

# On the Computational Complexities of Complex-valued Neural Networks

Kayol S. Mayer, Jonathan A. Soares, Ariadne A. Cruz, and Dalton S. Arantes

*Department of Communications, School of Electrical and Computer Engineering*

*Universidade Estadual de Campinas – UNICAMP, Campinas, SP, Brazil*

kayol@unicamp.br, j229966@dac.unicamp.br, a038481@dac.unicamp.br, dalton@unicamp.br

**Abstract**—Complex-valued neural networks (CVNNs) are nonlinear filters used in the digital signal processing of complex-domain data. Compared with real-valued neural networks (RVNNs), CVNNs can directly handle complex-valued input and output signals due to their complex domain parameters and activation functions. With the trend toward low-power systems, computational complexity analysis has become essential for measuring an algorithm’s power consumption. Therefore, this paper presents both the quantitative and asymptotic computational complexities of CVNNs. This is a crucial tool in deciding which algorithm to implement. The mathematical operations are described in terms of the number of real-valued multiplications, as these are the most demanding operations. To determine which CVNN can be implemented in a low-power system, quantitative computational complexities can be used to accurately estimate the number of floating-point operations. We have also investigated the computational complexities of CVNNs discussed in some studies presented in the literature.

**Index Terms**—Complex-valued Neural Networks, Low-power Systems, Quantitative Computational Complexity, Asymptotic Computational Complexity

## I. INTRODUCTION

Since the first steps of artificial neural models, a significant number of artificial neural network (ANN) architectures and learning methods have been proposed [1]–[3]. Interestingly, among these artificial neural networks, scarce attention is paid to the class of complex-valued neural networks (CVNNs) [4]. Unlike real-valued neural networks (RVNNs), CVNNs are capable of directly handling complex inputs and outputs [5]. As a result, CVNNs should be the natural choice for processing complex-valued signals, and they should also be explored for real-valued applications. Take for instance the XOR problem, derived from the two-dimensional “AND/OR” theorem. A single real-valued perceptron is unable to learn the XOR function. To solve the XOR problem, a three-layer RVNN is necessary at the very least. However, Minsky and Papert’s limitation can be circumvented using only a single complex-valued neuron [5]. Yet, the use of a single complex-valued neuron is not the only motivation; with CVNN architectures, it’s possible to enhance the functionality of neural networks, improve their performance, and reduce training time compared to RVNNs [6], [7]. Furthermore, it was recently proven by

Voigtlaender [8] that CVNNs also adhere to the universal approximation theorem.

For real-time systems, CVNNs have recently been implemented in photonic integrated circuits as an optical neural chip that obtained faster convergence and higher accuracy compared with RVNNs [9]. Not only in optical neural chips, CVNNs can also be efficiently implemented in graphics processing units (GPUs) and tensor processing units (TPUs) with matrix structures and field programmable gate arrays (FPGAs) with systolic arrays [10]. Additionally, with the development of adaptive computing platforms, such as PYNQ from Xilinx [11], CVNNs can be easily implemented in hardware using open-source Python libraries (e.g., RosenPy, developed by Cruz et al. [12]).

In many current applications, the most demanding algorithms are usually centralized in base stations with significant computational power. However, new technologies claim for desegregation, such as the Internet of Things (IoT), smart homes, and Industry 4.0, where a significant number of intelligent sensors are necessary [13]. Then, computational complexity analysis is crucial to choose the best approach for low-power systems.

For digital communication systems, CVNNs have also presented promising results for telecommunications, such as channel estimation and equalization, beamforming, detection, and decoding [14]–[21]. Liu et al. [14] proposed a CVNN based on extreme learning machines for channel estimation and equalization for OFDM systems. Enriconi et al. [15] demonstrated the beamforming tracking performance of a shallow phase transmittance radial basis function (PT-RBF) neural network under a dynamic military channel. Mayer et al. [16] employed a modified PT-RBF for transmitting beamforming, including the array currents into the CVNN architecture. Soares et al. [17] implemented a joint channel estimation and decoding for massive-MIMO communications using a shallow PT-RBF. Xu et al. [18] applied deep convolutional CVNNs for raw IQ signal recognition, achieving improved accuracy with lower computation complexity compared with RVNNs. Mayer et al. [19] compared some CVNN architectures for receiver beamforming operating with multiple users and interferences. Chu et al. [20] proposed

a channel estimation technique using a CVNN for optical systems operating with filter bank multicarrier with offset quadrature amplitude modulation (FBMC/OQAM). Soares et al. [21] proposed two inference learning approaches for channel estimation and decoding with CVNNs under highly dynamic channels.

In the literature, some CVNN computational complexities are addressed depending on the system implementation. In [22], the computational complexity of a shallow PT-RBF is presented in terms of a concurrent equalizer and a fuzzy controller. In [17], the computational complexity of a shallow PT-RBF is described as a function of the MIMO communication architecture. To the best of our knowledge, there is no work comparing the computational complexities of CVNNs, such as complex-valued feedforward NN (CVFNN) [23], split-complex feedforward NN (SCFNN) [24], multilayer feedforward NN based on multi-valued neurons (MLMVN) [25], complex-valued radial basis function (C-RBF) [15], fully complex-valued radial basis function (FC-RBF) [26], and PT-RBF [19].

This paper is an extension of Kayol S. Mayer’s Ph.D. Thesis [4], developed at the School of Electrical and Computer Engineering, Universidade Estadual de Campinas, in the area of Telecommunications and Telematics. In this context, this paper presents the quantitative and asymptotic computational complexities of the mentioned CVNNs in a comprehensive way, regardless of any specific application.

The remainder of this paper is organized as follows. Section II presents a brief discussion on CVNNs. Section III describes the quantitative and asymptotic computational complexities of CVNNs. In Section IV, we discuss the computational complexities of CVNNs proposed in the literature. Lastly, Section V concludes the paper.

## II. COMPLEX-VALUED NEURAL NETWORKS

One of the most studied CVNNs in the literature is the complex-valued feedforward neural network, a multilayer perceptron without feedback among layers in the forward step, adapted to directly process data in the complex domain [23]. CVFNNs can operate with fully-complex transcendental activation functions that satisfy the Cauchy-Riemann equations with relaxed conditions, such as circular, inverse circular, hyperbolic, and inverse hyperbolic functions. Also, an important and particular case of CVFNNs is the SCFNN, in which real and imaginary components are processed separately by holomorphic functions (i.e., analytic functions) in  $\mathbb{R}$  [24]. With similar architecture, but utilizing phase mappings onto unit circles as activation functions, the MLMVN is another relevant CVNN. In the MLMVN, the backpropagation algorithm is performed only using the multi-valued neurons error since no derivative is necessary because it is impossible to move in incorrect directions [25].

Based on a different CVNN architecture, the C-RBF neural network can also operate with complex numbers [15]. Due

to the C-RBF phase vanishing into the Euclidean norm of Gaussian neurons, Savitha et al. [27] proposed the FC-RBF neural network, where  $\text{sech}(\cdot)$  activation functions map  $\mathbb{C}^N \mapsto \mathbb{C}$  with Gaussian-like characteristics. Considering split-complex Gaussian neurons to circumvent any phase issue [22], Loss et al. [28] proposed the shallow and multiple-input single-output (MISO) PT-RBF. Recently, the PT-RBF has been extended to multiple outputs [17] and multiple layers [19].

In communication systems, the choice of architecture for complex-valued neural networks (CVNNs) can significantly affect performance. Notably, as detailed in Mayer [4], our research has shown that RBF-based architectures consistently outperform other CVNN architectures, especially in communication-related tasks such as channel equalization, beamforming, channel estimation, and decoding. This superior performance can be attributed to the inherent characteristics of RBF-based neural networks that make them well-suited for handling additive white Gaussian noise (AWGN), a common feature in communication systems. This phenomenon can be understood by the similarity between the activation functions of RBF-based CVNNs and the distribution function of AWGN noise — a distinction not found in other CVNNs like CVFNN, SCFNN, and MLMVN. This effect becomes more pronounced in challenging scenarios with lower signal-to-noise ratios (SNRs). However, this does not apply to FC-RBF, which becomes unstable in noisy situations. Further insights regarding performance and parameter estimation are available in [4], [19], [21].

## III. COMPUTATIONAL COMPLEXITIES

### A. Quantitative computational complexities

In order to estimate the computational complexity of an algorithm, one of the more straightforward and effective strategies is the mathematical operations analysis. Based on the CVFNN, SCFNN, MLMVN, C-RBF, FC-RBF, and PT-RBF architectures proposed in the literature, the mathematical operations are summarized into additions, multiplications, and activation functions. Although activation functions encompass a set of nonlinear functions that seem burdensome at first glance, they are not considered in our analysis since lookup tables can efficiently implement them [29]. For all CVNN architectures, the number of additions and multiplications are similar; thus, since the latter is much more demanding, additions are also not taken into account. For a complimentary analysis of the number of additions and activation functions of CVNNs, see [4].

The computational complexities of CVFNN, SCFNN, MLMVN, C-RBF, FC-RBF, and PT-RBF are assessed based on the number of inputs  $P$ , outputs  $R$ , layers  $L$ , and complex-valued artificial neurons per hidden layer  $N$ . Therefore, setting the number of inputs, outputs, and layers with neurons, we obtain the CVNNs computational complexities, depicted

in Tables I and II for shallow and deep CVNNs, respectively. In Table II, each CVNN layer is composed of  $I^{\{l\}}$  complex-valued neurons for  $l \in [1, 2, \dots, L-1]$ , except for the input layer where  $I^{\{l=0\}} = P$  and the output layer where  $I^{\{L\}} = R$ . Furthermore, for the deep PT-RBF, the number of bottleneck outputs is  $O^{\{l\}}$ . It is important to notice that the C-RBF and FC-RBF are not taken into account in Table II because they are only proposed for shallow architectures.

TABLE I  
SHALLOW CVNN COMPUTATIONAL COMPLEXITIES.

CVNN	Training	Inference
CVFNN	$N(8P + 12R + 8) + 8R$	$4N(P + R)$
SCFNN	$N(8P + 12R + 8) + 6R$	$4N(P + R)$
MLMVN	$N(8P + 12R + 16) + 12R$	$4N(P + R + 1) + 4R$
C-RBF	$N(4P + 6R + 5) + 4R$	$N(2P + 2R + 1)$
FC-RBF	$N(12P + 12R + 12) + 4R$	$4N(P + R)$
PT-RBF	$N(4P + 12R + 12) + 4R$	$2N(P + 2R + 1)$

### B. Asymptotic computational complexities

We assume that the neural networks have  $P$  inputs,  $R$  outputs,  $L$  hidden layers for deep CVNNs, and  $N$  neurons per layer. For the deep PT-RBF, the number of bottleneck

outputs is equal to the number of neurons per layer, i.e.,  $I^{\{l\}} = O^{\{l\}} = N$  for  $l \in [1, 2, \dots, L-1]$ , except for the output layer where  $O^{\{L\}} = R$ . The asymptotic computational complexities of shallow and deep CVNNs, based on Tables I and II, are depicted in Table III. In terms of asymptotic computational complexities, both training and inference have identical results, which is why the operation mode is not addressed in Table III. As the C-RBF and FC-RBF were only proposed for shallow architectures, their complexities are not specified for deep CVNNs.

From Table III, for shallow CVNNs with a number of neurons much lower than the number of inputs and outputs, i.e., first column, the computational complexities are asymptotically linear. However, for shallow CVNNs with a number of neurons proportional to the number of inputs and outputs, i.e., second column, and deep CVNNs with a number of neurons per layer much higher than the number of layers, i.e., third column, the computational complexities are asymptotically quadratic. Nevertheless, the computational complexities are asymptotically cubic for deep CVNNs with a number of neurons per layer proportional to the number of layers, i.e., the fourth column.

Relying on Table III asymptotic analysis, as shallow neural

TABLE II  
DEEP CVNN COMPUTATIONAL COMPLEXITIES.

CVNN	Training	Inference
CVFNN	$4 \sum_{l=1}^{L-1} I^{\{l\}} (2I^{\{l-1\}} + I^{\{l+1\}} + 2) + 8I^{\{L\}} (I^{\{L-1\}} + 1)$	$4 \sum_{l=1}^L I^{\{l\}} I^{\{l-1\}}$
SCFNN	$4 \sum_{l=1}^{L-1} I^{\{l\}} (2I^{\{l-1\}} + I^{\{l+1\}} + 2) + 2I^{\{L\}} (4I^{\{L-1\}} + 3)$	$4 \sum_{l=1}^L I^{\{l\}} I^{\{l-1\}}$
MLMVN	$4 \sum_{l=1}^{L-1} I^{\{l\}} (2I^{\{l-1\}} + I^{\{l+1\}} + 4) + 4I^{\{L\}} (2I^{\{L-1\}} + 3)$	$4 \sum_{l=1}^L I^{\{l\}} (I^{\{l-1\}} + 1)$
PT-RBF	$4 \sum_{l=1}^L I^{\{l\}} (O^{\{l-1\}} + 3O^{\{l\}} + 3) + 4 \sum_{l=1}^{L-1} O^{\{l\}} (I^{\{l+1\}} + 1) + 4O^{\{L\}}$	$2 \sum_{l=1}^L I^{\{l\}} (O^{\{l-1\}} + 2O^{\{l\}} + 1)$

TABLE III  
CVNN ASYMPTOTIC COMPUTATIONAL COMPLEXITIES.

CVNN	Shallow		Deep	
	$P = R \ll N$	$P = R \approx N$	$P = R = N \gg L$	$P = R = N \approx L$
CVFNN	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$
SCFNN	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$
MLMVN	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$
C-RBF	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$	–	–
FC-RBF	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$	–	–
PT-RBF	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$

– not applicable.

TABLE IV  
COMPUTATIONAL COMPLEXITIES OF CVNNs FOR APPLICATIONS PROPOSED IN THE LITERATURE.

CVNN	MIMO channel estimation and decoding [17]		FBMC/OQAM channel estimation in IM/DD [20]		Beamforming receivers with multiple users [19]		OFDM channel estimation and signal detection [30]	
	Training	Inference	Training	Inference	Training	Inference	Training	Inference
CVFNN	583,968	287,232	160	48	8,948	3,492	3,690,320	1,270,752
SCFNN	583,904	287,232	154	48	8,942	3,492	3,690,288	1,270,752
MLMVN	584,640	287,632	188	68	9,736	3,892	3,699,392	1,275,320
C-RBF	429,428	211,300	187	65	4,712	1,900	700,110	330,038
FC-RBF	1,268,528	422,400	432	120	12,012	3,600	1,987,144	657,792
PT-RBF	449,328	217,800	312	100	54,412	16,400	7,007,408	2,162,668

networks are usually designed with more neurons than inputs and outputs, thus shallow CVNNs have linear computational complexity when increasing the number of neurons. Notwithstanding, deep CVNNs have quadratic computational complexity with increasing the number of neurons because conventional deep neural networks operate with more neurons than hidden layers.

#### IV. USE CASES

To provide readers with a clear understanding, we present the computational complexities of CVNNs for some recent applications in communication systems proposed in the literature. Table IV depicts the computational complexities of CVNNs for MIMO channel estimation and decoding [17], FBMC/OQAM channel estimation in intensity modulation direct detection (IM/DD) [20], beamforming receivers with multiple users [19], and OFDM channel estimation and signal detection [30]. The computational complexities of training and inference have been computed using the equations presented in Tables I and II. The CVNN architectures were determined based on the descriptions presented in each referenced work. For instance, in [19], the CVNNs were designed with six inputs, three outputs, and 100 neurons. For comparison purposes, if the referenced work only discussed one CVNN architecture or exclusively employed RVNNs, we considered the number of inputs, outputs, layers, and neurons as parameters to calculate the equivalent computational complexity for the CVNNs. On the other hand, if the referenced work only considered deep architectures, we took into account the equivalent number of neurons (i.e., the sum of all neurons) to compute the computational complexity of the shallow CVNNs, specifically C-RBF and FC-RBF.

In Table IV, we observe that C-RBF achieved lower computational complexities in most of the applications, with the exception of [20]. When considering deep CVNN architectures, the PT-RBF presented higher computational complexity, as seen in [19], [30]. On the other hand, perceptron-based CVNNs exhibit intermediate computational complexity. Based on these results, we could recommend C-RBF as the primary choice for low-power communication systems

due to its lower complexity and satisfactory performance in noisy scenarios. However, in more demanding applications such as those required in base stations, the PT-RBF could also be employed, but at the cost of increased computing resources.

#### V. CONCLUSION

This paper offers a comprehensive analysis of the computational complexities associated with various complex-valued neural network (CVNN) architectures, including CVFNN, SCFNN, MLMVN, C-RBF, FC-RBF, and PT-RBF. Beyond simply cataloging these complexities, our work provides valuable technical insights that can guide both researchers and practitioners in the field. One of the key contributions of our analysis is the elucidation of how the asymptotic computational complexities of CVNNs evolve in relation to their architectural parameters, such as the number of inputs, outputs, neurons, and layers. By understanding these trends, practitioners can make informed decisions when selecting a CVNN architecture that aligns with their computational resource constraints. Moreover, our research goes beyond mere theoretical analysis. We demonstrate the practical utility of our findings by showcasing how quantitative computational complexities can be harnessed to accurately estimate the number of floating-point operations required for implementing CVNNs in communication systems. This insight empowers engineers and system designers to make informed choices when optimizing CVNNs for real-world applications, ultimately enhancing their efficiency and effectiveness.

#### ACKNOWLEDGMENTS

This work was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior — Brasil (CAPES) — Finance Code 001.

#### REFERENCES

- [1] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.
- [2] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *2017 34th International Conference on Machine Learning (ICML)*, vol. 70, 2017, pp. 1321–1330.

- [3] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 39–57.
- [4] K. S. Mayer, "Complex-valued neural networks and applications in telecommunications," Ph.D. dissertation, State University of Campinas, November 2022. [Online]. Available: <http://dx.doi.org/10.13140/RG.2.2.28044.80000/1>
- [5] T. Nitta, "Solving the XOR problem and the detection of symmetry using a single complex-valued neuron," *Neural Networks*, vol. 16, no. 8, pp. 1101–1105, 2003.
- [6] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 4, pp. 541–551, 2012.
- [7] S.-Q. Zhang, W. Gao, and Z.-H. Zhou, "Towards understanding theoretical advantages of complex-reaction networks," *Neural Networks*, vol. 151, pp. 80–93, 2022.
- [8] F. Voigtlaender, "The universal approximation theorem for complex-valued neural networks," *Appl. Comput. Harmon. Anal.*, vol. 64, pp. 33–61, 2023.
- [9] H. Zhang, M. Gu, X. D. Jiang, J. Thompson, H. Cai, S. Paesani *et al.*, "An optical neural chip for implementing complex-valued neural network," *Nature Communications*, vol. 12, no. 457, pp. 1–11, 2021.
- [10] R. Yang, J. Shen, M. Wen, Y. Cao, and Y. Li, "Integration of single-port memory (ISPM) for multiprecision computation in systolic-array-based accelerators," *Electronics*, vol. 11, no. 10, pp. 1–18, 2022.
- [11] Xilinx. PYNQ. [Online]. Available: <https://github.com/xilinx/pynq>
- [12] A. A. Cruz, K. S. Mayer, and D. S. Arantes, "RosenPy: An open source python framework for complex-valued neural networks," *SSRN*, pp. 1–18, 2022. [Online]. Available: <https://ssrn.com/abstract=4252610>
- [13] M. Pech, J. Vrchota, and J. Bednář, "Predictive maintenance and intelligent sensors in smart factory: Review," *Sensors*, vol. 21, no. 4, 2021.
- [14] J. Liu, K. Mei, X. Zhang *et al.*, "Online extreme learning machine-based channel estimation and equalization for OFDM systems," *IEEE Communications Letters*, vol. 23, no. 7, pp. 1276–1279, 2019.
- [15] M. P. Enriconi, F. C. C. de Castro, C. Müller, and M. C. F. de Castro, "Phase transmittance RBF neural network beamforming for static and dynamic channels," *IEEE Antennas Wireless Propagation Letters*, vol. 19, no. 2, pp. 243–247, 2020.
- [16] K. S. Mayer, J. A. Soares, and D. S. Arantes, "Complex MIMO RBF neural networks for transmitter beamforming over nonlinear channels," *Sensors*, vol. 20, no. 2, pp. 1–15, 2020.
- [17] J. A. Soares, K. S. Mayer, F. C. C. de Castro, and D. S. Arantes, "Complex-valued phase transmittance RBF neural networks for massive MIMO-OFDM receivers," *Sensors*, vol. 21, no. 24, pp. 1–31, 2021.
- [18] J. Xu, C. Wu, S. Ying, and H. Li, "The performance analysis of complex-valued neural network in radio signal recognition," *IEEE Access*, vol. 10, pp. 48 708–48 718, 2022.
- [19] K. S. Mayer, C. Müller, J. A. Soares, F. C. C. de Castro, and D. S. Arantes, "Deep phase-transmittance RBF neural network for beamforming with multiple users," *IEEE Wireless Communications Letters*, vol. 11, no. 7, pp. 1498–1502, 2022.
- [20] J. Chu, M. Gao, X. Liu *et al.*, "Channel estimation based on complex-valued neural networks in IM/DD FBMC/OQAM transmission system," *Journal of Lightwave Technology*, vol. 40, no. 4, pp. 1055–1063, 2022.
- [21] J. A. Soares, K. S. Mayer, and D. S. Arantes, "Semi-supervised ML-based joint channel estimation and decoding for m-MIMO with Gaussian inference learning," *IEEE Wireless Communications Letters*, pp. 1–5, 2023.
- [22] K. S. Mayer, M. S. de Oliveira, C. Müller, F. C. C. de Castro, and M. C. F. de Castro, "Blind fuzzy adaptation step control for a concurrent neural network equalizer," *Wireless Communications and Mobile Computing*, vol. 2019, pp. 1–11, 2019.
- [23] Z. Dong and H. Huang, "A training algorithm with selectable search direction for complex-valued feedforward neural networks," *Neural Networks*, vol. 137, pp. 75–84, 2021.
- [24] S. Scardapane, S. V. Vaerenbergh, A. Hussain, and A. Uncini, "Complex-valued neural networks with nonparametric activation functions," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 2, pp. 140–150, 2020.
- [25] I. N. Aizenberg, L. Sheremetov, L. Villa-Vargas, and J. Martínez-Muñoz, "Multilayer neural network with multi-valued neurons in time series forecasting of oil production," *Neurocomputing*, vol. 175, pp. 980–989, 2016.
- [26] R. Savitha, S. Suresh, and N. Sundararajan, "Metacognitive learning in a fully complex-valued radial basis function neural network," *Neural Computation*, vol. 24, no. 5, pp. 1297–1328, 2012.
- [27] R. Savitha, S. Suresh, and N. Sundararajan, "A fully complex-valued radial basis function network and its learning algorithm," *International Journal of Neural Systems*, vol. 19, no. 4, pp. 253–267, 2009.
- [28] D. Loss, M. C. F. De Castro, P. R. G. Franco, and F. C. C. De Castro, "Phase transmittance RBF neural networks," *Electronics Letters*, vol. 43, no. 16, pp. 882–884, 2007.
- [29] B. Zamanlooy and M. Mirhassani, "Efficient VLSI implementation of neural networks with hyperbolic tangent activation function," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 1, pp. 39–48, 2014.
- [30] 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, 2018.