This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/CITS58301.2023.10188712, 2023 International Conference on Computer, Information and Telecommunication Systems (CITS)

Deep Learning Polar Convolutional Parallel Concatenated (DL-PCPC) Channel Decoding for 6G Communications

Aya Kh. Ahmed
Department of Electronic and Electrical Engineering
College of Engineering, Design, and Physical Sciences
Brunel University London, London, United Kingdom
aya.ahmed@brunel.ac.uk

Hamed S. Al-Raweshidy
Department of Electronic and Electrical Engineering
College of Engineering, Design, and Physical Sciences
Brunel University London, London, United Kingdom
hamed.al-raweshidy@brunel.ac.uk

Abstract— **The new wireless generation 6G use of intelligent devices, sensors, and new applications like virtual reality and autonomous driving requires higher demands on the network with more users which needs higher data rate networks with minimum delay and less energy consumption. The current state for channel decoding does not meet the 6G requirements. In this paper, we design, evaluate, and proposes deep learning polar convolutional parallel concatenated (DL-PCPC) decoding, a new powerful decoding technique for 6G. The developed decoding technique dynamically reduces errors by 99.8%. It provides up to 80% better system efficiency than iterative decoding algorithms, with a 100% reduction in system delay. The novel proposes design works with a 6G communication frequency range of 300 and 400 GHz with terahertz data rates, providing correctly received data with a minimum amount of detected errors.**

Keywords—Channel Decoding, Concatenated Codes, Data Rate, Deep Learning, Delay, Terahertz, 6G.

I. Introduction

In the past few years, wireless communication generations have changed quickly, catching the attention of many countries and organisations [1]. Heterogeneous networking paradigms, satellite networks, and carrier networks are expected to be supported by the new wireless generation 6G to make transmissions more robust and reliable. Given that most current communication channels support Wi-Fi and cellular technologies within specific limitations, it is crucial to integrate 6G performance indicators to improve the user experience effectively [2]. One of the essential factors in wireless communication to meet 6G requirements is an error-free transmission with minimum delay and minimum energy consumption.

Wireless communication systems use channel coding techniques to correct transmission errors, whereas long iterative methods are used on the decoder side. Several channel coding techniques, such as turbo code, Low Density Parity Check (LDPC) code, and Polar Code (PC), have evolved in the last two decades, approaching the Shannon limit exceptionally closely and providing higher throughput and a lower bit error rate.

These codes, however, have some imperfections in terms of their long codeword length and long iterative decoding process. The 6G communication environment's Key Performance Indicators (KPIs) state that the new generation will provide; high reliability, low latency, and more bandwidth to send higher data rates up to Terahertz (THz) in real time.

New technological requirements have emerged for 6G to efficiently provide higher coverage and connections, including higher throughput, high dependability, low power consumption, and minimal encoding/decoding latency [3].

In our previous work [4], a new channel coding technique was exhaustively demonstrated to comply with 6G objectives, reaching the transmission limit of a terabit per second with higher reliability and 99.99% transmission throughput.

Moreover, Artificial Intelligent (AI) techniques are fundamental regarding their capability to support 6G KPIs. Deep Learning (DL) is introduced as a subset of Machine Learning (ML), which has recently become known as a robust set of methods that can produce impressive results in many research areas. DL is based on the architecture of neural networks and employs multiple layers ("deep") of artificial neurons [5]. DL has also been used in wireless communications, introducing a data-driven approach. Researchers are increasingly interested in how we can use DL in 6G communication.

In the past several years, techniques based on deep learning have been utilised to construct channel coders and decoders; these approaches have demonstrated exceptional performance across various communication channels. Research showed that using feed-forward deep learning to design modulated channel decoders for high-interference communication channels outperforms channels with standard modulation techniques [5].

However, previous research has shown that the current wireless generations cannot provide the adequate bandwidth capability expected from 6G. On the other hand, researchers focused on deep learning for channel coding of previous wireless generations; the so-called polar code performs better under the DL perspective [6], [7]. Deep learning is widely considered in channel encoding [8]-[11].

Our novel channel decoding, DL-PCPC, which uses a deep learning network, contributes to the 6G communication system with terahertz data rates by:

- This is the first time a deep learning decoding technique is used with parallel concatenated channel decoding (DL-PCPC). This novel technique complies with 6G terahertz transmission KPIs.
- The novel technique of DL-PCPC shows remarkable performance in minimising the decoding errors to zero compared with Successive Cancellation Decoding (SCD).

- The novel DL-PCPC has 80% system efficiency, better than SCD.
- DL-PCPC's performance indicators demonstrate error-free channel decoding and 99.8% system correctness.
- The novel technique maximises the data rate without additional system load or delay. This improvement can be stated as a near 100% improvement in the consumed time for the decoding process, with 14 iterations only being compared to 25 iterations using SCD.

The rest of this paper is organised as follows. Section II introduces the basics of parallel concatenated channel coding. Section III illustrates the architecture of our deep learning network model decoding scheme with the network training process. Section IV demonstrates the analysis of the results, and Section V is the conclusion.

II. Parallel Concatenated Channel Coding

This paper suggests a different way to find a robust decoding technique for parallel concatenated channel coding using deep learning. We utilised a deep learning decoding technique for polar convolution parallel concatenated (DL-PCPC) channel coding instead of the regular decoding algorithms. The standard decoding algorithm puts the network into undesirable delay and causes more burst errors, resulting in much lower reliability. The simulation results show that the deep learning model performs better than stand-alone SCD.

A. Channel Encoding

To fully understand the advantages and system performance for channel decoding with deep learning, our new parallel concatenated channel coding must first be demonstrated. The parallel concatenation formula uses data interleaver to reduce error bursts. The interleaved data is then applied to the first polar code encoder. At the same time, the original message is used directly as an input into the second encoder convolutional code, as shown in Fig.1. The resultant codeword from the first and second encoders is then multiplexed into a single codeword $\hat{c}$.

Polar code depends on two parameters N, the codeword length and k the message length, which $(N-k)$ equals the redundant bits r. The coding parameters we use are $N=1024$ and $k=528$.

The second encoder convolutional code has one more additional parameter D, the number of finite shift registers. Convolutional codes depend on the coding rate $R=k/N$ to identify the message's and codeword's relation. In our proposed code, we set the code rate to $1/3$ and $D=3$.

The encoding procedure for polar code is performed by applying the information sequences into a generator matrix defined as G, in which the generated matrix for $N=2$ equal to $G_2=\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$.

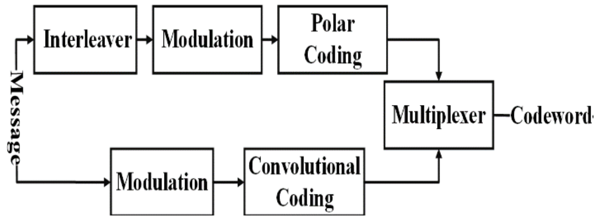

Fig. 1. Polar convolutional parallel concatenated code.

The generator matrix for any value of N is obtained using (1), which, $N=2^n, F=\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$, B_N is calculated using (2) and R_N is the data rate for the polar code.

$$G_N = B_N F^{\otimes n} \tag{1}$$

$$B_N = R_N\left(I_2 \otimes B_{\frac{N}{2}}\right) \tag{2}$$

Then the codeword is obtained by (3), where u is the original data stream:

$$\hat{c}_p = uG \tag{3}$$

The convolutional codeword is processed as full sequences length k into the finite shift register D. The codeword vector for every shift register is calculated using (4), (5), and (6), and the final codeword from the convolutional code is obtained using (7).

$$\hat{c}_c^{(1)} = \hat{c}_0^{(1)}\hat{c}_1^{(1)}....\hat{c}_{k-1}^{(1)} \tag{4}$$

$$\hat{c}_c^{(2)} = \hat{c}_0^{(2)}\hat{c}_1^{(2)}....\hat{c}_{k-1}^{(2)} \tag{5}$$

$$\hat{c}_c^{(3)} = \hat{c}_0^{(3)}\hat{c}_1^{(3)}....\hat{c}_{k-1}^{(3)} \tag{6}$$

$$\hat{c}_c = \hat{c}_0^{(1)}\hat{c}_0^{(2)}\hat{c}_0^{(3)}\hat{c}_1^{(1)}\hat{c}_1^{(2)}\hat{c}_1^{(3)}....\hat{c}_{k-1}^{(1)}\hat{c}_{k-1}^{(2)}\hat{c}_{k-1}^{(3)} \tag{7}$$

The resultant code word $\hat{c}$ is obtained by multiplexing $\hat{c}_p$ and $\hat{c}_c$.

B. Channel Decoding

The decoding process used in our previous work [4] depends on an iterative approach for serial and parallel techniques. However, the iterative approach worked better in serial concatenation; parallel concatenation decoding performance is poor.

As the obtained results show, the iterative decoding process consumes more time and energy resources since it involves more iterative steps from data demultiplexing and re-multiplexing and more iterative complicated decoding algorithms to process the decoding operation, as shown in Fig. 2.

Moreover, the original data word is not obtained in the parallel decoding algorithms until a specific iteration limit is reached; the applied application typically determines this limit. The iterative decoding technique depends on belief propagation (BP) and Viterbi sequential decoding algorithms,

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/CITS58301.2023.10188712, 2023 International Conference on Computer, Information and Telecommunication Systems (CITS)

which will overload the system with a more complicated and time-consuming decoding process without reaching the preferred error threshold.

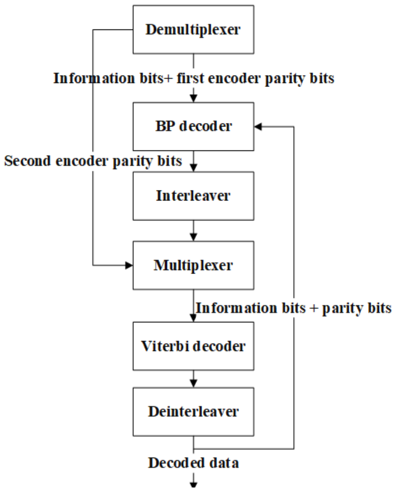

Fig. 2. Successive Cancellation Decoding (SCD).

III. Deep Learning Decoding Modeling

This section describes the newly introduced deep learning decoding technique network model, the training methodology, and the specifications. The new deep learning technique indicates that the coded data blocks are processed as one block for the number of network neurons. This procedure allows the decoding process to perform more bits per second than the iterative decoding.

A. Deep Network Decoding Design Architecture

The decoding process maps the received (coded) data to an original message estimation using the decode function $F:\hat{c}\rightarrow\hat{m}$. The received signal passes through a series of transposed convolutional layers and the ReLU activation function. The decoding function is designed to minimise the average distribution between the original message and the reconstructed (estimated) message by introducing a minimum average function as defined in $F_{\omega l}$ (8):

$$Minavg=\arg \mathrm{E}_{p(m_l,\hat{m}_l)}\left[d\left(m_l,\hat{m}_l\right)\right] \tag{8}$$

The minimum average function parameters are $d\left(m_l,\hat{m}_l\right)$ is the distribution measurement and $p\left(m_l,\hat{m}_l\right)$ is the probability distribution for the original and the reconstructed messages.

In the training process, the decoder will update the minimum average function iteratively and use the received coded signal. The decoding function will perform multiple iterations; each iteration will use the total codeword length into the DL network block parametrised by the weights ω_l, where the network function has 132 weighted neurons. The weighted inputs (output sequent) $q_l=\hat{c}.\omega_l$ are added with a bias b where at the final stage, the result is filtered with a rectified linear unit (ReLU).

$$\mathrm{Re}LU(q_l)=\max\{0,q_l\} \tag{9}$$

As shown in Fig. 3, the first decoder function $F_{\omega l}$ takes the codeword $\hat{c}_1$, a demultiplexed version of the codeword $\hat{c}_2$, and a prior version of the predicted original message $\hat{m}$; the first iteration initial value of the original message is set to 0.

The first decoder output is the sequent q_1, which will be the input for the second decoder block. The last iteration output from the decoder stage will feed into the ReLU function to predict the final message.

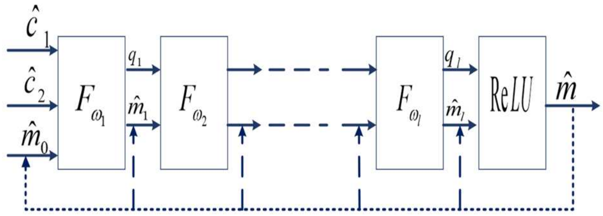

Fig. 3. Deep Learning Decoding Function.

B. Data Generation and Network Specification

The deep network model is tested and validated to extract the data stream features with a 200,000 codeword dataset for parallel concatenated decoding. Each code word $\hat{c}$ is processed by decoding blocks F_ω with the parameters $\hat{c}_1$ $\hat{c}_2$ and $\hat{m}$. In other words, we decode $\hat{c}$ for a known coding scheme with the abovementioned initial parameters.

Each block represents one of the predictions in the prediction stream. The average value for each prediction is calculated as in (8), and the one with the minimum average distribution is the most accurate prediction.

C. Data Preparation and System Validation

After code word generation is complete, the generated dataset is split into three parts, 20% for validation data, 20% for test data, and 60% as network training data. The tested decoding data is performed with the 6G terahertz frequency range; the tested frequency is 300 and 400 GHz.

The training procedure optimises the weight of the trainable network parameters using back-propagation, and the optimisation technique used is the Nesterov-accelerated adaptive moment estimation (Nadam) algorithm. Moreover, we employ binary cross entropy loss as our loss function because it improves the performance of the deep network when data imbalance is unpredictable.

The Nadam algorithm combines mini-batch gradient descent with Nesterov momentum to quicken the learning process. To correspond to our activation function, the weights of the hidden layers are initialised using the lecun normal initialiser.

D. Training Process

Training settings are set to perform for multiple epochs, where the training dataset is randomly shuffled and fed into the model in every epoch. The training process will end when the minimum average function stops changing for a maximum of 25 consecutive epochs, as in our previous iteration limit for the decoding algorithm.

The DL-PCPC network consists of six convolutional fully connected function layers. ReLU is used as an activation

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/CITS58301.2023.10188712, 2023 International Conference on Computer, Information and Telecommunication Systems (CITS)

function and batch normalisation to improve the overall system performance and reduce the internal multivariable shifts.

The DL-PCPC network is demonstrated in Fig. 4; the codeword data is processed into a demultiplexer, and the output is fed into the deep learning network iterative blocks from Fig. 3. The output from the deep network will be multiplexed before the final original data estimation $\hat{m}$ is established.

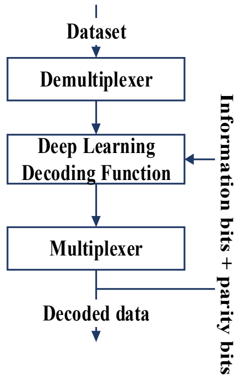

Fig. 4. Deep Learning Decoding Design Architecture (DL-PCPC).

IV. Results Analysis

We evaluate the minimum decoding error based on the new optimisation decoding technique. As shown in Fig. 5, our new deep learning decoding for concatenated parallel decoding reaches the confidence level of minimum error only after ten iterations. The best point at which the system detects no errors is reached on iteration number 14, which indicates 50% better performance than SCD used before, where the no errors limit is reached on iteration number 25.

The simulation results show error-free performance with 300 GHz and 400 GHz, providing an error-free channel decoding algorithm with 99.8% correct data within terahertz data with 6G wireless communication.

To facilitate the power of the new deep learning decoding approach, the results of successive cancellation decoding are demonstrated in Fig. 6. SCD fails to deliver an accepted decoding error for both 300 GHz and 400 GHz frequencies.

The minimum decoding error achieved in SCD remains within 30% of errors for the full-time transmission and 25 iterations.

The new communication systems are going towards the 6G frequency band with terahertz data throughput. As a result, this iterative decoding algorithm is not promising as our DL-PCPC to be considered in 6G communication.

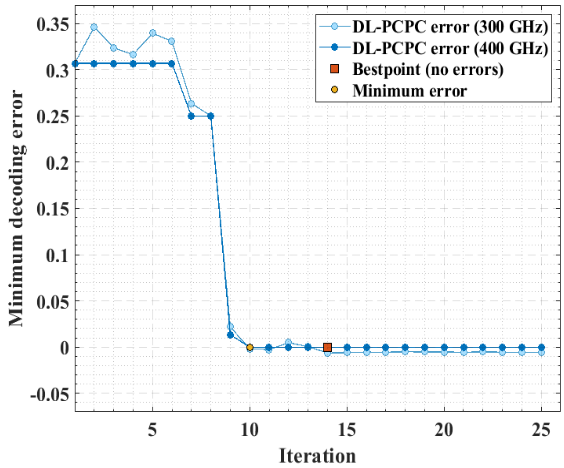

Fig. 5. Performance of DL-PCPC Decoding Error.

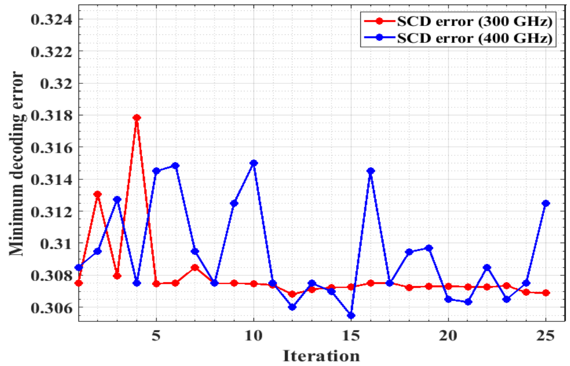

Fig. 6. Performance of SCD Decoding Error.

Another performance indicator that proves the validity of our system is the data rate of correctly received data stream against received data stream with errors. Fig. 7 shows the performance of our DL-PCPC through different iterations; as the number of iterations increases, the percentage of correctly received data increases as well. Only five iterations can provide up to 94% correct data using DL-PCPC.

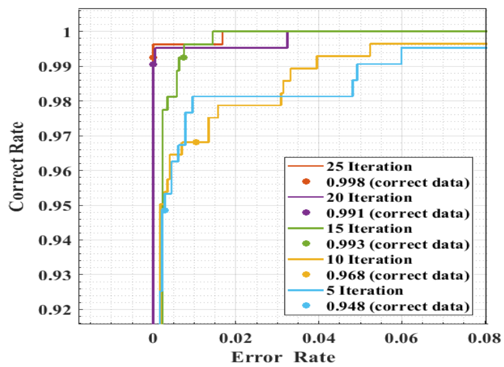

Fig. 7. Performance of DL-PCPC Data Rate of Correct Data.

These performances cannot be achieved using the successive cancellation decoding as experienced in our previous work. At the maximum iteration reach of 25, the correctly decoded data is up to 99.8%.

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/CITS58301.2023.10188712, 2023 International Conference on Computer, Information and Telecommunication Systems (CITS)

System Energy efficiency in (Tbit/Joule) is an essential key performance indicator to facilitate system proficiency.

Fig. 8 shows the energy efficiency of DL-PCPC against SCD for 25 iterations. Results show that DL-PCPC surpasses SCD by 80% at the last iteration, where the energy efficiency for the SCD algorithm starts to drop sharply after ten iterations to reach only 20% system energy efficiency at the last iteration.

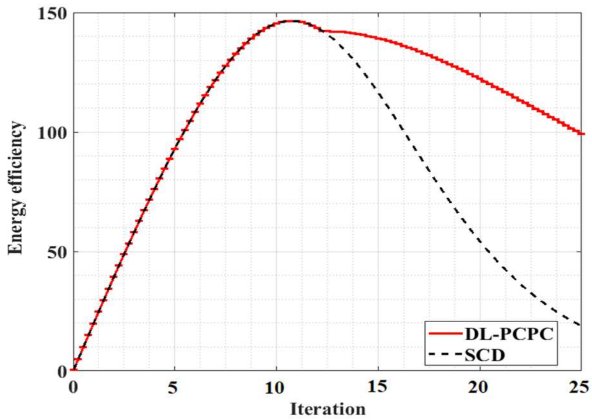

Fig. 8. Performance of DL-PCPC Energy Efficiency.

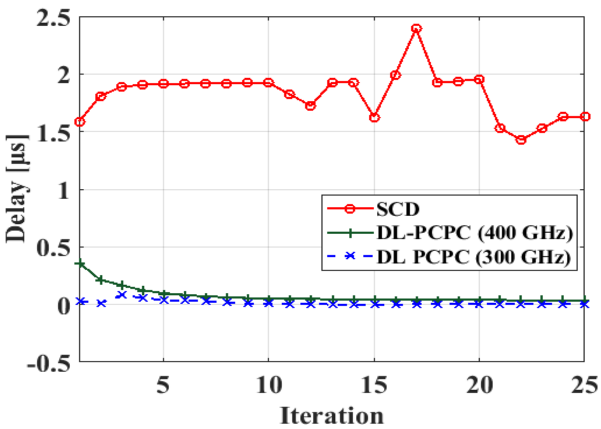

Fig. 9. System Delay for DL-PCPC and SCD.

System delay is a significant performance indicator for validating deep learning design efficiency. Fig. 9 shows the massive difference in system delay using DL-PCPC compared with SCD. The new deep learning design mange to minimise the system delay to 0.003µs for the overall decoding process at the last iteration.

On the opposite of that, SCD, where the delay goes no under 1µs for the whole decoding time. This improvement can be stated as near 100% improvement in the consumed time for the decoding process.

V. Conclusion

In this paper, we designed and proposed a new channel decoding technique for 6G communicanotion based on our previous concatenated channel coding.

Our new decoding technique DL-PCPC which incorporates a deep learning approach into the decoding technique proves its validity to work in the 6G communication system frequency range with terahertz data rates. DL-PCPC deliver data with 80% system efficiency, more than the usualy used iterative decoding algorithms.

The performance indicators prove that DL-PCPC can provide channel decoding with no errors and 99.8% system accuracy. The designed system provides the maximum archivable data rate with minimum load and only 14 iterations. Moreover, system delay maintains to remain at a deficient level compared with frequently used iterative decoding.

It is increasingly encouraging to outperform the 6G performance metrics through deep learning for higher data block length. However, the results are expected to improve as the data block length increases; it cannot be confirmed until tested on the system.

Additionally, neural codes must be taught how to use memory throughout encoding and decoding together to achieve superior performance for extended data block lengths. We provide evidence showing how deep learning can improve the decoding process for the performed code word data length, which is expected to deliver the same promising performance for greater code word lengths.

References

[1] M. Vaezi et al., “Cellular, Wide-Area, and Non-Terrestrial IoT: A Survey on 5G Advances and the Road Toward 6G,” IEEE Commun. Surv. Tutorials, vol. 24, no. 2, pp. 1117–1174, 2022.

[2] X. Xu, S. Wu, D. Dong, J. Jiao and Q. Zhang, “High-Performance Short Polar Codes: A Concatenation Scheme Using Spinal Codes as the Outer Code,” in IEEE Access, vol. 6, pp. 70644-70654, 2018, doi: 10.1109/ACCESS.2018.2879827.

[3] M. Z. Chowdhury, M. Shahjalal, S. Ahmed and Y. M. Jang, “6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions,” in IEEE Open Journal of the Communications Society, vol. 1, pp. 957-975, 2020, doi: 10.1109/OJCOMS.2020.3010270.

[4] A. K. Ahmed and H. S. Al-Raweshidy, “Performance Evaluation of Serial and Parallel Concatenated Channel Coding Scheme With Non-Orthogonal Multiple Access for 6G Networks,” in IEEE Access, vol. 10, pp. 39681-39690, 2022, doi: 10.1109/ACCESS.2022.3166943.

[5] V. P. Rekkas, S. Sotiroudis, P. Sarigiannidis, S. Wan, G.K. Karagiannidis, S. K. Goudos, “Machine Learning in Beyond 5G/6G Networks—State-of-the-Art and Future Trends”, in *Electronics*. 2021; 10(22):2786. https://doi.org/10.3390/electronics10222786

[6] A. Irawan, G. Witjaksono and W. K. Wibowo, “Deep Learning for Polar Codes over Flat Fading Channels,” 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2019, pp. 488-491, doi: 10.1109/ICAIIC.2019.8669025.

[7] B. He, S. Wu, Y. Deng, H. Yin, J. Jiao and Q. Zhang, “A Machine Learning Based Multi-flips Successive Cancellation Decoding Scheme of Polar Codes,” 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 2020, pp. 1-5, doi: 10.1109/VTC2020-Spring48590.2020.9128875.

[8] S. Dehdashtian, M. Hashemi and S. Salehkaleybar, “Deep-Learning-Based Blind Recognition of Channel Code Parameters Over Candidate Sets Under AWGN and Multi-Path Fading Conditions,” in IEEE Wireless Communications Letters, vol. 10, no. 5, pp. 1041-1045, May 2021, doi: 10.1109/LWC.2021.3056631.

[9] E. Bourtsoulatze, D. Burth Kurka and D. Gündüz, “Deep Joint Source-Channel Coding for Wireless Image Transmission,” in IEEE Transactions on Cognitive Communications and Networking, vol. 5, no. 3, pp. 567-579, Sept. 2019, doi: 10.1109/TCCN.2019.2919300.

[10] M. Yang, C. Bian and H. -S. Kim, “Deep Joint Source Channel Coding for Wireless Image Transmission with OFDM,” ICC 2021 - IEEE International Conference on Communications, 2021, pp. 1-6, doi: 10.1109/ICC42927.2021.9500996.

[11] K. Chahine, N. Ye and H. Kim, “DeepIC: Coding for Interference Channels via Deep Learning,” 2021 IEEE Global Communications Conference (GLOBECOM), 2021, pp. 01-06, doi: 10.1109/GLOBECOM46510.2021.9685430.K. Chahine, N. Ye, and H. Kim, “DeepIC: Coding for Interference Channels via Deep Learning; DeepIC: Coding for Interference Channels via Deep Learning,” 2021 IEEE Glob. Commun. Conf., 2021.