Performance of Cellular Neural Network Based Channel Equalizers

A. ÖZMEN, B. TANDER, and H. ŞENOL

Abstract—**In this paper, a popular dynamic neural network structure called Cellular Neural Network (CNN) is employed as a channel equalizer in digital communications. It is shown that, this nonlinear system is capable of suppressing the effect of intersymbol interference (ISI) and the noise at the channel. The architecture is a small-scaled, simple neural network containing only 25 neurons (cells) with a neighborhood of r = 2 , thus including only 51 weight coefficients. Furthermore, a special technique called repetitive codes in equalization process is also applied to the mentioned CNN based system to show that the two-dimensional structure of CNN is capable of processing such signals, where performance improvement is observed. Simulations are carried out to compare the proposed structures with minimum mean square error (MMSE) and multilayer perceptron (MLP) based equalizers.**

Index Terms—**Cellular Neural Networks, channel equalization, MLP equalizer, MMSE equalizer, repetitive codes.**

I. Introduction

IN DIGITAL communication systems, the signal at the receiver will be the linear combination of time delayed and original transmitted signals, as a result of reflections and diffractions at the media. Therefore, the transmitted signal will reach to the receiver with a significant loss, which is called ISI. Typically, the ratio of the total erroneous bits to the total transmitted bits, which is called Bit-Error-Rate (BER) or Symbol-Error-Rate (SER) at the receiver, can be used as a measure. Equalization is the process that compensates the ISI and the impact of noise at the receiver by providing the maximum possible BER.

In literature, various methods are introduced for equalization. Equalizers that are complex and require too much computational power are designed for the applications where precision is needed rather than speed [1]–[6]. On the other hand, less complex however, relatively less accurate methods are also proposed when the speed is taken into account. A structure that models the equalizer with a transversal filter approach called zero-forcing (ZF) and the algorithm based on minimizing the mean square error between the equalizer output and the transmitted signal called MMSE are the most popular ones among these [7], [8].

Although these are commonly used linear transversal filters in channel equalization, their Bit-Error-Rates (BER) are not satisfactory. For this reason, alternative methods were developed in literature including Neural Network based architectures [9], [10]. However, even their BERs are better than the conventional techniques, because of their complex structures, they require too much computational power. At this point, CNN can be a good alternative to them with its simple topology. Furthermore, since the outputs of a CNN can take either -1 or +1 values, it is logical to use it in the reconstruction of Binary Phase Shift Keying (BPSK) signals. Formerly, CNN was employed to compute the coefficients of the linear transversal filters [11], somehow in our work, we directly used it as an equalizer itself. In [12], CNN structure uses a template with size of $3 \times 3 (r = 1)$ and also the method is applied for only non-repetitive codes. But in this work, CNN template size is selected as $5 \times 5 (r = 2)$ and the proposed method is applied to non-repetitive codes as well as to repetitive codes.

The paper is organized as follows: Firstly, the channel equalization process is introduced; secondly, a brief theory of the CNNs and how they are employed as an equalizer are presented. The MMSE and CNN channel equalization processes for repetitive codes are given in Section 4 and Section 5 respectively. The performance of the MMSE, MLP and the proposed CNN based structures are compared at the simulations section, both for classical and repetitively coded data. Finally, the advantages and drawbacks of CNN Equalizers are discussed at the conclusion section.

II. Channel equalization

At a digital communication system, the transmitted signal is distorted by ISI and noise factors as shown in Fig.1. Here, $s[n]$ is the original transmitted BPSK signal having a value of either -1 or $+1$, $h[n]$ is the transmission channel causing an ISI, $w[n]$ is the Gaussian noise, $y[n]$ is the received signal with the expression given in (1), $\hat{s}[n]$ is the equalized output.

$$y[n] = s[n] * h[n] + w[n]; \tag{1}$$

Where " $*$ " denotes the linear convolution.

BER is a measure of the perfectness of the $\hat{s}[n]$ output signal which can be expressed as follows:

$$BER = \frac{\text{total number of bit errors}}{\text{total number of transmitted bits}} = \frac{N_e}{N_b} \tag{2}$$

ATİLLA ÖZMEN, is with Department of Electrical-Electronics Engineering, Kadir Has University, Istanbul, Turkey (e-mail:aozmen@khas.edu.tr)
https://orcid.org/0000-0002-3868-1927

BARAN TANDER, is with Department of Mechatronics Engineering, Kadir Has University, Istanbul, Turkey (e-mail:tander@khas.edu.tr)
https://orcid.org/0000-0002-9037-4668

HABİB ŞENOL, is with Department of Computer Engineering, Kadir Has University, Istanbul, Turkey (e-mail:hsenol@khas.edu.tr)
https://orcid.org/0000-0001-5724-0839

Manuscript received January 31, 2019; accepted Nov 11, 2019.
DOI: 10.17694/bajece.519464

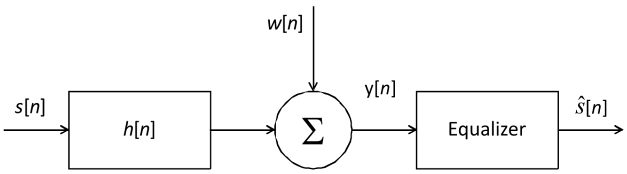


Fig. 1: Model of a digital communication system.

Channel equalization must be carried out in order to reconstruct the original information at the output. Therefore, the motivation of the channel equalization is to design a system that will minimize the difference between the output and the original $s[n]$ signals. In ideal case, making BER=0. Generally, the signal-to-noise-ratio (SNR) versus BER plots are utilized to evaluate the performances of the equalizers for different signal levels.

Although linear transversal filters as well as some special neural network structures, such as MLP can be used for this purpose; in this paper, a system based on CNN is proposed, and the performances are compared for various channel models.

III. Cellular neural networks

CNNs are a class of dynamic neural networks, first proposed by Chua and Yang in 1988 [13] and afterwards, because of their two-dimensional structures, found many impressive applications, especially in image processing [14]. When compared with other neural networks, their demand to less number of weight coefficients also appears to be an advantage, beyond the pros of the two dimensional architecture. As a dynamic neural network, a CNN neuron (Cell) seen in Fig.2, which is defined by the partial differential equation in (3), consists of an addition unit, an integration unit and a piecewise-linear activation function.

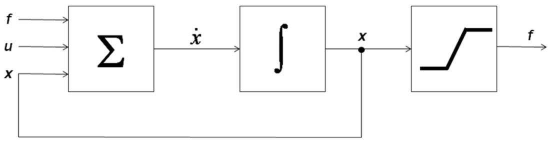


Fig. 2: A CNN cell model.

$$\dot{\mathbf{x}} = -\mathbf{x} + \mathbf{A}f(\mathbf{x}) + \mathbf{B}\mathbf{u} + I \tag{3}$$

Here, $\mathbf{x}$ is the "State", $\dot{\mathbf{x}}$ is its derivative, $\mathbf{u}$ is the input matrix and $f(\mathbf{x})$ is the activation function. $\mathbf{A}$ and $\mathbf{B}$ matrices are called "Cloning" and "Control" templates respectively that assign the interconnection coefficients (weight coefficients) between cells-cells and inputs-cells. Finally, I is a threshold value common for all neurons at the structure.

A stable CNN generates binary outputs $\{-1, 1\}$, since the activation function at the cells defined with (4) is a piecewise-linear function as shown in Fig.3. In other words, a stable CNN is said to approach either to positive or negative saturation regions on the mentioned characteristic. This feature can bring restrictions to many image processing applications however, from another point of view; the two-level output will be capable of reconstructing the distorted BPSK signals at the receiver end in a digital communication system.

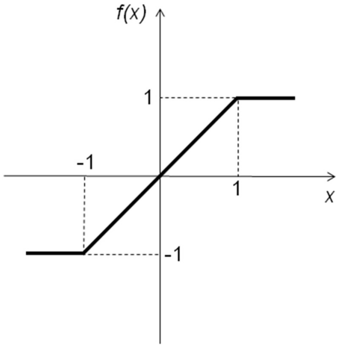


Fig. 3: Activation function of CNN.

$$f(x) = \frac{1}{2}\big\{|x+1| - |x-1|\big\} \tag{4}$$

At an MxN-cell CNN layer having M cells at its rows and N cells at its columns, the neurons will interact with each other by an r neighbourhood definition given below:

$$N_{ij} = \{C_{kl}|max\big\{|k{-}i|, |l{-}j|\big\} \leq r; 1 \leq k \leq M, 1 \leq l \leq N\} \tag{5}$$

One can see from the above equation that, if $r = 1$, like in most cases, a cell will be only connected to its nearest neighbors, that will dramatically decrease the number of weight coefficients: Specifically, if $r = 2$, the number of coefficients will be 25 for the $\mathbf{A}$ cloning, 25 for the $\mathbf{B}$ control templates which will form 5×5 matrices and an I threshold scalar; totally 51, allowing a relatively simpler implementation.

IV. Linear MMSE channel equalization

In a discrete-time communication system the received signal $y[n]$ can be given as follows:

$$y[n] \;\; = \;\; \sum_{\ell=0}^{N-1} H_{n,\,\ell}\, s[\ell] + w[n], \tag{6}$$

Since our system is Linear Time Invariant (LTI), $H_{n,\,\ell} = h[n-\ell]$. Where $s[n]$ is the input signal, $w[n]$ is additive white Gaussian noise (AWGN) and $H_{n,\,\ell}$ is the nth row ℓth column entry of the convolution matrix $\mathbf{H}$.

Suppose $s[\ell]$ symbols are M-repetitive coded of $d[k]$. By defining $\ell = kM + m$, it is straightforward that $s[kM+m] = d[k]$. So the observation model in (6) can be rewritten as follows:

$$\begin{aligned} y[n] &= \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} H_{n,\,kM+m}\, s[kM+m] + w[n] \\ &= \sum_{k=0}^{K-1} \breve{H}_{n,k}\, d[k] + w[n] \end{aligned} \tag{7}$$

where $\breve{H}_{n,k} = \sum\limits_{m=0}^{M-1} H_{n,\,kM+m}$ and this equation actually shows that, $\mathbf{H}$ is separated into K groups each containing M columns and then $\breve{\mathbf{H}}$ is constructed by the summation of the columns in each group. By collecting observation samples, (7) can be represented in vectorial form as follows

$$\mathbf{y} = \breve{\mathbf{H}}\mathbf{d} + \mathbf{w} \tag{8}$$

The linear MMSE estimate of the symbols determined from (8) will be as follows [15]

$$\hat{\mathbf{d}} = \left(\breve{\mathbf{H}}^{\dagger}\breve{\mathbf{H}} + \sigma_w^2 \mathbf{I}_K\right)^{-1} \breve{\mathbf{H}}^{\dagger}\mathbf{y} \tag{9}$$

However, since $d[k]$ is discrete, belonging to a signal constellation point, we must quantize to its nearest constellation point. Consequently, the detected symbols take the following form

$$\tilde{\mathbf{d}} = \text{Quant}(\hat{\mathbf{d}}), \tag{10}$$

where Quant$(\cdot)$ denotes the quantization process that quantizes its argument to its nearest data symbol constellation point.

V. Channel equalization with CNN

CNNs with 25 cells and with a neighborhood of $r = 2$, shown in Fig.4 is employed for the equalization process. Therefore, two 5×5 cloning and control template matrices $\mathbf{A}$, $\mathbf{B}$ and an I scalar, totaly 51 parameters have to be computed as depicted above.

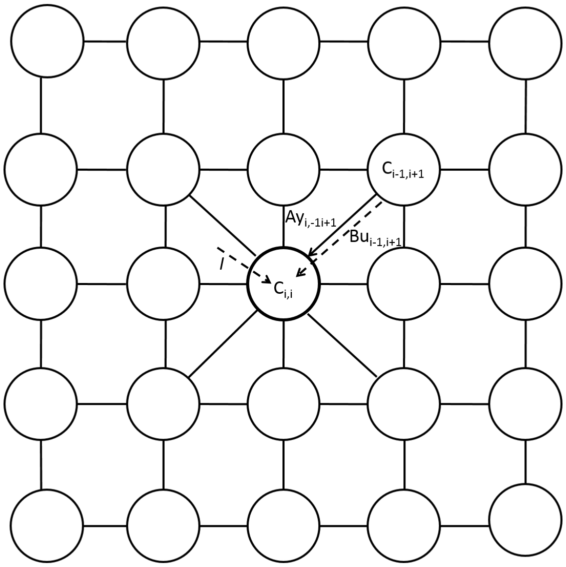


Fig. 4: The 5×5 CNN for the equalization.

A. Classical CNN equalizer design

In order to reorganize the one-dimensional distorted data in two-dimension to work with a CNN, five copies of the distorted data is placed one under the other and the designed CNN is slided above these five rows and a two-dimensional convolution with the 5×5 templates is performed as seen in Fig.5. In this case, each column at the the structure will interact between the copies of the $y[k-4]$, $y[k-3]$ ... $y[k]$ samples at these data rows. Now our goal became the determination of the weight coefficients of the mentioned 5×5 CNN.

Since CNN based equalizer design is to determine the 51 unknown weight coefficients of $\mathbf{A}$, $\mathbf{B}$ and I, a training process must be carried out as follows: Unequalized data with SNR values varying between 0 and 18dB, employing 1000 symbols for each $1dB$ range thus, total 19000 samples are applied as the inputs, and the desired uniformly distributed BPSK signals are used as the corresponding outputs for a chosen specific channel. The mentioned unknown templates are found by minimizing the mean square error between the input and desired output with the Genetic Algorithm [16], [17]. After the training, this CNN will be ready to be employed for the new test data.

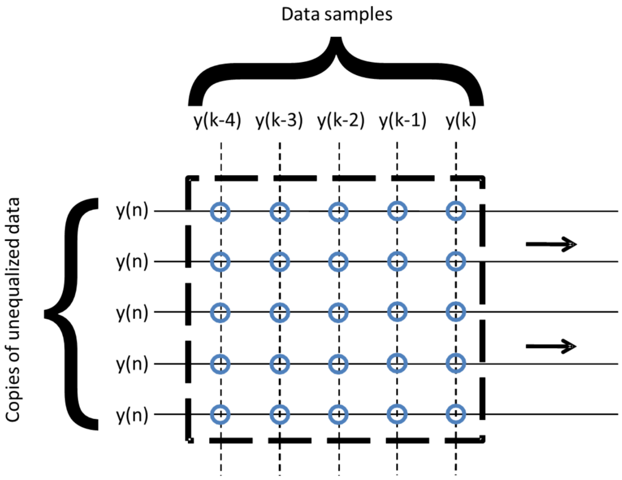


Fig. 5: CNN Equalizer design.

B. CNN Equalizer design with repetitive codes

In this section, the proposed system is applied to repetitively coded data, which means that the transmitted BPSK signal with a chosen length is repeated multiple times consecutively and reorganized as in the classical CNN equalizer design. It is known that, repetition codes provides better BER performance than the classically transmitted data. The detailed process is given below:

- Firstly, the random data is formed as a zero mean BPSK signal with M bits.
- Secondly, the data is converted to a repetitively coded signal by repeating each bit N times, therefore an MN bit sequence is obtained.
- Then, this repetitively coded data is passed through the h[n] channel in Figure 1 while $w[n]$ noise is added.
- Afterwards, before it is applied to the CNN equalizer, the MN bit sequence is paralleled to a 2D structure having N rows and M columns.
- Mentioned NxM structure is convolved with the CNN equalizer, to obtain stable outputs for each cell.
- If the number of +1s exceeds -1s on a particular column, then the system decides the output as +1 or vice-versa. This decision is indeed the reconstructed BPSK sequence.

All of the steps above at the receiver end are summarized at the block diagram in Fig.6. The training phase of the CNN

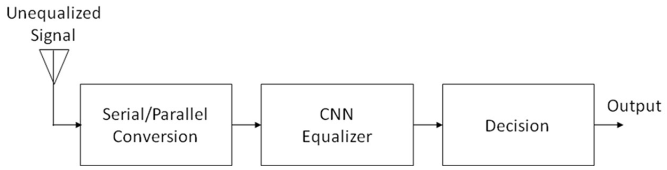


Fig. 6: The receiver for repetition codes

equalizer for repetitive codes, is performed by selecting a transmission signal with various SNRs between -10 and 2 dB in steps of 2 dB and the desired outputs corresponding to these inputs. The data length is chosen as $M = 5000$ for each SNR value. Each bit is repeated three times for 3 repetition codes and five times for 5 repetition codes in order to form a repetitively coded BPSK sequence. Again, the genetic algorithm is employed, where the weight coefficients of the 5×5 $\mathbf{A}$ and $\mathbf{B}$ templates are found after.

Afterwards, to observe the performance of the proposed CNN equalizer, BER curves are sketched for data having various SNR values.

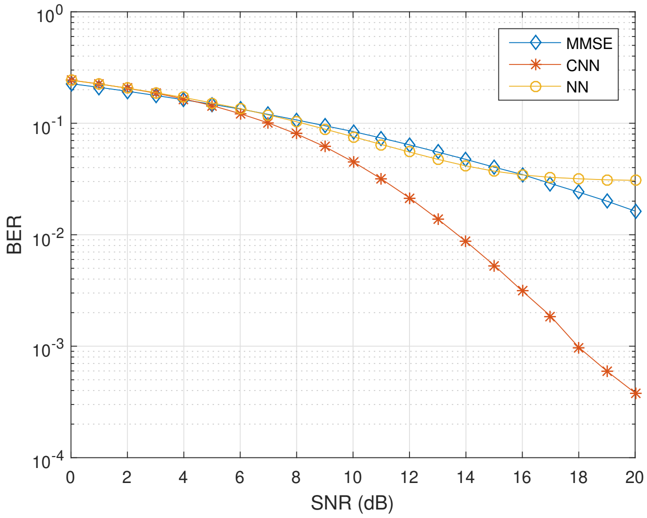


Fig. 7: Comparison of MMSE and CNN BER performances curves for channel B (without repetition).

VI. Simulation Results

The CNNs are trained and their BER performances are compared with the MMSE and MLP equalizers for the two individual channels defined with h_B and h_C below with and without repetition codes [7].

$$h_B = \{0.407, 0.815, 0.407\}$$

$$h_C = \{0.227, 0.460, 0.688, 0.460, 0.227\}$$

Simulations are carried out for non-repetition, 3-repetition and 5-repetition cases. After the training phase, the $\mathbf{A}$, $\mathbf{B}$ templates and the I thresholds for the CNNs are computed and then the performance of the MMSE, MLP and CNN equalizers are compared for these cases. Fig.7, Fig.9 and Fig.11 show the MMSE, MLP and CNN equalizer's BER curves for channel B in the case of non-repetition, 3-repetition and 5-repetition coded signals respectively. Fig.8, Fig.10 and Fig.12 show the BER curves of the equalizers for the channel C at the same cases.

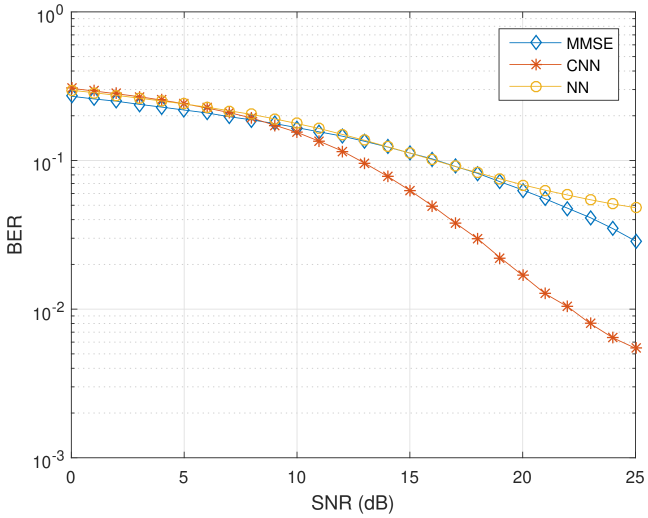


Fig. 8: Comparison of MMSE and CNN BER performances curves for channel C (without repetition).

VII. Conclusions

In this work, a new channel equalization method employing a CNN is proposed. As CNN's output generates only +1 and -1 values, equalization is performed for BPSK signals. Performance of the proposed equalizer is then compared with MMSE and MLP equalizers with and without repetition codes. Furthermore, 3 end 5 repetitions are used for each channel. Performance of the proposed method gives better results specifically at the high SNR values for non-repetition codes. For the 3-repetition coded signals, as shown in the Figures 9 and 10, the performance of CNN equalizer for channel B gives better performance and for channel C gives same performance as MMSE. But, on the other hand, as shown in the Figures 11 and 12 ,if the number of repetition is increased, the better the performance of MMSE equalizer. However, the matrix that represents the MMSE is quite a large one thus, including too many components therefore, requiring too much computational burden, since the matrix inversion for this purpose consumes too much time and memory when compared with the proposed CNN structure. One can see that, the MLP performance is the worst among aforementioned techniques for all cases.

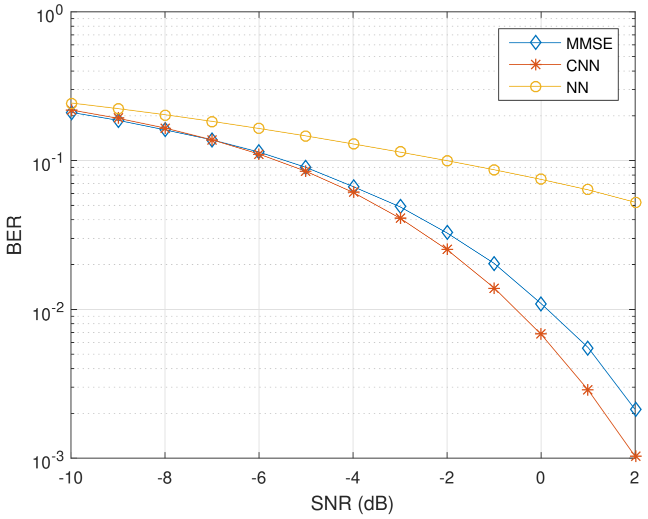


Fig. 9: Comparison of MMSE and CNN BER performances curves for channel B (with 3-repetition).

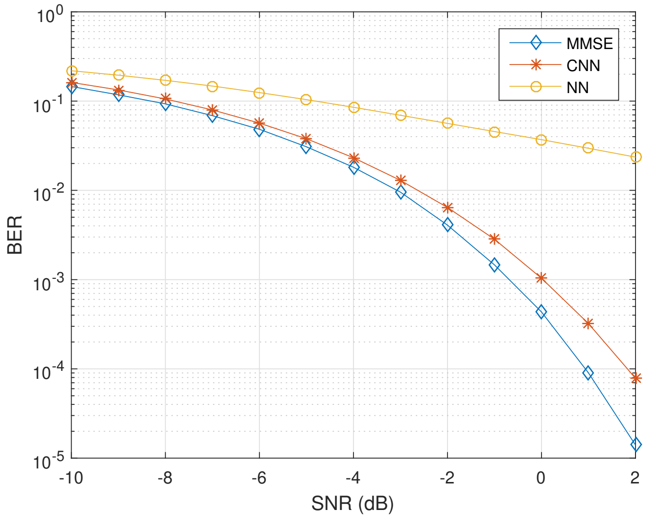


Fig. 11: Comparison of MMSE and CNN BER performances curves for channel B (with 5-repetition).

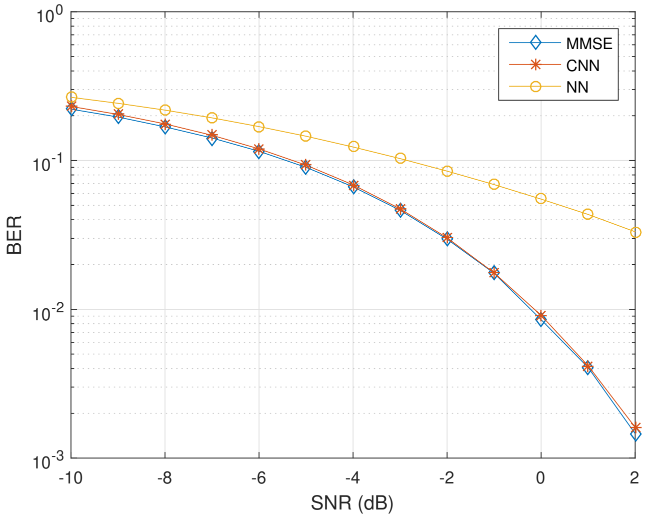


Fig. 10: Comparison of MMSE and CNN BER performances curves for channel C (with 3-repetition).

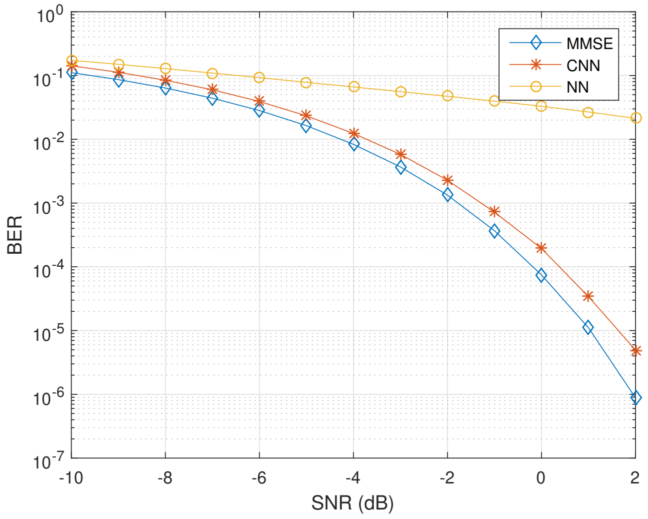


Fig. 12: Comparison of MMSE and CNN BER performances curves for channel C (with 5-repetition).

REFERENCES

[1] C. Berrou, A. Glavieux, P. Thitimajshima, “”Near Shannon limit error-correcting coding and decoding: Turbo-codes,” in *Proceeding of EEE International Conference on Communications*, Geneva, Switzerland, Nov. 30 - Dec. 4 1993, pp. 1064–1070.

[2] M.K. Lee, K. Yang, “Scheduling for an adaptive number of iterations in turbo equalizers combined with LDPC decoders,” *IEEE T Commun*, vol. 58, no. 10, pp. 2759–2764, 2010.

[3] M. Tuchler, R. Koetter, A. Singer, “Turbo equalization: Principles and new results,” *IEEE T Commun*, vol. 50, no. 5, pp. 754–767, 2002.

[4] S. Talakoub S., L. Sabeti, B. Shahrrava, M. Ahmadi, “An improved max-log-MAP algorithm for turbo decoding and turbo equalization,” *IEEE T Instrum Meas*, vol. 56, no. 3, pp. 1058–1063, 2007.

[5] C. He, L. Jing, R. Xi, H. Wang, F. Hua, Q. Dang, Q. Zhang, “ Time-Frequency Domain Turbo Equalization for Single-Carrier Underwater Acoustic Communications,” *IEEE Access*, vol. 7, pp. 73 324 – 73 335, 2019.

[6] N. Sharma, J. Sharma, S. Mishra, “Design & implementation of efficient turbo equalizer for noise reduction,” in *In: 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, Chennai, India, August 2017, pp. 2193–2198.

[7] P. J.G., *Digital Communications 4th ed.* McGraw Hill, 2001.

[8] B.Gupta, G. Gupta, D. S. Saini, “BER performance improvement in OFDM system with ZFE and MMSE equalizers,” in *In: 2011 3rd International Conference on Electronics Computer Technology*, Kanyakumari, India, April 2011, pp. 193–197.

[9] J. Lee, C. Beach, N. Tepedelenlioglu, “A practical radial basis function equalizer,” *IEEE T Neural Networ*, vol. 10, no. 2, pp. 450–455, 1999.

[10] L. Biao, BL. Evans , “Channel equalization by feed forward neural networks,” in *In: Proceeding of IEEE International Symposium on Circuits and Systems*, Orlando, FL, USA, May 30 - June 2 1999, pp. 587–590.

[11] R. Perfetti, “CNN for fast adaptive equalization,” *J Circ Theor*, vol. 21, no. 2, pp. 165–175, March 1993.

[12] A. Ozmen, B. Tander, “Channel equalization with cellular neural networks,” in *In: IEEE Mediterranean Electrotechnical Conference*, Valletta, Malta, April 2010, pp. 1597–1599.

[13] L.O. Chua, L. Yang, “A Cellular neural networks: Theory,” *IEEE Trans CAS*, vol. 35, no. 10, pp. 1257–1272, 1988.

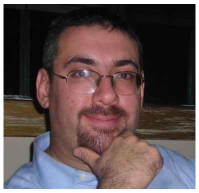
[14] G. Costantini, D. Casali, M. Carota, “Detection of moving objects in 2-D images based on a CNN algorithm and density based spatial clustering,” *WSEAS Trans CAS*, vol. 4, no. 5, pp. 440–447, 2005.

[15] K. S. M., *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993.

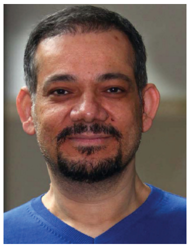
[16] T. Tozek, T. Roska, L.O. Chua, “ Genetic algorithm for CNN template learning,” *IEEE Trans. CAS I: Fund. Theo. and Apps*, vol. 40, no. 6, pp. 392–402, 1993.

[17] W. Ali, A. A. Ahmed, “ Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting,” *IET Information Security*, vol. 13, no. 6, pp. 659 – 669, 2019.

ATİLLA ÖZMEN was born in Batman, Turkey in 1971. He received the B.Sc., M.Sc. and Ph.D. degrees, all in Electronics Engineering, from Istanbul University, Istanbul, Turkey in 1993, 1996 and 2001 respectively. He served as a research assistant at the Department of Electrical and Electronics Engineering, Istanbul University from 1994 to 2001. Currently he is an assistant professor at the Electronics Engineering Department, Kadir Has University. His research interests are neural networks, image processing, signal processing and genetic algorithms.



BARAN TANDER was born in Ankara, Turkey on 1971. He received his BS, MS and PhD degrees in Electronics Engineering all from Istanbul University. He served as a research assistant at the same institution between 1995 and 2001. He is currently an assistant professor at Kadir Has University. His research interests include; neural networks, design and simulation of electronic circuits and image processing and is the author or co-author of more than 20 national and international articles about these subjects.



HABİB ŞENOL (M’87) received the B.S. and M.S. degrees from Istanbul University, Istanbul, Turkey, in 1993 and 1999, respectively, both in Electronics Engineering. He received the Ph.D. degree in Electronics Engineering from Isik University, Istanbul, Turkey, in 2006. He is currently with the faculty of Computer Engineering at Kadir Has University, Istanbul, Turkey. Dr. Senol spent the academic year 2007-2008 at the Department of Electrical Engineering, Arizona State University, USA, working on channel estimation and power optimization algorithms for Wireless Sensor Networks. Dr. Senol’s recent research interests include communication theory, advanced signal processing techniques and their applications to wireless electrical, underwater acoustic and optical communications.