**IAES International Journal of Artificial Intelligence (IJ-AI)**
Vol. 15, No. 1, February 2026, pp. 547∼558
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i1.pp547-558


# An efficient method to improve machine learning decoders using automorphisms group

**Imrane Chemseddine Idrissi, Said Nouh, El Mehdi Bellfkih, Mohammed El Assad, Abdelaziz Marzak**
Department of Mathematics and Informatics, Faculty of Science Ben M'sick, Hassan II University of Casablanca, Casablanca, Morocco

**Article Info**

***Article history:***

Received Oct 29, 2023
Revised Nov 14, 2025
Accepted Jan 10, 2026

***Keywords:***

Automorphisms group
Bose-Chaudhuri-Hocquenghem codes
Error correcting code
Machine learning for decoding
Multilayer perceptron
Syndrome decoding

**ABSTRACT**

The decoding of error-correcting codes (ECCs) is a critical aspect of communication systems, yet traditional decoding techniques can often be computationally demanding or ineffective for certain codes, necessitating innovative approaches. In this study, we introduce a hybrid approach that combines machine learning and automorphism techniques to optimize the decoding process. Specifically, we train multilayer perceptron (MLP) models to learn the mapping between error syndromes and their corresponding errors. While these models exhibit robust learning capabilities, their performance sometimes does not reach 100%. To mitigate this limitation, we exploit the automorphism group of the code—a set of structure-preserving transformations—to convert the errors that the MLP struggles to decode into ones it can process more effectively. We use a minimum number of $p$ permutations, pre-calculating and storing all possible automorphisms to ensure computational efficiency. Our experimental results reveal that this hybrid approach substantially enhances the decoding performance of the MLP model, presenting a promising avenue for decoding ECCs. Importantly, this approach is not limited to MLP models and can be applied to any machine learning model with a learning score less than 100%, broadening its applicability and impact. By integrating machine learning with traditional algebraic coding theory, we propose a new paradigm that holds the potential to revolutionize the design of decoding systems, making them more efficient and effective.



***Corresponding Author:***

Imrane Chemseddine Idrissi
Department of Mathematics and Informatics, Faculty of Science Ben M'sick
Hassan II University of Casablanca
Casablanca, Morocco
Email: imran.chems@gmail.com

## 1. INTRODUCTION

Communication channels are pivotal in transmitting information between a transmitter and a receiver across various applications, from telecommunication systems to computer networks and wireless communication systems. Understanding communication channels is key to grasping the limitations and possibilities of communication systems and designing efficient and robust techniques for encoding and decoding information [1]. A communication channel is the medium through which information travels from the transmitter to the receiver. Communication channels are broadly categorized into two types: wired and wireless. Wired channels encompass copper cables, optical fibers, and coaxial cables, while wireless channels involve transmitting information through airwaves using radio frequency (RF) or optical signals [2].

The performance of a communication channel is profoundly affected by the presence of noise and other impairments that can degrade the quality and reliability of the transmitted information [3]. Noise sources include thermal noise, shot noise, and interference from other signals, whereas channel impairments may comprise fading, multipath propagation, and signal attenuation [4]. The concept of channel capacity, introduced by Shannon, sets a fundamental limit on the maximum rate at which information can be transmitted reliably over a communication channel [1]. This limit is dependent on the channel's signal-to-noise ratio (SNR) and bandwidth, which are crucial factors in determining the performance of communication systems [3].

Error-correcting codes (ECCs) and modulation techniques are extensively employed to enhance the reliability and efficiency of communication systems in the presence of channel noise and impairments [5]. Modern communication systems also incorporate adaptive techniques, such as adaptive modulation and coding, to optimize their performance based on varying channel conditions [6]. In conclusion, comprehending communication channels and their characteristics is essential for designing and optimizing communication systems to achieve reliable and efficient transmission of information across various applications and environments.

Since all channels are noisy and unreliable, transmitting binary data over the aforementioned channels can cause errors in messages by changing 0 to 1, or vice versa. Here, ECCs make it possible to extract the original binary data from the altered binary data due to noise. In other words, the primary objective of ECCs is to enable reliable digital communication over unreliable channels, as shown in Figure 1.

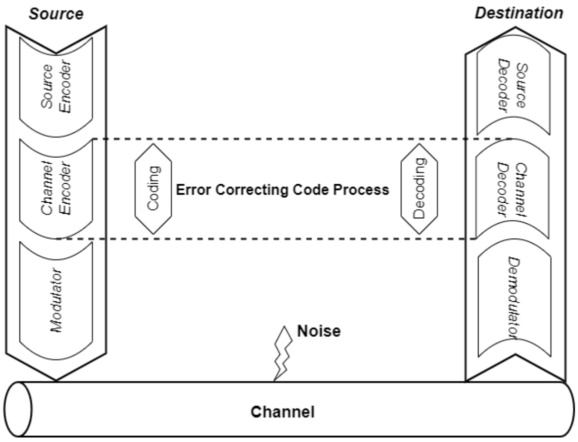


Figure 1. Communication model

As an example, we discuss the fundamental idea behind ECC: communication over unreliable channels. To reduce the probability of altering the original messages, we add redundancy, making the transmitted messages easier to distinguish from each other. There are various classes of ECC; however, the main goal of any ECC is to recover the original message using different types of techniques, such as algebraic, heuristic, meta-heuristic, or machine learning techniques.

The use of machine learning to enhance communication networks is not a new concept. The information theory and machine learning communities have long shared a nebulous belief that they are one and the same since they employ similar statistical techniques to address comparable issues. This belief was first expressed by MacKay [7]. ECCs, as shown in Figure 2 can be broadly classified into two major families: block codes and convolutional codes. These two families have distinct characteristics and are used in different applications depending on their specific advantages.

Block codes and convolutional codes are two classes of ECCs used in digital communication systems. Block codes operate on fixed-size blocks of data, encoding each block independently and correcting errors upon decoding [1]. Examples of block codes include Hamming codes, Reed-Solomon codes, and Bose-Chaudhuri-Hocquenghem (BCH) codes, each with specific error-correction capabilities and applications [8], [5]. Convolutional codes, on the other hand, work on continuous streams of data, using a different encoding scheme involving convolving the data stream with generator polynomials [9]. They are widely used in digital communication systems and often combined with block codes for enhanced error-correction performance [7], [5].

In our research paper, we have organized our exploration of decoding methods into distinct sections. We begin with a review of existing approaches, including algebraic decoders, heuristic and meta-heuristic decoders, and machine learning decoders, in section 2. Next, in section 3, we introduce novel machine learning techniques, focusing on the multilayer perceptron (MLP) decoder and its enhanced version, the improved multilayer perceptron decoder (MLPDec) with an automorphism set. Section 4 presents empirical findings, including performance comparisons with other methods, while the section 5 summarizes our key findings and suggests future research directions.

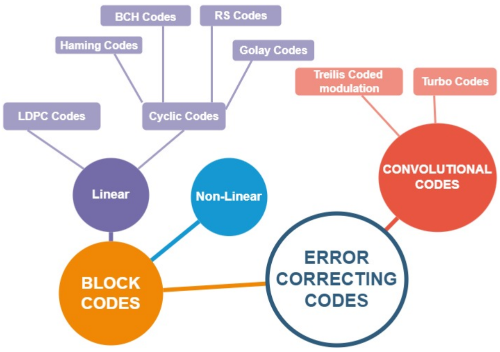


Figure 2. Error correcting codes classification

## 2. RELATED WORK

ECCs are essential tools in modern digital communication systems, providing a means to detect and correct errors that may occur during data transmission. Among various ECCs, BCH codes stand out for their capability to correct multiple errors while maintaining relatively low complexity. BCH codes are a class of cyclic ECCs with the ability to correct multiple errors in a data block. They were independently discovered by Bose and Chaudhuri, and by Hocquenghem in the early 1960s. Since then, BCH codes have been widely used in various communication and storage systems due to their excellent error-correcting capabilities and efficient decoding algorithms.

In ECCs, many approaches have been developed to improve the coding and decoding process. Decoders can be classified based on decoding algorithms into four major classes: algebraic, heuristic, meta-heuristic, and machine learning decoders (Figure 3). These classifications can help to categorize decoding algorithms based on their properties, computational approaches, and adaptability. Each type of decoder has its advantages and disadvantages, and the choice between them depends on the specific requirements of the application and the desired trade-offs between performance, complexity, adaptability, and implementation [5].

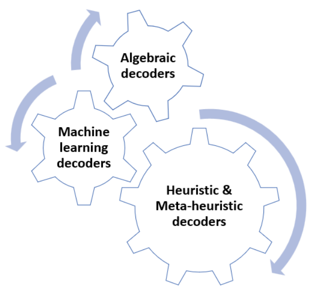


Figure 3. Error correcting codes techniques

### 2.1. Algebraic decoders

These decoders are based on algebraic techniques and rely on the underlying mathematical structure of the ECC. They use deterministic algorithms to find and correct errors. Examples of algebraic decoding techniques include Berlekamp-Massey, Peterson, and Euclidean algorithms for BCH and Reed-Solomon codes. Algebraic decoders typically have well-defined performance and complexity characteristics but may have limited adaptability to different channel conditions or code structures. Some recent developments and optimizations in algebraic decoding techniques highlight ongoing efforts to improve efficiency and performance of algebraic decoders in various communication systems and applications. A novel algebraic decoding technique for BCH codes using Gröbner bases is proposed. The proposed algorithm offers efficient and flexible decoding process while also providing a better understanding of mathematical properties of BCH codes. Li and Salehi [10] presents algebraic soft-decision decoding algorithm for concatenated Reed-Solomon codes. The proposed algorithm reduces the decoding complexity while maintaining good performance in the presence of noise and channel impairments. Puchinger *et al.* [11] presents an enhanced algebraic-geometry decoding technique for Hermitian codes, which improves decoding performance while simplifying the decoding procedure. For Reed-Solomon codes, there is a novel algebraic soft-decision decoding algorithm that seeks to increase error-correcting efficiency while lowering decoding complexity and computing cost [12].

### 2.2. Heuristic and meta-heuristic decoders

Heuristic decoders utilize simplified problem-solving strategies, often relying on rules of thumb or educated guesses to approximate optimal solutions during the decoding process. These approaches are particularly attractive due to their relatively low computational complexity when compared to algebraic decoders. While they can yield good performance in many scenarios, their effectiveness is heavily influenced by the quality of the chosen heuristic, and they may fail to achieve optimal decoding performance in more complex or noisy environments. Notable examples of heuristic decoders include chase decoding and generalized minimum distance (GMD) decoding algorithms, which are commonly applied to Reed-Solomon codes.

In contrast, meta-heuristic decoders employ more generalized and flexible optimization frameworks capable of solving a broad class of decoding problems. These methods are inspired by natural processes and include algorithms such as particle swarm optimization, simulated annealing, and genetic algorithms. The primary advantage of meta-heuristic approaches lies in their adaptability to various code structures and dynamic channel conditions, allowing for improved performance in non-ideal or evolving transmission environments. Nevertheless, the increased performance often comes at the cost of higher computational complexity and the necessity for careful parameter tuning to achieve satisfactory results.

An illustrative example is the artificial reliabilities based decoding genetic algorithm (ArDecGA) decoder, which applies genetic algorithm principles to the decoding of BCH codes. This method searches the solution space of candidate codewords, evolving them iteratively based on fitness scores that reflect how closely each candidate approximates the correct codeword [13]. Another efficient decoding family includes the hard and soft decoder (HSDec) [14] and hard weights decoder (HWDec) [15], which prioritize decoding speed and simplicity. These decoders are particularly beneficial in low-latency, real-time systems, although their reliance on hash tables can result in significant memory requirements—posing potential limitations in embedded or resource-constrained applications. Overall, these decoding strategies represent a valuable spectrum of alternatives to traditional methods. The continued development of heuristic and meta-heuristic decoders highlights the dynamic nature of ECC research, showcasing novel trade-offs between decoding accuracy, computational demands, and system constraints.

### 2.3. Machine learning decoders

In recent years, deep learning has become increasingly influential in the field of ECC decoding. Among the most studied cases are deep learning-based decoders for BCH codes, which involve training neural networks to learn the mapping between noisy received codewords and their corresponding original codewords. These networks, typically composed of multiple layers with non-linear activation functions, are capable of identifying intricate patterns within data that traditional methods may overlook.

To effectively train these neural models, large datasets consisting of noisy input codewords and their known transmitted outputs must be generated. The learning process is centered around minimizing the error between the predicted and actual codewords. Once trained, the network can generalize this mapping to new, unseen data by selecting the most probable codeword, enhancing decoding reliability under variable noise conditions. A significant benefit of such models is their adaptability to a wide range of channel conditions and noise characteristics. However, this adaptability comes at the cost of substantial computational resources, particularly during the training phase, and a strong dependence on the quality and size of the dataset used.

Several works have been proposed to enhance the effectiveness of these models. For example, Nachmani *et al.* [16] introduces multiple neural architectures and training techniques that outperform traditional algebraic decoders in high-noise scenarios. Similarly, Kim *et al.* [17] explores various learning strategies and network structures applicable to multiple code families. These studies collectively demonstrate promise of machine learning approaches in improving decoding performance, particularly when traditional algorithms face performance degradation due to complex or unpredictable noise patterns.

Another influential study analyze the use of deep neural networks for channel decoding, comparing their method against classical approaches [18]–[20]. Their results underscore both the strengths and limitations of neural decoders—showing improved performance but also pointing to challenges in scalability and interpretability. Extending this research, Cammerer *et al.* [21] presents a deep learning strategy for polar codes, leveraging partitioning methods to manage longer code lengths efficiently. Though the focus is on polar codes, the proposed techniques are likely transferable to BCH code decoding, suggesting broader applicability of these innovations. In addition to deep neural networks, other machine learning models such as logistic regression decoders (LRDec) have been explored. The LRDec model [22] uses list decoding and combines algebraic and combinatorial strategies to achieve robust performance, particularly under high-error conditions. Nonetheless, its computational burden increases with code length and error density, presenting practical trade-offs.

Beyond classical settings, recent developments have also extended machine learning and error correction into the quantum domain. Chao and Reichardt [23] propose a quantum error correction approach that requires only two ancillary qubits, offering a lightweight and practical framework for quantum systems. Kribs *et al*. [24] introduce operator quantum error correction as a theoretical framework for managing errors in quantum computations. Meanwhile, Terhal [25] highlights the critical role of quantum error correction in maintaining coherence within quantum memory architectures.

In summary, the body of recent research illustrates the substantial progress made in machine learning-based decoding. Whether through deep neural networks or hybrid techniques like LRDec, modern approaches are pushing the boundaries of error correction. These methods show significant promise in increasing decoding efficiency and robustness while also introducing new challenges in terms of computational cost and model complexity.

## 3. PROPOSED DECODERS

### 3.1. Machine learning techniques

Machine learning algorithms, specifically MLP neural networks, have transformed data processing and analysis by enabling various applications across different fields. One significant application is data decoding, where MLPs, a category of artificial neural networks (ANNs) as shown in Figure 4, have become widely adopted due to their capability to effectively model complex, nonlinear relationships between inputs and outputs. In our study, the initial focus was on developing a model with optimal performance using MLP by selecting appropriate model parameters during the training phase for linear BCH codes. The first step in creating a decoder model for linear BCH codes involves defining the inputs (X) and outputs (Y) [22].

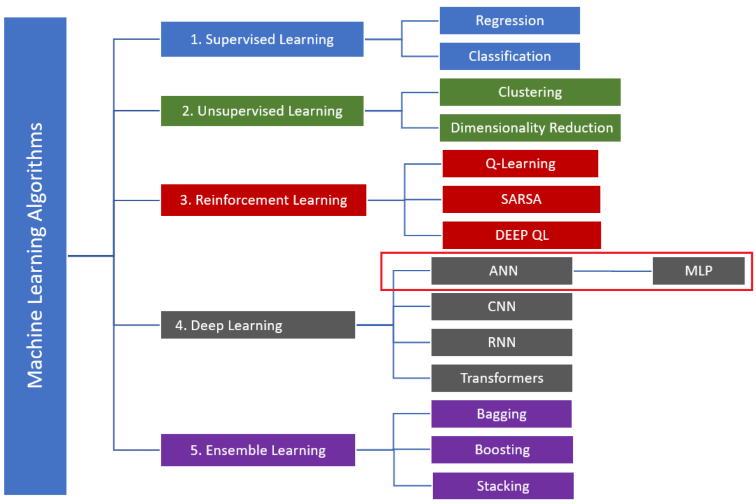


Figure 4. Machine learning algorithms classification

Initially, a list of all possible errors of length ‘n’ with a weight less than or equal to ‘t’ (the error-correcting capability of the code) must be created. This list encompasses all the classes of the model. Moreover, to differentiate between correctable and uncorrectable errors, a null value was added as output for uncorrectable errors in the list, and the received word (with a detectable weight error) was marked as uncorrectable. The objective here is to preserve the integrity of the word if the error weight is greater than the code’s error-correcting capability. By incorporating this approach, our model not only identifies the errors but also differentiates between correctable and uncorrectable errors, which is crucial for maintaining the integrity of the transmitted data. This methodology ensures that the decoder model developed is robust and efficient in correcting errors in linear BCH codes.

### 3.2. Multilayer perceptron decoder

The MLP is a foundational architecture in neural networks, comprising an input layer, one or more hidden layers, and an output layer. The hidden layers employ nonlinear activation functions, such as rectified linear unit (ReLU) or sigmoid, which empower the network to capture complex, non-linear relationships within the data. This makes MLPs particularly adept at handling noisy or incomplete inputs. Furthermore, the feedforward nature of the MLP architecture ensures efficient training and enables generalization to unseen data—a crucial feature in decoding applications. In the context of ECCs, Nachmani *et al.* [16]. demonstrated the successful application of MLPs to the decoding of linear codes, underscoring their effectiveness in real-world communication systems.

We propose a novel MLP architecture for hard decoding of BCH codes. The model learns the mapping between syndromes and correctable errors to enable error correction in noisy communication channels. This approach leverages MLP pattern recognition capabilities while preserving BCH structural properties for robust decoding. For training, input data $X$ consists of syndromes for BCH $(n, k, t)$ codes, where each $x_i$ has length $n - k$. Output data $Y$ contains binary error vectors of length $n$, with each $y_i$ indicating error positions. The training set is constructed by enumerating all correctable errors and pairing them with their respective syndromes.

- Input data ($X$): syndromes generated from BCH$(n, k, t)$ codes, where each instance $x_i$ is of length $n - k$.
- Output data ($Y$): binary error vectors of length $n$, where each instance $y_i$ indicates a correctable error pattern.

The mapping can thus be formalized as: $(X = x_i) \rightarrow (Y = y_i)$

Once the training dataset is assembled, the MLP model is trained to approximate the function that maps syndromes to error vectors. The model comprises multiple layers, where each layer contains a set of neurons connected to the previous and next layers. During training, the model iteratively updates its weights to minimize the prediction error using backpropagation and gradient descent techniques.

To evaluate the effectiveness of the MLP model, we conducted a series of experiments analyzing the impact of different activation functions and architectural depths. Specifically, we compared models utilizing the sigmoid function and the ReLU, both with single and double hidden layers. These variations allow us to assess how activation choices and network depth influence decoding performance. The comparative results of these experiments are summarized in a table, offering insight into optimal configurations for specific BCH code parameters.

### 3.3. The improved multilayer perceptron decoder using automorphsim set

In our study, we aim to enhance the MLP technique by incorporating mathematical automorphism groups, which can potentially improve the decoding performance. Automorphism groups have been previously used in the context of ECCs, offering significant benefits in terms of code symmetry and reducing decoding complexity. By harnessing the capabilities of MLPs and the intrinsic properties of automorphism groups, we aim to develop a decoding method that is both more efficient and robust. Previous research has explored the application of automorphism groups in decoding, specifically studying the role of permutation automorphisms in the decoding of linear codes [26]. Building upon these foundational studies and integrating automorphism groups with MLPs, our proposed approach, as illustrated in Figure 5, presents the potential to push the boundaries of existing decoding techniques and contribute to the advancement of machine learning-based decoding methods.

We propose to enhance the decoding process by incorporating the sets of automorphism groups for BCH codes. We leverage permutations generated by the automorphism group to improve decoding accuracy by applying them iteratively until the correct decoding is achieved. The initial step involves generating automorphism group permutations, which are the set of transformations that preserve the properties of the BCH codes. The Algorithm 1 outlines the decoding procedure for the multi-layer perceptron with automorphism (MLPAut) decoding and the requirements for its implementation.

**Algorithm 1** The algorithm of decoding by the MLPAut decoder

**Require:** $\mathcal{A}ut = \{\sigma_i : \sigma_i(C) = C\}, d(Hamming \quad distance), t = \lfloor \frac{d-1}{2} \rfloor$
**Require:** MLP-nR the multilayer perceptron model
$M \leftarrow ReceivedMessage$
$M_c \leftarrow M$
$\sigma \leftarrow \sigma_{Id}$
**while** (True) **do**
  $MsgAuto \leftarrow \sigma(M)$
  $MsgMod \leftarrow MLP - nR(MsgAuto)$
  $dist \leftarrow d(MsgMod, M_c)$
  **if** $(Syndrome(MsgMod) \quad ! = 0 \quad or \quad dist > t)$ **then**
    $\sigma \leftarrow \sigma_i$ ▷ chose permutation
    $M \leftarrow MsgMod$
  **else**
    $MesDec \leftarrow \sigma^{-1}(MsgMod)$
    $break$
  **end if**
**end while**

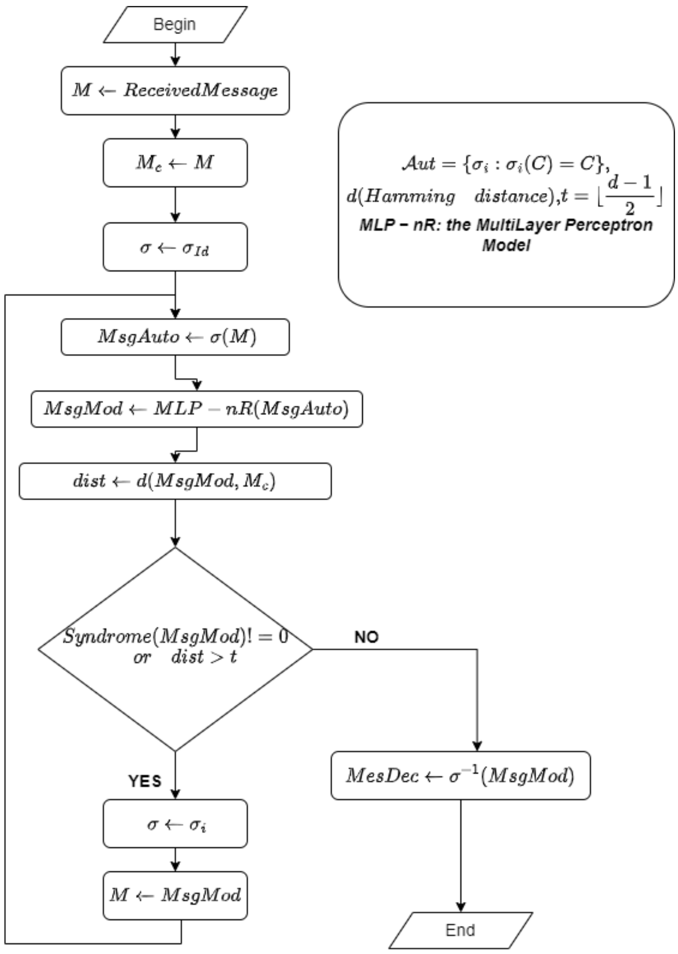


Figure 5. MLPAut diagram

## 4. RESULTS AND DISCUSSION

In this section, we present an overview of the experimental results conducted over an additive white Gaussian noise (AWGN) channel. The focus is on evaluating the performance of two decoding models: the standard MLPDec and its enhanced version. The improved model incorporates automorphism group techniques and is referred to as multi-layer perceptron automorphism decoder (MLPAutDec). We compare both models under various BCH code configurations to assess their decoding accuracy. The results demonstrate the impact of automorphisms on enhancing error correction capabilities.

### 4.1. Multilayer perceptron decoder results

The first work to create our decoders for different BCH codes is to create a model decoder using MLP with different parameters. Table 1 provides a comparison of different models trained on BCH (31,21,5) with various configurations, including the number of layers, neurons per layer, activation functions, iteration numbers, and corresponding training scores. The same for the code BCH (31,26,5), the results of training score are given in Table 2.

Table 1. Scores for BCH (31-21-5)

| Models training BCH (31,21,5) | Layer neurones | Activation function | Iteration number's | Score (%) |
|---|---|---|---|---|
| MLP-2nR | 2*n | Relu | 10 000 | 57 |
| MLP-2nL | 2*n | Logistic | 10 000 | 51 |
| MLP-nR | n | Relu | 10 000 | 51 |
| MLP-nL | n | Logistic | 100 000 | 49 |
| MLP-nR | n | Relu | 100 000 | 54 |

Table 2. Scores for BCH (31-26-5)

| Models training BCH (31,26,5) | Layer neurones | Activation function | Iteration number's | Score (%) |
|---|---|---|---|---|
| MLP-nR | n | Relu | 10 000 | 100 |
| MLP-n4R | [n/4] | Relu | 100 000 | 100 |
| MLP-n6R | [n/6] | Relu | 100 000 | 93 |
| MLP-n8R | [n/8] | Relu | 100 000 | 58 |

When considering execution complexity, it is advisable to select the model with the fewest neurons and the simplest activation function, in this case, the MLP-nL (MLP with 'n' neurons and logistic activation function). This model has the lowest execution complexity compared to others, as it has the fewest neurons, and the logistic activation function is generally computationally less expensive than ReLU. However, it is important to note that the choice of model should not be based solely on execution complexity. The accuracy and performance of the model are equally important factors. Therefore, it is essential to consider the trade-off between execution complexity and model performance. In this case, the idea is to enhance the performance of the models by using automorphism groups.

### 4.2. Performing multilayer perceptron decoder with automorphism set MLPAut

As outlined in the previous section, this part presents several experiments that combine the standard MLPDec with a subgroup of automorphisms denoted as $Sg$, where the length of the group is represented by Len$(Sg) = p$. The main objective is to determine the minimum number of automorphisms $p$ required to achieve optimal decoding performance for each MLP model. Experimental results indicate that performance improves as $p$ increases but eventually reaches a saturation point beyond which further gains are minimal.

Figure 6 illustrates the bit error rate (BER) performance for the BCH (15,7,5) code under various values of $p$. The analysis shows that when $p \geq 3$, the model achieves stable performance in terms of BER, with a notable improvement in SNR gain. This stabilization suggests that a small number of well-selected automorphisms can significantly enhance the decoder's accuracy without excessive computational cost.

Extending the analysis to other codes, such as BCH (31,26,3) and BCH (31,21,5), further supports the positive impact of integrating automorphism groups with MLPDec. Figures 7 and 8 display the BER performance for these codes, demonstrating that the improved decoder (MLPAut) consistently outperforms or matches the original MLPDec. Specifically, the MLPAut model achieves convergence at $p = 5$ for BCH (31,21,5) and at $p = 6$ for BCH (31,26,3), indicating that a small automorphism set is sufficient for performance enhancement.

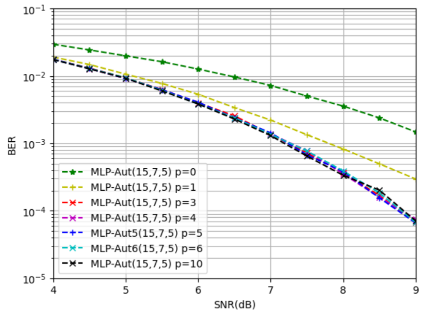


Figure 6. BER performance for BCH (15,7,5) code

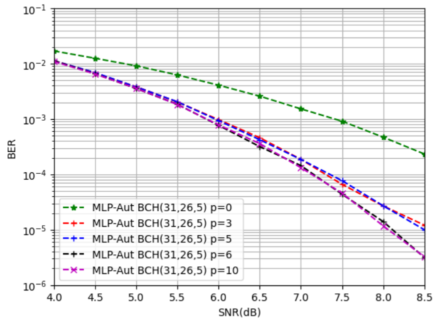


Figure 7. BER performance MLPDec and MLPAut for BCH (31,26,3) code

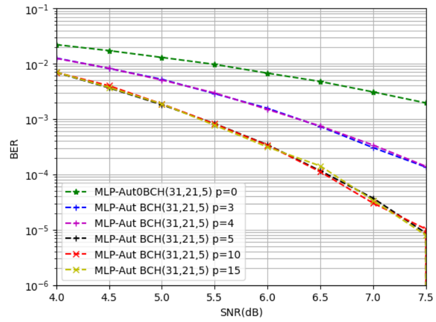


Figure 8. BER performance MLPDec and MLPAut for BCH (31,21,5) code

The convergence values for each tested BCH code are summarized in Table 3. This table confirms that the integration of automorphism groups enhances model convergence and decoding performance with relatively few permutations. Identifying these thresholds provides a practical guideline for balancing performance gains with computational efficiency.

Incorporating the findings from this section, it is evident that automorphisms play a significant role in enhancing the performance of machine learning-based decoders. These results suggest that automorphism sets, when carefully selected, offer a scalable and effective strategy for improving error correction. As decoding systems evolve, such hybrid techniques may pave the way for more robust and efficient implementations in communication systems.

Table 3. Minimum automorphism count $p$ for model convergence

| BCH code | Convergence at $p = 3$ |
|---|---|
| BCH (15,7,5) | $p = 3$ |
| BCH (31,26,3) | $p = 6$ |
| BCH (31,21,5) | $p = 5$ |

### 4.3. Comparison with competitors

Figures 9 and 10 provide a comparative analysis of three distinct models: HSDec, HWDec, and MLPDec, all applied to a (31,21,5) and (31,26,3) codes. Notably, HSDec and HWDec show closely aligned performances across various SNR levels, suggesting a shared efficacy in error correction. In contrast, MLPDec (p =5) in Figure 9 and MLPDec (p =6) in Figure 10 demonstrates a substantial performance advantage over both HSDec and HWDec, especially as SNR increases.

This disparity in performance indicates that MLPDec, when enriched with five automorphisms (p =5)- and (p =6) (Figures 9 and 10), excels in error correction and noise mitigation for this specific code. The evident advantage of MLPDec in higher SNR scenarios highlights its potential to significantly improve the BER and overall reliability of data transmission. Also, the results have shown that the presence of automorphisms in the MLPDec model, seems to confer a substantial advantage in handling noise and improving error correction.

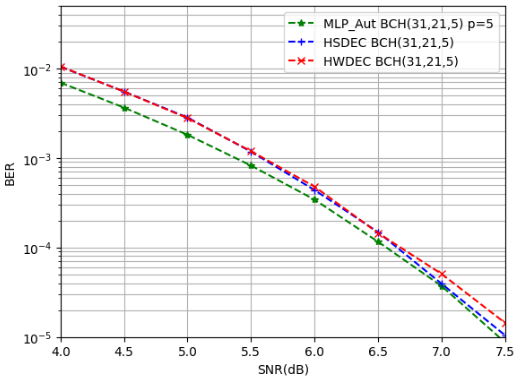


Figure 9. BER comparison of HSDec, HWDec, and MLPDec for BCH (31,21,5) code

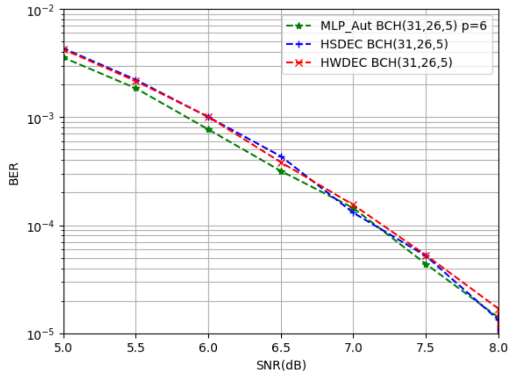


Figure 10. BER comparison of HSDec, HWDec, and MLPDec for BCH (31,26,3) code

## 5. CONCLUSION

In this research paper, we introduced a novel approach for decoding linear ECCs by utilizing MLPs and automorphism groups. Our results indicate that our proposed decoder significantly outperforms all existing machine learning decoders with a learning score of less than 100 percent. This underscores the effectiveness of integrating automorphism groups and deep learning to enhance the performance of ECCs, particularly BCH codes. This research not only opens up new avenues for exploration but also holds practical implications for designing more efficient error correction systems. Ultimately, our work contributes to the progression of the field of ECCs and offers a promising strategy for optimizing the performance of linear codes. Optimizing the decoder in terms of performance and complexity remains a pivotal aspect of ECC research. Although our current study presents a promising approach for optimizing the performance of linear codes using MLPs and automorphism groups, there is still scope for improvement in both performance and complexity. As such, we see a clear path for future research aimed at further refining our decoder. Specifically, we plan to explore advanced deep learning techniques and assess the potential of amalgamating our approach with other error correction methods. Our objective is to continually push the boundaries of error correction performance and develop new, efficient solutions for real-world applications.

## FUNDING INFORMATION

Authors state no funding involved.

## AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

| Name of Author | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Imrane Chemseddine Idrissi | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| Said Nouh | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| El Mehdi Bellfkih | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| Mohammed El Assad | | ✓ | | | | ✓ | | | | ✓ | ✓ | | | |
| Abdelaziz Marzak | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | | | |

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

## CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

## DATA AVAILABILITY

Authors state that no external data were used in this study. All data were generated by the algorithm developed by the authors.

## REFERENCES


[1] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948, doi: 10.1002/j.1538-7305.1948.tb01338.x.

[2] T. S. Rappaport, *Wireless communications: principles and practice*. 2nd edition, Cambridge, United Kingdom: Cambridge University Press, 2024.

[3] T. M. Cover and J. A. Thomas, *Elements of information theory*. Hoboken, United States: John Wiley Sons, Inc., 2001, doi: 10.1002/0471200611.

[4] J. G. Proakis and M. Salehi, *Digital communications*. 5th edition, New York, United States: McGraw-Hill, 2008.

[5] E. Dubrova, M. Näslund, G. Selander, and K. Norrman, "Error-correcting message authentication for 5G," in *the 2nd International Workshop on 5G Security*, Dec. 2016. Accessed: Jan. 14, 2026. [Online]. Available: https://eudl.eu/doi/10.4108/eai.18-6-2016.2264115

[6] A. Goldsmith, *Wireless communications*. Cambridge, United Kingdom: University Press, 2005, doi: 10.1017/CBO9780511841224.

[7] D. J. MacKay, *Information theory, inference and learning algorithms*. Cambridge, United Kingdom: University Press, 2003.

[8] G. C. Clark and J. B. Cain, *Error-correction coding for digital communications*. Boston, United States: Springer New York, 1981, doi: 10.1007/978-1-4899-2174-1.

[9] I. B. Djordjevic, "QKD-enhanced cybersecurity protocols," *IEEE Photonics Journal*, vol. 13, no. 2, pp. 1–8, Apr. 2021, doi: 10.1109/JPHOT.2021.3069510.

[10] Y. Li and M. Salehi, "An efficient decoding algorithm for concatenated RS-convolutional codes," in *2009 43rd Annual Conference on Information Sciences and Systems*, Mar. 2009, pp. 411–413, doi: 10.1109/CISS.2009.5054755.

[11] S. Puchinger, J. Rosenkilde, and G. Solomatov, "Improved power decoding of algebraic geometry codes," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 509-514, doi: 10.1109/ISIT45174.2021.9517938.

[12] Y. Zhu and S. Tang, "New algebraic soft decision decoding algorithm for reed-solomon code," *arXiv:1407.8069*, Jan. 2015.

[13] S. Nouh, I. Chana, and M. Belkasmi, "Decoding of block codes by using genetic algorithms and permutations set," *International Journal of Communication Networks and Information Security*, vol. 5, no. 3, Apr. 2022, doi: 10.17762/ijcnis.v5i3.428.

[14] M. S. E. K. Alaoui, S. Nouh, and A. Marzak, "A low complexity soft decision decoder for linear block codes," *Procedia Computer Science*, vol. 127, pp. 284–292, 2018, doi: 10.1016/j.procs.2018.01.124.

[15] M. Seddiq, S. Nouh, and A. Marzak, "A fast method to estimate partial weights enumerators by hash techniques and automorphism group," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 9, 2017, doi: 10.14569/IJACSA.2017.080937.

[16] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep learning methods for improved decoding of linear codes," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 119–131, Feb. 2018, doi: 10.1109/JSTSP.2017.2788405.

[17] H. Kim, Y. Jiang, R. Rana, S. Kannan, S. Oh, and P. Viswanath, "Communication algorithms via deep learning," *International Zurich Seminar on Information and Communication (IZS)*, Feb. 2018, pp. 47-50, doi: 10.3929/ethz-b-000245054.

[18] T. Gruber, S. Cammerer, J. Hoydis, and S. T. Brink, "On deep learning-based channel decoding," in *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, USA: IEEE, Mar. 2017, pp. 1–6, doi: 10.1109/CISS.2017.7926071.

[19] C. T. Leung, R. V. Bhat, and M. Motani, "Multi-label neural decoders for block codes," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, IEEE, 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9148786.

[20] A. Gunturu, A. Agrawal, A. K. R. Chavva, and S. Pedamalli, "Machine learning based early termination for turbo and LDPC decoders," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, Nanjing, China, 2021, pp. 1-7, doi: 10.1109/WCNC49053.2021.9417420.

[21] S. Cammerer, T. Gruber, J. Hoydis, and S. Ten Brink, "Scaling deep learning-based decoding of polar codes via partitioning," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Singapore: IEEE, Dec. 2017, pp. 1–6, doi: 10.1109/GLOCOM.2017.8254811.

[22] C. I. Imrane, N. Said, B. E. Mehdi, E. K. A. Seddiq, and M. Abdelaziz, "Machine learning for decoding linear block codes: case of multi-class logistic regression model," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 1, pp. 538-547, Oct. 2021, doi: 10.11591/ijeecs.v24.i1.pp538-547.

[23] R. Chao and B. W. Reichardt, "Quantum error correction with only two extra qubits," *Physical Review Letters*, vol. 121, no. 5, 2018, doi: 10.1103/PhysRevLett.121.050502.

[24] D. W. Kribs, R. Laflamme, D. Poulin, and M. Lesosky, "Operator quantum error correction," *Quantum Information and Computation*, vol. 6, no. 4-5, pp. 382-399, 2006, doi: 10.26421/QIC6.4-5-6.

[25] B. M. Terhal, "Quantum error correction for quantum memories," *Reviews of Modern Physics*, vol. 87, no. 2, pp. 307-346, doi: 10.1103/RevModPhys.87.307.

[26] X. Chen and M. Ye, "Neural decoders with permutation invariant structure," *Journal of the Franklin Institute*, vol. 360, no. 8, pp. 5481–5503, May 2023, doi: 10.1016/j.jfranklin.2023.03.024.

**BIOGRAPHIES OF AUTHORS**


**Imrane Chemseddine Idrissi** has earned a Ph.D. in Computer Science from the Faculty of Sciences Ben M'Sik, Hassan II University, Casablanca, Morocco. He previously completed a master's degree in Data Science and Big Data at ENSIAS, Mohammed V University, in 2019. His research interests encompass computer networks, telecommunications, information systems, coding theory, machine learning, and deep learning. He can be contacted at email: imran.chems@gmail.com.

**Said Nouh** received his Ph.D. in Computer Science from National School of Computer Science and Systems Analysis (ENSIAS) in Rabat, Morocco, in 2014. He currently holds the position of professor and habilitation to direct research (HDR) at Hassan II University, Faculty of Sciences Ben M'Sik in Casablanca. His research focuses on artificial intelligence, machine learning, deep learning, telecommunications, information theory, and coding theory. He can be contacted at email: said.nouh@univh2m.ma.

**El Mehdi Bellfkih** earned his Ph.D. in Applied Mathematics from Hassan II University, where his research primarily focused on coding theory and machine learning, with a particular emphasis on ECCs. He can be contacted at email: elmehdi.bellfkih@gmail.com.

**Mohammed El Assad** is a Ph.D. candidate in Computer Science at the Faculty of Sciences Ben M'Sik, Hassan II University in Casablanca. He completed his master's degree in Computer Science at Faculty of Sciences Ben M'Sik in 2021. His research interests include computer networks, telecommunications, information systems, coding theory, as well as machine learning and deep learning. He can be contacted at email: mohammed.elassad-etu@etu.univh2c.ma.

**Abdelaziz Marzak** holds a Ph.D. in Computer Science and currently serves as a professor of higher education at the Faculty of Sciences Ben M'Sik, Hassan II University, Casablanca. His research spans several domains, including artificial intelligence, machine learning, fuzzy logic, and the internet of things (IoT). He can be contacted at email: abdelaziz.marzak@univh2m.ma.