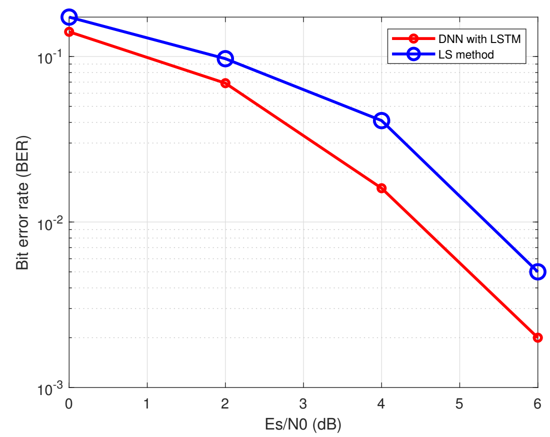

Fig. 10. BER performance; Different SNR for training (offline) and testing with time-invariant channel.

Fig. 9 and Fig. 10 illustrate the BER performance of the proposed system in offline training method in a time-invariant channel. 50,000 OFDM packets with 64 data subcarriers and 64 pilot subcarriers are used for the training data generation and the DNN with 5 layers is used as the network in simulations. Fig. 9 shows that, the BER performance of the DNN system is significantly better than the LS method, when the same SNRs are used for both the training and the testing. Fig. 10 shows the performance comparison for the DNN trained with a fixed SNR and tested with a range of SNRs. In this case SNR= 50dB is used to generate the training data set and SNR from 0 to 50 dB is used to generate the testing data.

In Fig. 11, the BER performances are illustrated for the proposed DNN with an LSTM layer, traditional LS method and the DNN without an LSTM layer. For offline training 80,000 OFDM packets of 64 data subcarriers and 64 pilot subcarriers are used to generate the training data with a time-varying channel. The channel is slightly different as (5) and (6) with $\sigma^2 = 0.06$ for each OFDM packet. A five layered LSTM DNN is used as the network, which has an input layer with 256 neurons followed by an LSTM layer with 500 neurons, two fully connected layers with 250 and 4 neurons respectively,

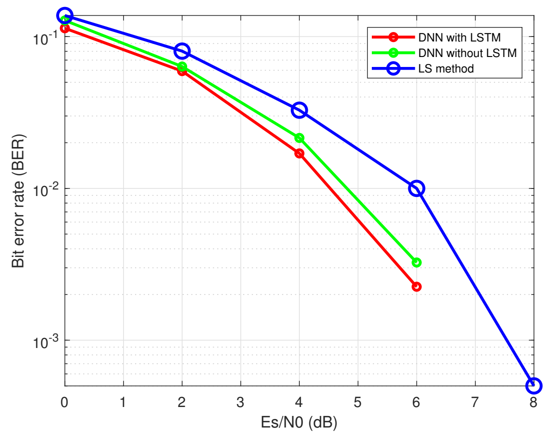

Fig. 11. Offline training with time-varying channel for testing and training.

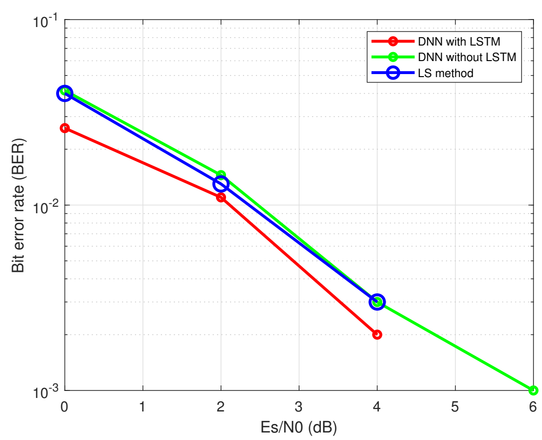

Fig. 12. Online training with time-invariant channel for testing and training.

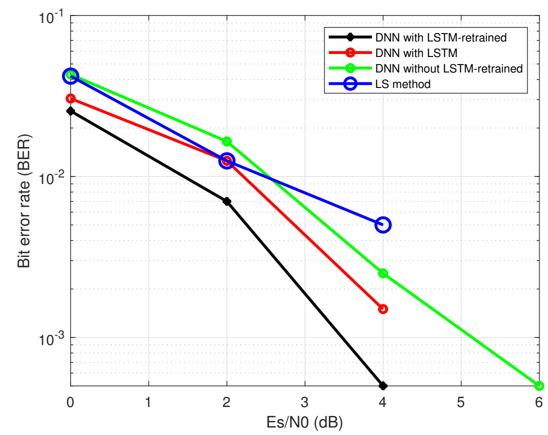

Fig. 13. Online trained and retrained network performances with time-invariant channel.

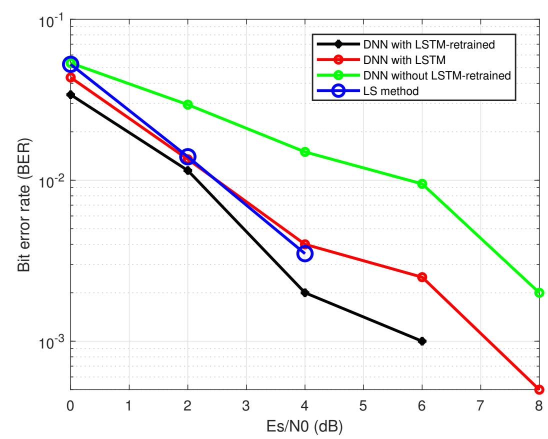

Fig. 14. Online trained and retrained network performances with time-varying channel.

and a regression layer as the output layer.

For online training, 2,000 OFDM packets of randomly selected 10 data subcarriers and 10 pilot subcarriers are used to generate the training data with a time-invariant channel. Better performance is achieved for LSTM DNN with 4 layers, which contains an input layer with 40 neurons followed by an LSTM layer with 30 neurons, a fully connected layer with 4 neurons and a regression layer as the output layer. BER performance comparison is shown in Fig. 12. The DNN with LSTM shows better BER performance than the traditional LS method and the DNN proposed in [23], which has no LSTM layer.

Retraining from checkpoint is performed in online training methods. 2,000 and 1,000 data packets are used for training and retraining the network respectively. Randomly selected 10 data subcarriers, 10 pilot subcarriers, and a time-invariant channel are used for the online training data sets. The performance is improved by retraining the network from the checkpoint of last epoch with a new set of data. Fig. 13 illustrates the performance of the trained and retrained networks during testing with a time-invariant channel. The figure shows that, the performance of the DNN with the LSTM layer is improved after the retraining. The BER performance after testing with a time-varying channel is illustrated in Fig. 14. The retraining is performed on both the proposed LSTM network and the network designed in [23]. However, the DNN with LSTM layer achieves better performance after the retraining.

V. CONCLUSION

In this paper, a UA OFDM receiver integrated with a regression-based deep learning is proposed. The results show that, the proposed approach achieves a better performance than the traditional receiver which uses the LS channel estimation method or the DNN proposed in [23]. The simulation results also show that, the proposed regression-based DNN receiver is capable to learn the time-varying channel and adapt accordingly to decode the transmitted bits. The DNN can achieve higher accuracy in decoding the transmitted data in varying channel scenarios by improving the training methods in both online and offline training. Better results are achieved by reducing the training time for the online training method and increasing the robustness against channel variation in the offline training method.

REFERENCES

[1] G. Qiao, Z. Babar, L. Ma, S. Liu, and J. Wu, "MIMO-OFDM underwater acoustic communication systems—a review," *Physical Communication*, vol. 23, pp. 56–64, 2017.

[2] R. M. Mehmood, R. Du, and H. J. Lee, "Optimal feature selection and deep learning ensembles method for emotion recognition from human brain EEG sensors," *IEEE Access*, vol. 5, pp. 14 797–14 806, 2017.

[3] W. Liu, M. Zhang, Z. Luo, and Y. Cai, "An ensemble deep learning method for vehicle type classification on visual traffic surveillance sensors," *IEEE Access*, vol. 5, pp. 24 417–24 425, 2017.

[4] X.-L. Zhang and D. Wang, "A deep ensemble learning method for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 967–977, 2016.

[5] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.

[6] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab, "Robust optimization for deep regression," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[7] C.-H. Lin, Y.-C. Lin, Y.-J. Wu, W.-H. Chung, and T.-S. Lee, "A survey on deep learning-based vehicular communication applications," *Journal of Signal Processing Systems*, vol. 93, no. 4, pp. 369–388, 2021.

[8] P. J. Werbos, "Computational intelligence for the smart grid-history, challenges, and opportunities," *IEEE Computational Intelligence Magazine*, vol. 6, no. 3, pp. 14–21, 2011.

[9] S. Lathuilière, P. Mesejo, X. Alameda-Pineda, and R. Horaud, "A comprehensive analysis of deep regression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 9, pp. 2065–2081, 2019.

[10] S. Wang, Z. He, K. Niu, P. Chen, and Y. Rong, "New results on joint channel and impulsive noise estimation and tracking in underwater acoustic OFDM systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2601–2612, 2020.

[11] P. Chen, Y. Rong, S. Nordholm, and Z. He, "Joint channel and impulsive noise estimation in underwater acoustic OFDM systems," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 567–10 571, 2017.

[12] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2017.

[13] B. Pranitha and L. Anjaneyulu, "Performance evaluation of underwater communication system using block codes," in *2016 Fifth International Conference on Eco-friendly Computing and Communication Systems (ICECCS)*, 2016, pp. 58–62.

[14] N. Vij, "Deep learning in physical layer for various networks," Ph.D. dissertation, California State University, Sacramento, 2019.

[15] J. Brownlee. How to use learning curves to diagnose machine learning model performance, https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance.

[16] Z. Chen, Z. He, K. Niu, and Y. Rong, "Neural network-based symbol detection in high-speed OFDM underwater acoustic communication," in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2018, pp. 1–5.

[17] J. Zhang, Y. Cao, G. Han, and X. Fu, "Deep neural network-based underwater OFDM receiver," *IET Communications*, vol. 13, no. 13, pp. 1998–2002, 2019.

[18] J. A. Yacim and D. G. B. Boshoff, "Impact of artificial neural networks training algorithms on accurate prediction of property values," *Journal of Real Estate Research*, vol. 40, no. 3, pp. 375–418, 2018.

[19] Y. Wang, H. Zhang, Z. Sang, L. Xu, C. Cao, and T. A. Gulliver, "Modulation classification of underwater communication with deep learning network," *Computational Intelligence and Neuroscience*, vol. 2019, 2019.

[20] R. Jiang, X. Wang, S. Cao, J. Zhao, and X. Li, "Deep neural networks for channel estimation in underwater acoustic OFDM systems," *IEEE Access*, vol. 7, pp. 23 579–23 594, 2019.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 12 1997.

[22] A. Graves, "Long short-term memory," *Supervised Sequence Labelling with Recurrent Neural Networks*, pp. 37–45, 2012.

[23] Y. Zhang, J. Li, Y. Zakharov, X. Li, and J. Li, "Deep learning-based underwater acoustic OFDM communications," *Applied Acoustics*, vol. 154, pp. 53–58, 2019.