STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **13**, January 2025, pp 222–247.
Published online in International Academic Press (www.IAPress.org)

Advanced Big Data Analytics: Integrating Fuzzy C-Means, Encoder-Decoder CNNs, and Genetic Algorithms for Efficient Clustering and Classification

Fatima Belhabib [1,*], Mohamed Benslimane [2], Karim El Moutaouakil [2]

[1] *Engineering Science Laboratory, Mohamed Ben Abdellah University,ISPITS , FES, Morocco*
[2] *Engineering, Systems and Applications, Sidi Mohamed Ben Abdellah University, FES, Morocco*

Abstract In the realm of Big Data analysis, the pivotal question of data clustering takes center stage. This study delves into optimizing this analysis by adopting a hybrid approach that integrates the Fuzzy C-Means (FCM) methodology, Encoder-Decoder Convolutional Neural Networks (CNN), Genetic Algorithms (GAs), and an optimal classification strategy for data clustering and categorization. FCM provides a flexible clustering foundation with its fuzzy logic, while the Encoder-Decoder CNN contributes to extracting complex features and refining the model. Genetic Algorithms finely adjust the parameters of the hybrid model. The optimal classification strategy complements this approach by ensuring precise data categorization. This hybrid strategy leverages the specific strengths of each component, thereby overcoming inherent limitations in each technique. FCM ensures robust cluster formation the Encoder-Decoder CNN improves feature representation, Genetic Algorithms optimize the hyper-parameters of the hybrid model, and optimal classification reinforces the accuracy of data categorization. Experiments conducted on various Big Data sets reveal a significant enhancement in clustering and classification accuracy, as well as overall analysis efficiency. This research represents a substantial contribution to the evolution of Big Data analysis by proposing an integrated solution harnessing the power of FCM, Encoder-Decoder CNN, Genetic Algorithms, and optimal classification The results suggest that this hybrid approach not only increases clustering and classification accuracy but also provides a versatile and adaptable solution to address challenges in large-scale data analysis.

Keywords Fuzzy C-Means (FCM) clustering, optimized Encoder-Decoder, classification, Genetic Algorithms in Clustering, Optimisation

DOI: 10.19139/soic-2310-5070-1978

1. Introduction

The exponential growth in data volume within the Big Data ecosystem has presented significant challenges, fundamentally reshaping our approach to extracting information from these vast data reservoirs. This rapid growth underscores the need to rethink our analysis methods to extract relevant information from this ocean of data. At the core of these challenges, data clustering and classification emerge as a central concern, emphasizing the need to develop innovative methodologies to address these challenges. This urgency is highlighted by various scholars who have emphasized the transformed impact of Big Data on data analysis paradigms [3]. Data classification, defined as the process of organizing and grouping similar elements based on certain common features or attributes, is crucial in simplifying the complexity of data, making it easier to interpret and use in the future [3, 58]. This perspective aligns with the works of [31], who emphasize the importance of effective data classification for information organization and retrieval [3, 31]. Similarly, data clustering, or clustering, stands out as a key methodology in the landscape of massive data analysis. This fundamental approach seeks to organize the elements of a dataset into groups or clusters based on their similarity or intrinsic relationships. Its primary goal is to identify

*Correspondence to: Fatima Belhabib (Email: f.belhabib@yahoo.fr). Engineering Science Laboratory, Mohamed Ben Abdellah University,ISPITS , FES, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

subsets of elements sharing common characteristics, paving the way for a profound understanding of the data and valuable insights. This aligns with the observations of [6]., who emphasize the importance of clustering in revealing hidden patterns within datasets, especially in fields like market analysis and medicine [6, 32]. However, despite the undeniable importance of data clustering and classification, traditional techniques face significant challenges when applied to the context of Big Data. The complexity arising from the diversity of data, its massive volume, and the variety of formats pose substantial obstacles. Classical algorithms may prove inefficient in processing these vast datasets, resulting in reduced performance in terms of precision, execution speed, and the ability to extract relevant information. It is precisely in this dynamic and demanding context that this study is situated. It focuses on the urgent need to optimize Big Data analysis by adopting an innovative approach to data clustering. The primary goal is to transcend conventional methods by synergistic-ally integrating different approaches to overcome specific challenges related to the variety, volume, and velocity of Big Data. Our hybridization relies on a systematic and informed approach, divided into various phases to develop, implement, and evaluate our hybrid Big Data analysis model. We initiated our process with an in-depth review of the literature in the field of Big Data analysis and clustering methods. This step was crucial to understand current advancements, identify gaps in existing approaches, and pinpoint opportunities for improvements. This is consistent with the methodology employed by [7]., who advocate for a thorough literature review to inform the development of innovative solutions in Big Data analysis [7]. Consequently, we conducted a thorough analysis of traditional clustering methods such as K-means and hierarchical clustering. This allowed us to identify the strengths and weaknesses of these approaches in the context of Big Data, laying the foundation for our innovative approach. Our examination of traditional methods aligns with the works of [1]., who stress the importance of assessing the performance of conventional clustering algorithms in the context of Big Data [1, 32].

Subsequently, our methodology shifted towards the exploration of current hybrid approaches. We closely examined how different techniques, including fuzzy logic, convolutional neural networks (CNN) of the Encoder-Decoder type, and genetic algorithms, are combined to overcome individual limitations. This exploration was essential to identify best practices and potential synergies. This aligns with the findings of [8]., who highlight the effectiveness of hybrid approaches in addressing the challenges of Big Data analysis [9]. Our innovative approach hinges on the skillful integration of three specific techniques: Fuzzy C-Means (FCM) for its flexibility, Convolutional Neural Networks (CNN) of the Encoder-Decoder type for their ability to capture complex features, and Genetic Algorithms (GA) [17] for the meticulous adjustment of parameters. Once these components were carefully selected, we embarked on the meticulous design of our hybrid model, detailing how each element interacts and the key parameters governing our model. At the core of this approach lies the recognition of the crucial importance of extracting relevant information from these vast datasets. This information not only fuels crucial insights for businesses and organizations but also paves the way for a better understanding of the inherent trends, patterns, and complex relationships within Big Data. The proposed approach relies on a strategic and harmonious fusion of the strengths of three distinct techniques: Fuzzy C-Means (FCM), Convolutional Neural Networks (CNN) with an Encoder-Decoder architecture, and Genetic Algorithms (GA)[17]. This deliberate combination aims to overcome the individual limitations of each method, capitalizing on the specific advantages of each, resulting in an overall improvement in the efficiency of Big Data analysis.

Firstly, Fuzzy C-Means (FCM) is selected for its ability to integrate fuzzy logic into the clustering process. This flexible clustering approach is well-suited for addressing the inherent uncertainty in massive data. FCM contributes to the creation of more robust clusters by considering the possibility that an element may belong to multiple groups with different membership degrees, particularly relevant in complex environments. This choice is influenced by the works of Bezdek et al, who pioneered the application of fuzzy logic in clustering for handling uncertainty in data [5]. In parallel, Convolutional Neural Networks (CNN) with an Encoder-Decoder architecture are incorporated due to their ability to capture and represent complex features in the data. Originally developed for computer vision, these networks prove effective in detecting subtle patterns and relationships, essential for in-depth Big Data analysis. This integration aligns with the findings of [9], who highlight the effectiveness of CNNs in feature extraction and pattern recognition [9].

Finally, Genetic Algorithms (GA)[4] are integrated to achieve fine optimization of the parameters of the hybrid model. Inspired by biological processes such as natural selection and genetic recombination, GA dynamically adjusts the model's parameters to achieve optimal performance. This integration is informed by the works of [10],

who introduced the concept of genetic algorithms for optimization problems [10].
This synergistic fusion of FCM, CNN Encoder-Decoder, and GA offers significant advantages. It promises an improved representation of data features, enabling a deeper understanding of the underlying structures. Moreover, it facilitates the formation of more robust clusters by leveraging the flexibility of FCM and the representation capacity of CNN. Lastly, fine-tuning of parameters by GA contributes to refining the overall performance of the hybrid model, paving the way for a more precise and efficient Big Data analysis. This innovative approach aspires to overcome specific challenges related to the complexity and variety of Big Data, providing an integrated solution for a more in-depth and precise exploration of extensive datasets.
The rest of the essay is structured as follows: a discussion of related works in the next section, a presentation of our methodology in section 3, our model in Section 3.2, an illustration of the results proposed, and a summary in Section 4, and finally a conclusion in Section 5.

2. Related works

In this section, we explore various existing strategies for data clustering and classification, adapting them to our innovative method. Firstly, a clustering method based on a dissimilarity matrix has been proposed [48]. This approach transforms the dissimilarity matrix to highlight distinct groups by representing them as dark blocks along the diagonal. Although initially designed to detect halo-like structures in dark matter data, it proves to be adaptable to our context. Another perspective [29] focuses on anomaly detection within the clustering context. This method uses dendrograms to provide a visual representation of clusters, an approach that we can integrate to enhance the visualization of our results in various application contexts. Data classification is a dynamic research field exploring various approaches to assign labels to datasets. Supervised learning remains a dominant approach, where models are trained on labeled data [23]. In parallel, unsupervised learning, particularly clustering, is extensively studied to uncover intrinsic structures within data [21]. The growing prominence of neural networks, especially in deep learning, has significantly enhanced model performance. Convolutional Neural Networks (CNNs) have become indispensable for image classification [9], while Recurrent Neural Networks (RNNs) are employed for processing temporal sequences [23]. Ensemble methods, such as Random Forests [24] and boosting [25] are commonly used to enhance model robustness. Semi-supervised learning, applied in contexts where only partially labeled data is available [26] and active learning, where the model actively selects examples to label, represent promising research directions. In the field of natural language processing, text classification remains a major subject. Models based on techniques like word embeddings [28] and transformers [**?**] have made significant advances. Managing imbalanced data, interpreting complex models, and holistic evaluations considering various metrics are crucial aspects [30]. Specific applications, whether in the medical, financial [32], or object recognition [31] domains, pose particular challenges that are extensively researched. An alternative approach introduces a clustering method based on a single linkage specifically designed for segmenting time series data, especially in the field of patient medical monitoring [15]. Although initially conceived for time series, we can explore its potential for adaptation to our method to handle complex datasets.

Deep learning has become a leading research area in data clustering over the years. Supervised learning methods have shown promising results for clustering large datasets [18, 20]. However, our innovative approach, based on the fusion of specific techniques such as Fuzzy C-Means (FCM), Convolutional Neural Networks (CNN), and Genetic Algorithms (GA), provides an alternative that can overcome the limitations of supervised learning methods, especially when dealing with raw data. Approaches based on roughness or non-determinism have been developed to address uncertainty in the clustering process. A hierarchical approach combining a neural network with fuzzy logic has been proposed [18]. In the literature [20], a Convolutional Neural Network (CNN) model based on non-determinism has been developed for classification and clustering. Although promising, this approach relies primarily on supervised learning. Our hybrid methodology aims to incorporate these non-deterministic concepts while offering a more flexible and adaptive approach. Semi-supervised clustering has also been explored as a solution to simultaneously manage clustering and classification [16], [9], [23]. A pseudo-labeling technique has been used to create limited labeled data. This approach can inspire our methodology to leverage learning from partially labeled data. Beyond these works, other studies have focused on the advantages of the Fuzzy C-Means

(FCM) algorithm compared to other clustering techniques [27, 28]. However, despite these existing research efforts, many clustering mechanisms still face challenges related to computation time, absolute clustering, and performance metrics. Our innovative approach, based on the fusion of specific techniques and the design of a hybrid model, aims to overcome these challenges by offering a more precise, flexible, and performant method for Big Data analysis.

3. Material and methods

This section details the sequential process employed to attain the presented results. To commence, we present the methodology, delve into The Proposed Model, and conduct a thorough exploration of the Proposed Model.

3.1. methodology

In this section, we elaborate on our innovative approach aimed at optimizing Big Data analysis through a hybrid methodology that integrates Fuzzy C-Means (FCM), a Convolutional Neural Network (CNN) encoder-decoder, and genetic algorithms into the clustering and classification process. This section is subdivided into three phases: In the first phase, we introduce our approach using the Fuzzy C-Means (FCM) algorithm as the foundation for both clustering and classification. FCM is deployed to create meaningful clusters within the dataset based on the structural similarities between data points. Utilizing the concept of ”fuzziness,” each data point can belong to multiple clusters with varying degrees of membership, allowing for a flexible representation of relationships between the data. This iterative process gradually converges towards a configuration where stabilized cluster centers and membership degrees optimally reflect the intrinsic structure of the data. The FCM approach, with its ability to model nuanced relationships between data points, forms a robust foundation for simultaneous classification and clustering. By combining the flexibility of FCM with other innovative techniques, our goal is to create an integrated approach globally optimized for the complex challenges posed by Big Data analysis. In the second phase of our CNN Encoder-Decoder model, the encoder plays a crucial role. It consists of several convolutional layers. Convolutional layers are essential as they allow for the analysis of incoming data to extract relevant features or patterns. Convolutional filters detect various patterns based on their configuration, ranging from simple edges to more complex structures as we progress through the network. In parallel, pooling layers are used to reduce the dimensionality of the processed data. This reduction is strategic: it decreases the volume of data, simplifying the network’s processing while preserving the most salient information. Pooling also facilitates a certain invariance to the position and scale of the features detected in the image. At the output of the encoder, we obtain what is called the latent space. This is a compressed representation of the initial data, capturing the essential elements necessary for reconstruction or other forms of processing such as classification. In this context, the latent space can already be used to perform a preliminary classification based on the extracted features. The role of the decoder is to take this condensed representation and reconstruct the original image or input data. For this purpose, it uses deconvolution and upsampling layers, which gradually restore the original dimensions of the data. Deconvolution layers work to reverse the convolution process, while upsampling increases the resolution of the image step by step. The final output is a representation that attempts a faithful reconstruction of the original data, using the condensed information contained in the latent space. This process demonstrates the model’s effectiveness in capturing and reconstructing data from essential features, despite the dimensionality reduction carried out by the encoder. The third phase integrates genetic algorithms In the third phase, the parameters of the classification and clustering models can be adjusted through the use of genetic algorithms. These algorithms begin by creating a range of basic model configurations, drawing inspiration from the ideas of natural evolution. Each configuration, or ”individual,” is assessed according to how well it can carry out the tasks of clustering and classification. The most advantageous configurations are then chosen for reproduction, in which they trade some of their parameters to produce new configurations that incorporate the best features of the original ones. This process of selection, crossover, and mutation repeats over several generations, gradually refining the model parameters. The iterative adjustment through these genetic algorithms not only improves the accuracy of classifications and clusterings but also dynamically adapts the models to changes in data or new analytical requirements. This strategic approach ensures continual improvement in precision and efficiency, which is crucial in handling large volumes of data in Big Data contexts.

3.2. The Proposed Model

At the core of our research endeavor, focused on optimizing Big Data analysis through a hybrid approach integrating Fuzzy C-Means, Encoder-Decoder CNN, and genetic algorithms in clustering, we embark on our exploration by conceptualizing the principle of normality tailored to our specific context. The primary objective is to define a tailored solution to enhance Big Data analysis by effectively detecting abnormal patterns within the network. Normality takes shape through the creation of a formal model that elucidates the relationships between key variables associated with the dynamics of the system. Our methodological approach centers on improving efficiency from the clustering phase onward, strategically reducing the number of features. To achieve this, we employ feature selection algorithms, particularly favoring subset consistency and genetic search approaches. This selection phase explicitly aims to eliminate irrelevant features before undertaking clustering and categorization operations, followed by the Fuzzy C-Means clustering formulation process. The crucial importance of this step lies in reducing processing time, dataset training requirements, and the overall complexity of the model. As for the classification process, it relies on our novel hybridization method, thereby reinforcing the quality of detection within our hybrid approach. Figure 1 visually depicts the model schema, illustrating the flow of our methodology applied to optimizing Big Data analysis.

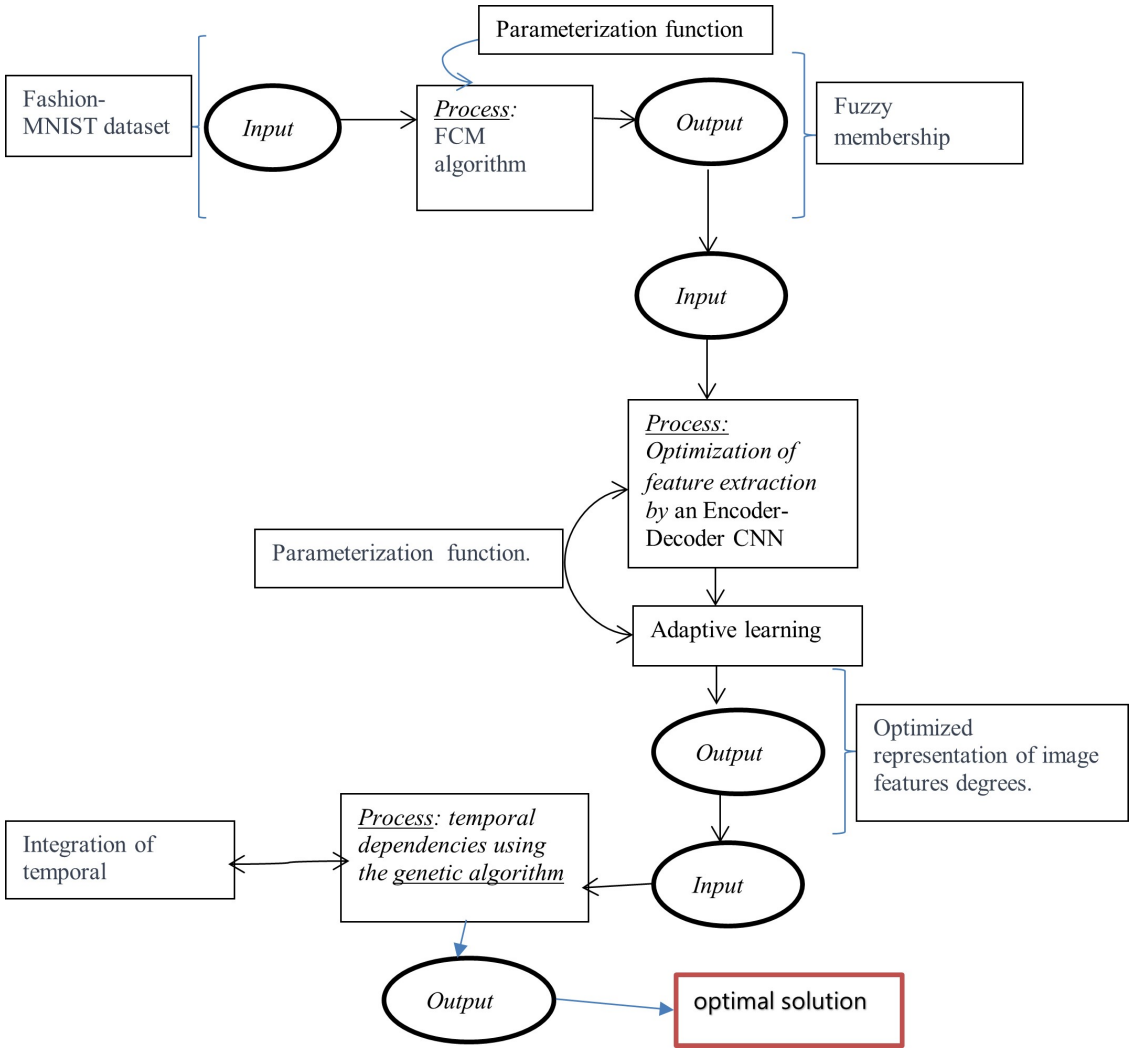


Figure 1. The Proposed Model

3.3. Comprehensive Exploration of the Proposed Model

Our new method consists of two phases. Data preprocessing is an important phase in both data analysis and the machine learning process. Data preparation is a critical step, involving a variety of methods and tasks applied to raw data to make them suitable for subsequent analysis, modeling, and interpretation. Data normalization is a process of scaling data to a common range, making it more suitable for machine learning algorithms. In the context of the Fashion-MNIST dataset. Given a dataset $X = \{x_1, x_2, x_3, \ldots, x_n\}$consisting of n data points, where each xi represents an image in the dataset, and the pixel values of the images typically range from 0 to 255. The normalize the pixel values to a common scale, you can perform the following mathematical transformation:

$$x_i' = \frac{x_i}{\text{max\_pixel\_value}} \tag{1}$$

where:

- x_i' represents the normalized pixel value of the i-th image
- x_i is the original pixel value of the i-th image,
- max_pixel_value is the maximum pixel value in the dataset, which is 255 for the Fashion-MNIST dataset.

This normalization process scales the pixel values to the range [0, 1]. After normalization, the pixel values of the images are within this common range, which is often preferred for machine learning models. This step can enhance the model's convergence and performance, especially when using deep learning [3] models like Optimized Encoder-Decoder CNN.

3.3.1. Fuzzy C-Means (FCM) We have implemented a novel strategy by combining Fuzzy C-Means (FCM), a fuzzy clustering algorithm, with the Euclidean distance to form a hybridization method. FCM is widely used for grouping data based on their similarity, and its integration with the Euclidean distance is a popular approach for assessing the similarity between data points. This hybrid method employs standard Euclidean geometry to calculate the distance between two points. The Fuzzy C-Means (FCM) algorithm determines membership values for each cluster based on the distances between each data point and the cluster centroids. One of the notable advantages of FCM is its robustness in producing reliable results, especially when dealing with overlapping data. Moreover, FCM has the flexibility to assign a data point to multiple clusters if necessary. Despite these strengths, there are certain limitations to be mindful of, such as computational time, precision, and the substantial number of iterations required for convergence. It is worth noting that FCM often relies on the Euclidean distance metric, which has implications for the weight assigned to different data points. This choice can influence the algorithm's behavior and outcomes. Therefore, careful consideration of these factors is essential when applying FCM with the Euclidean distance for clustering analysis. Consider the dataset

$$X = \{x_1, x_2, x_3, \ldots, x_q\}$$

with the cluster set

$$Y = \{y_1, y_2, y_3, \ldots, y_p\}$$

and the membership

$$W = \{w_{kl} \mid 1 \leq k \leq q, 1 \leq l \leq p\}$$

. This means that the membership set W is composed of all elements w_{kl} where k ranges from 1 to q and l ranges from 1 to p. These$w_{kl}w_{kl}$ elements represent the membership degrees of each data point k to each cluster l; FCM can be formulated.

$$\gamma = \sum_{k=1}^{e} \sum_{l=1}^{p} w_{kl}^{o} \|x_l - y_k\|^2 \tag{2}$$

$$\sum_{l=1}^{e} w_{KL} = 1, \quad w_{KL} \geq 0 \tag{3}$$

Therefore, optimizing the equation helps in updating the membership matrix as well as cluster centers, as shown below:

$$y_k = \frac{\sum_{l=1}^{p} w_{kl}^{o} x_l}{\sum_{l=1}^{p} w_{kl}} \tag{4}$$

Membership matrix:

$$w_{kl} = \left(1 + \left(\frac{e_{kl}}{\gamma_k}\right)^{-\frac{1}{o-1}}\right)^{-1} \tag{5}$$

Table 1. Description of the Algorithm Steps

Step	Description
1	Initialization: Set the number of clusters and the cluster centers to random.
2	For each 3 to 6 until convergence or a maximum number of iterations is reached.
3	Update Cluster Centers: Recalculate cluster centers using the updated membership degrees: $$y_k = \frac{\sum_{l=1}^{p} w_{kl}^{o} x_l}{\sum_{l=1}^{p} w_{kl}} \quad (6)$$
4	For $k = 1; k < e; k$: do{ $$\gamma = \sum_{k=1}^{e} \sum_{l=1}^{p} w_{kl}^{o} \|\|x_l - y_k\|\|^2 \quad (7)$$ while$(l < p)$}
5	For $k = 1; k < p; k$: For$(l = 1; l < p; l++)$\{ $$w_{kl} = \left(1 + \left(\frac{e_{kl}}{\gamma_k}\right)^{-\frac{1}{o-1}}\right)^{-1} \quad (8)$$ End of for End of for
6	End of for
7	End of for

3.3.2. Optimized Encoder-Decoder CNN In our research, a pivotal innovation lies in the incorporation of a key element an encoder-decoder based on an optimized Convolutional Neural Network (CNN). This renowned deep learning architecture, widely applied in tasks like image segmentation and classification, comprises two integral segments: an encoder network compressing input data into a latent representation and a decoder network

reconstructing output from this representation. The model unfolds in three phases: Encoder Phase: Within the optimized CNN encoder-decoder, the Encoder Phase manipulates input data to extract crucial features, generating a condensed latent representation. This phase abstains from specific details, concentrating on the fundamental transformation of data for further processing. Decoder Phase: In an optimized CNN encoder-decoder, the Decoder Phase reconstructs the output from the condensed latent representation, avoiding a detailed focus on specific aspects. Optimization Phase: The Optimization Phase involves a tailored set of procedures and techniques to enhance network performance and efficiency. These methods encompass adjusting hyperparameters, applying regularization techniques, utilizing optimization algorithms, and overall network architecture design, collectively aimed at improving efficiency. The Fuzzy C-Means (FCM) module serves a crucial role as the initial step in the model, generating fuzzy membership degrees for each clothing class in the Fashion-MNIST dataset. This fuzzy approach offers a nuanced representation of membership relations among images and different classes, reflecting the complexity of shared features among clothing categories. The fuzzy membership degrees from the FCM module are then seamlessly integrated into the optimized CNN encoder-decoder. The Encoder, employing convolutional layers, extracts significant features from input images while effectively reducing dimensionality. The synergy of convolution and pooling optimizes feature extraction, capturing intricate patterns and reducing redundant information. The Encoder's layer compresses extracted features, creating a dense and informative representation—a pivotal transition point between feature extraction and the reconstruction phase. The Decoder, consisting of deconvolution layers, takes the representation at the layer and reconstructs it into an image, preserving essential features. Deconvolutions enable the restoration of dimensionality while retaining crucial details. Illustrated in Figure 2, our optimized CNN encoder-decoder model comprises two CNN layers, showcasing the intricate interplay between feature extraction and reconstruction.

Figure 2 presents the conceptual architecture of our meticulously designed optimized neural network model, featuring a CNN encoder-decoder tailored explicitly for the classification task. in algorithm 1 the encoder commences with two convolutional layers (CNN), the first equipped with 64 filters and the second with 128 filters. Subsequent max-pooling layers facilitate the reduction of spatial dimensions, enabling the gradual extraction of pivotal features from the input data. The pivotal link between the encoder and decoder is forged by a dense layer housing 256 neurons and an activation function. This intermediary layer plays a crucial role in connecting the features extracted by the encoder to the subsequent decoding process. On the decoder side, two additional convolutional layers come into play. The initial layer, boasting 128 filters, implements an operation to augment spatial dimensions, followed by another layer with 64 filters and a distinct operation . These decoding operations are strategically designed to reconstruct spatial information from the features extracted by the encoder. At the model's output, the layer of fuzzy membership degrees, constituting a dense layer, accommodates several neurons equivalent to the number of classes in the classification task. This layer employs an activation function to generate normalized membership degrees, ranging from 0 to 1, for each class. Crucial model parameters, including the loss function, optimizer with a learning rate of 0.001, batch size of 32, and an iteration over several epochs (specifically 50 in this instance), are meticulously chosen to facilitate effective model learning while mitigating the risk of overfitting. It is imperative to highlight that these parameters remain adjustable, contingent upon the unique characteristics of the dataset and the specific requirements of the classification task.

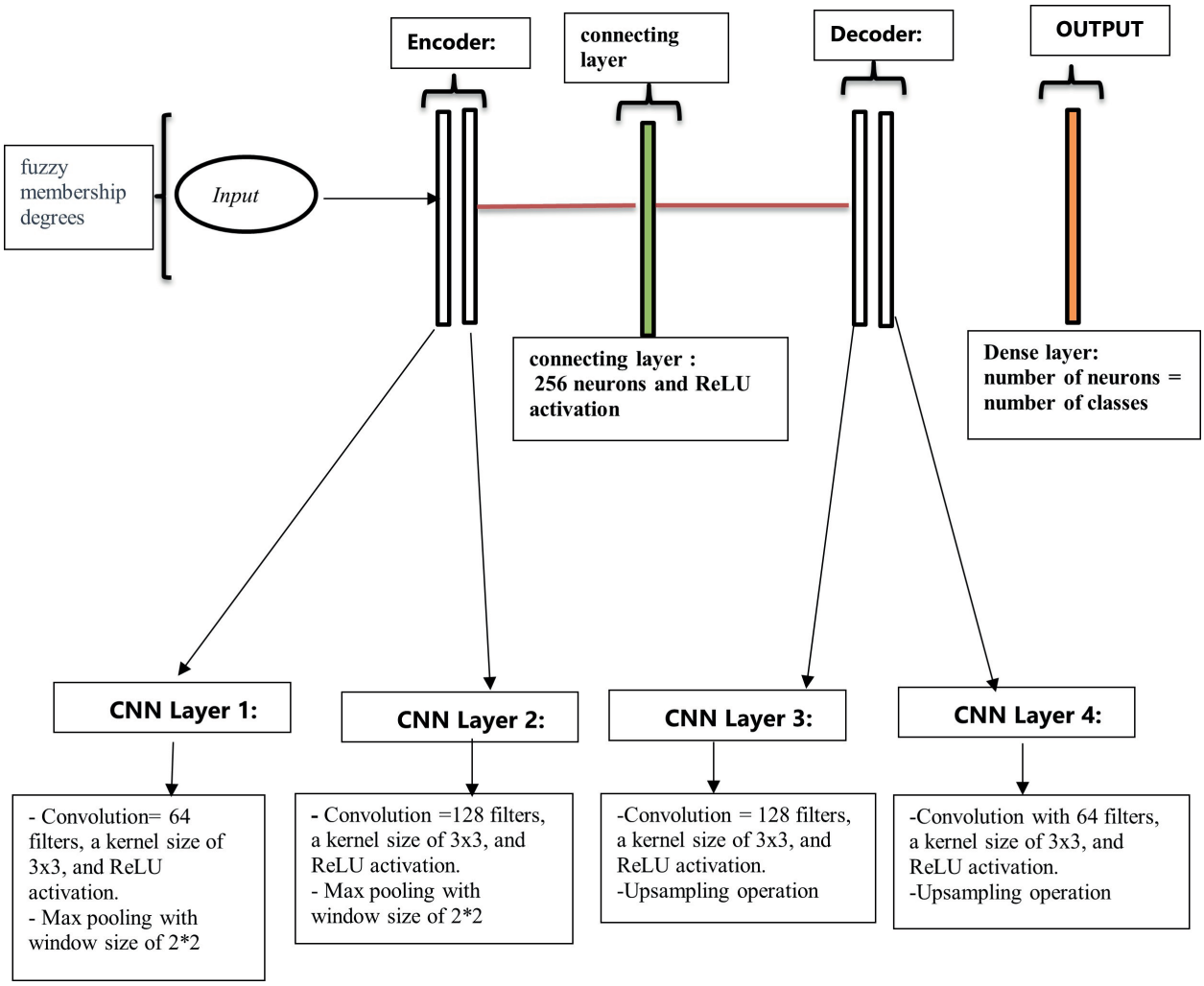


Figure 2. The Proposed Model

Algorithm 1 Optimized Encoder-Decoder CNN Algorithm

Step 0: FCM Clustering with Euclidean Distance Calculate centroids $Y = \{y_1, y_2, \ldots, y_K\}$ using the objective function:

$$\gamma = \sum_{k=1}^{e} \sum_{l=1}^{p} w_{kl}^{o} \|x_l - y_k\|^2 \quad (9)$$

Step 1: Encoder Training Loop Initialize cluster assignments for all images x_i Initialize the CNN architecture
for *each training iteration* t **do**
  for *each input image* x_i **do**
    Select x_i for the current iteration t Calculate CNN features using the encoder part of CNN Update cluster assignment $E(x_i)$ based on FCM-like update rules Perform backpropagation and update CNN weights using the assigned cluster as the target
  Move to the next training iteration $t + 1$
Repeat the training loop for a specified number of iterations or until convergence criteria are met

Step 2: Decoder Network for Image Reconstruction Input:

$$H = \text{enc}(Z) \left(\sum_{l_{(0..l_p)}}^{M_1 \ldots M_T} d_{k1\ldots k0}^2 + Y_{\beta k_1 \ldots k_o}^{(1)} \right) \quad (10)$$

Utilizing Transposed Convolutional Layers Training Neural Networks:

$$Y = Y - \phi \left(\frac{1}{o} \sum_{k=1}^{o} \phi Y + \delta d_k \right) \quad (11)$$

Understanding Backpropagation:

$$\Delta d = d - \phi \left(\frac{1}{o} \sum_{k=1}^{o} \delta d_k \right) \quad (12)$$

Forward propagation computes input and output values:

$$\rho_k^{(4)} = \left(\sum_{i=1}^{k1\ldots ko} i_k l (z_k^{(3)} - a_k) \right) \cdot h'(b_k^{(4)}) \quad (13)$$

$$\rho_{m1\ldots mT}^{(3)} = \left(\sum_{i=1}^{k1\ldots ko} i_k l \left(Y_{kl1\ldots mT}^{(3)} - \rho_k^{(4)} \right) \right) \cdot h'(b_k^{(4)}) \quad (14)$$

Compute output using Equation (11)

$$\Delta d = d - \phi \left(\frac{1}{o} \sum_{k=1}^{o} \delta d_k \right) \quad (12)$$

Update parameters the Y

3.3.3. Training and Optimization In this research, we conduct a comprehensive exploration aimed at enhancing the efficiency of our Convolutional Neural Network (CNN) model. Specifically, we introduce a novel strategy for optimizing the Fashion-MNIST dataset's performance through the application of Deep Convolutional Neural Networks. Our approach involves the implementation of an Encoder-Decoder Convolutional Neural Network (CNN) architecture. The utilized CNN architecture encompasses several layers, incorporating two Convolutional layers with ReLU activation functions and max-pooling layers for effective feature extraction. Subsequently, a flattened layer is employed to preprocess the data for subsequent fully connected layers. The network expansion involves the integration of an Encoder, comprising two fully connected dense layers characterized by the use of ReLU activation functions, as well as a Decoder with a similar structure. This evolution results in the creation of an output layer activated by the softmax function. A detailed summary of the recommended architectural design is presented comprehensively in Table 2 and Figure 3, providing an in-depth insight into the network structure and its various components.

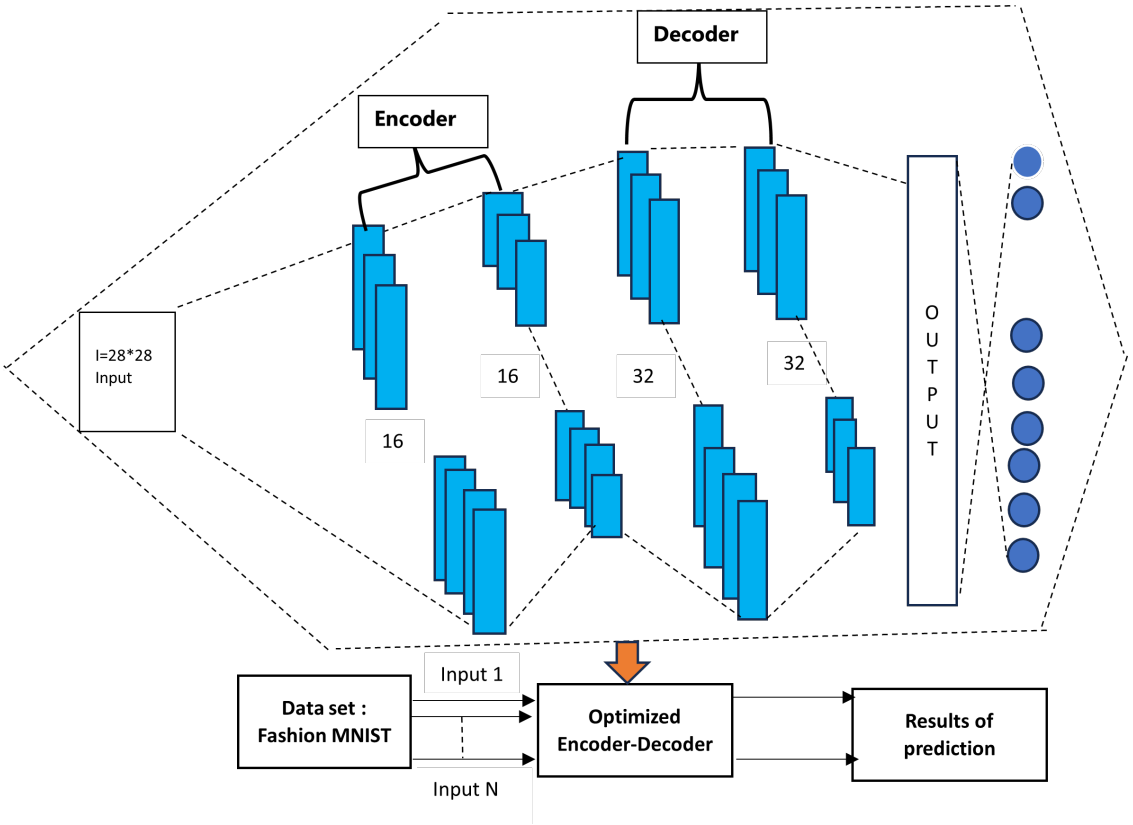


Figure 3. The architecture of the proposed method

We undertake the crucial task of optimizing model hyperparameters, focusing on parameters such as filter size, kernel dimensions, pool size, and the number of densely connected units. This step is designed to pinpoint the most effective configuration for our Encoder-Decoder Convolutional Neural Network (CNN), to improve the model's convergence during training and subsequently enhance its accuracy in classifying Fashion-MNIST.

To achieve robust classification, we introduce an innovative hybrid approach that combines Convolutional Neural Networks (CNNs) with Genetic Algorithms (GAs). In this pioneering methodology, GAs, drawing inspiration from natural selection principles, play a central role in systematically exploring and refining hyperparameter configurations within the existing CNN architecture. Across multiple generations, the genetic algorithm meticulously identifies and evolves the most promising hyperparameter combinations, effectively preparing the model for superior performance.

Table 2. Hyperparameters and Layer Descriptions

Layer	Output Shape (D × R × C)	Kernel/Pooling Size	N° of Parameters
Input Image	28×28	-	0
Convolutional Layer 1	32 × 12× 1	2 × 2	$(2 \times 2 \times 16 \times 32) + 32 = 2048$
MaxPooling Layer 1	16 × 14 × 1	2 × 2	0
Convolutional Layer 2	32 × 12 × 1	3 × 1	$(3 \times 16 \times 32) + 32 = 1568$
MaxPooling Layer 2	32 × 6 × 1	2	0
Flatten Layer	192	-	0
Dense Layer 3 (Hidden)	8	-	$(192 \times 8) + 8 = 1544$
Dense Layer 4 (Output)	10	-	$(8 \times 10) + 10 = 90$

Our research endeavors to offer valuable insights into the design and optimization of Deep Convolutional Neural Networks. The employed standard Genetic Algorithm 2, for optimizing the hyperparameters of the CNN neural network is outlined as follows:

3.1.3.1 Chromosome Representation The Genetic Algorithm (GA) was employed to address hyperparameters, encompassing filter size, kernel dimensions, pool size, and the number of densely connected units. In our methodology, each potential solution is symbolized by an individual chromosome within the Genetic Algorithm population. These chromosomes encode hyperparameter values as genes, enabling our algorithm to methodically explore and assess diverse architectural configurations. Refer to Figure 4 for a visual representation, illustrating how our genetic encoding captures hyperparameter values and facilitates comprehensive hyperparameter optimization. Following the establishment of the initial population, each individual undergoes evaluation and is assigned a fitness value determined by the fitness function.

Algorithm 2 Genetic Algorithm for Optimizing autoEncodeCNN Hyperparameters

Data: $P[0]$: Initial population of hyperparameter configurations generated randomly
Data: $pautoEncodeCNN$: Original autoEncodeCNN architecture
Data: Number of generations
Data: Termination condition
Result: Optimized autoEncodeCNN with tuned hyperparameters, Final trained autoEncodeCNN model with the best hyperparameters, Performance metrics (e.g., accuracy, precision, recall, F1-score, ROC curves) on testing data

Initialization:
Initialize $P[0]$ with random hyperparameter configurations Initialize new autoEncodeCNN with $pautoEncodeCNN$ Initialize Generation $i = 1$
Main Loop:
repeat
  for $i = 1$ *to the number of generations* **do**
    Update the autoEncodeCNN model in $P[i]$ with the architecture of $pautoEncodeCNN$ and random initial hyperparameters Train and evaluate the autoEncodeCNN model in $P[i]$ Calculate $Fitness[i]$ based on mean accuracy Perform Selection, Crossover, and Mutation to generate $NewP$ Increment Generation i by 1 Set $P[i] = NewP$
until *termination condition is satisfied*;
Final Step:
Choose the best-performing autoEncodeCNN hyperparameters from the final population based on the fitness values Initialize a new autoEncodeCNN model with the selected hyperparameters Train the final autoEncodeCNN model using the training data Evaluate the performance of the trained model on the testing data to obtain metrics Output the optimized autoEncodeCNN with tuned hyperparameters

3.1.3.2 Selection and Fitness function An individual's level of external adaptation can be assessed through a statistical metric generated by the fitness function. This metric is designed to focus on identifying traits that enhance an individual's adaptability or effectiveness in performing a given role. Our approach derives the fitness function from the average accuracy achieved through a 3-fold cross-validation process. The tournament selection

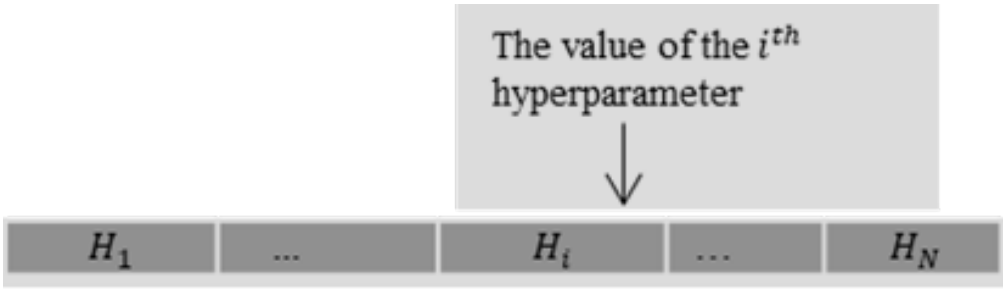


Figure 4. Chromosome representation of hyperparameters optimization

method is employed for both encoding and decoding tasks in our approach. This method is utilized to assess a population of Convolutional Neural Networks (autoEncodeCNNs) comprising both encoders and decoders. The evaluation of this population is contingent on their respective performances on the training data.

The tournament selection process involves creating "tournaments" among randomly chosen individuals within the population. In each tournament, two individuals are selected, and their performances are compared. The individual demonstrating superior performance is designated as the "winner" of the tournament. This process iterates until a sufficient number of individuals are chosen to constitute the next generation. The autoEncodeCNN encoders and decoders are subsequently ranked based on their performances. The top-ranked individuals, those attaining the highest performances, are selected to form the succeeding generation. This strategy ensures that the most high-performing individuals have an increased likelihood of being chosen for reproduction, thereby facilitating the transmission of genetic traits associated with their superior performances.

3.1.3.3 Reproduction Operators Following the selection phase, individuals from the reproduction pool are once again brought together, or crossed, to produce improved offspring. In this study, we have chosen the one-point crossover method. This process involves randomly selecting a crossover site along the genetic chain and exchanging alleles on one side of this site between the selected individuals.

In the context of a genetic algorithm, the two-point crossover [51] entails the exchange of alleles on both sides between parents, generating two distinct descendants. This approach facilitates the transmission of advantageous genetic traits from each parent to their offspring, introducing crucial variability into the population. By adjusting the crossover points, this method successfully generates diverse offspring, contributing across generations to the exploration and enhancement of solutions within the search space. The utilization of the two-point crossover represents a compromise between exploration and exploitation, as illustrated in Figure 5. This balance promotes genetic diversity while preserving beneficial traits, guiding the search towards potentially more promising solutions. The mutation operation plays a pivotal role in the genetic optimization process [52], entailing the random alteration of gene information within a population. In the context of this study, we have adopted the uniform mutation as our chosen mutation method. This method involves selecting a range of uniformly distributed values to replace the value of a specifically chosen gene within the uniform mutation operator. It's important to highlight that this operator is grounded in the Gaussian distribution. To illustrate this process, let's consider Figure 6. Suppose (x_i) represents a gene randomly selected from our chromosome and situated between (a_i, b_i). In this scenario, a random integer uniformly distributed in the range (a_i, b_i), denoted as $U(a_i, b_i)$, will be used to replace the current value of (x_i). The random replacement approach yields several advantages. Firstly, it introduces variability within the genetic population, fostering a more diversified exploration of the solution space. Additionally, the use of a range of uniform values ensures an equitable disturbance of each gene, mitigating potential biases introduced

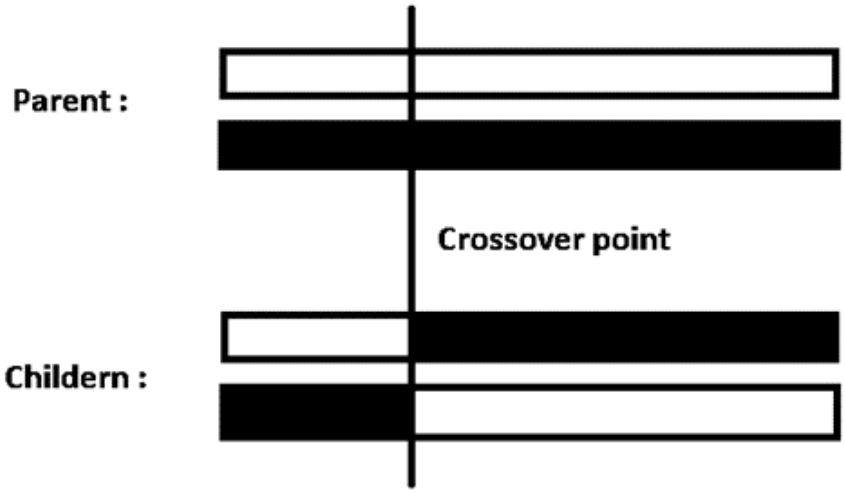


Figure 5. Crossover Operator

by other mutation methods. These genetic variations hold significance in the context of evolutionary algorithms, particularly in genetic algorithms, as they contribute to improving the likelihood of discovering optimal solutions.

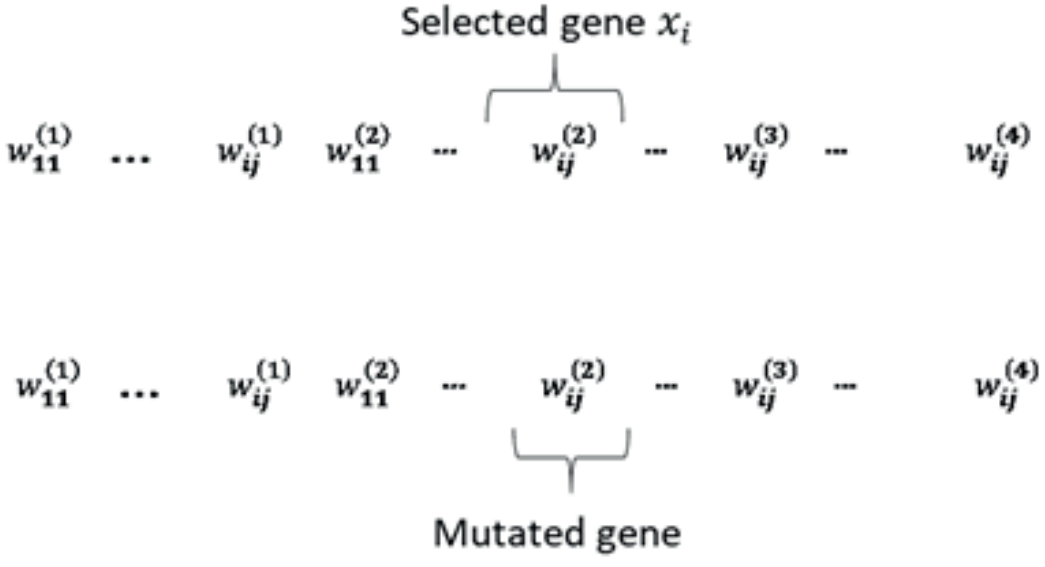


Figure 6. Mutation operator

4. Experiments and Results

In this section, our focus is squarely on evaluating our newly proposed method, encompassing both clustering and classification tasks, using datasets specifically designed for these purposes. We conduct an in-depth comparative analysis to assess the performance of our mechanism against other existing approaches or methods in both clustering and classification domains. The primary objective is to gain detailed insights into the effectiveness and relevance of our new method within the context of real-world data clustering and classification. This evaluation relies on appropriate metrics tailored to each task, allowing for a nuanced assessment of our method's capabilities. The results from this evaluation will play a pivotal role in understanding the robustness and applicability of our new method in real-world scenarios involving both clustering and classification of data. These findings will serve as a solid foundation for comprehending the strengths and specific application areas where our method excels, shedding light on its potential value in diverse real-world contexts.

4.1. Dataset details

To assess the effectiveness of our method, we have chosen to utilize the Fashion-MNIST dataset. This dataset comprises 60,000 training images and 10,000 test images, providing a comprehensive collection for evaluation. Each image in the dataset is a grayscale representation measuring 28x28 pixels, totaling 784 pixels per image. The dataset covers ten distinct categories of clothing, including items such as t-shirts, trousers, sweaters, dresses, coats, sandals, shirts, sneakers, handbags, and ankle boots. The Fashion-MNIST dataset stands out as a favored selection among both researchers and practitioners for assessing the effectiveness of machine learning algorithms, especially within the realm of image classification. Its importance stems from its capability to tackle the growing complexity and diversity found in real-world applications. Additionally, the dataset's accessibility through widely-used machine learning libraries has contributed to its widespread adoption in the fields of computer vision and machine learning research. This dataset has become a staple in the evaluation of algorithms due to its diverse range of grayscale images representing various clothing categories. It encapsulates the challenges inherent in image classification tasks, making it an ideal benchmark for assessing the robustness and adaptability of new approaches. Therefore, we have specifically chosen the Fashion-MNIST dataset to rigorously evaluate our novel approach. The decision is grounded in its seamless integration into experiments, facilitating the combination of learning processes, data extraction via clustering techniques, and ultimately, the classification of images. This comprehensive evaluation aims to showcase the applicability and efficacy of our method in handling real-world scenarios within the intricate domain of image classification.

4.2. Results and Discussion

In this section, we detail the results, highlighting the performances, findings, and implications stemming from our innovative approach. The analysis of collected data is conducted meticulously, and the presented results encompass a variety of parameters and indicators, thus illustrating the effectiveness and relevance of our innovative approach. To bolster the credibility and validity of our conclusions, comparisons are made with traditional methods or previous approaches.

Table 3. Training and Validation Accuracy Over Epochs 1/5

Epoch/5	Run Time	Loss	Accuracy	Validation Loss	Validation Accuracy
1	182s 19ms/step	0.5592	0.7924	0.4582	0.8297
2	214s 22ms/step	0.3856	0.8581	0.3762	0.8598
3	176s 18ms/step	0.3373	0.8748	0.3443	0.8750
4	155s 16ms/step	0.3086	0.8853	0.3358	0.8780
5	181s 19ms/step	0.2895	0.8925	0.3470	0.8702

Table 4. Training and Validation Accuracy Over Epochs 2/5

Epoch/5	Run Time	Loss	Accuracy	Validation Loss	Validation Accuracy
1	166s 17ms/step	0.5925	0.7818	0.4492	0.8366
2	157s 16ms/step	0.4031	0.8516	0.4029	0.8483
3	168s 18ms/step	0.3541	0.8700	0.3775	0.8601
4	263s 27ms/step	0.3219	0.8825	0.3367	0.8763
5	155s 16ms/step	0.3005	0.8889	0.3390	0.8752

Table 5. Training and Validation Accuracy Over Epochs 3/5

Epoch/5	Run Time	Loss	Accuracy	Validation Loss	Validation Accuracy
1	166s 17ms/step	0.5925	0.7818	0.4492	0.8366
2	157s 16ms/step	0.4031	0.8516	0.4029	0.8483
3	168s 18ms/step	0.3541	0.8700	0.3775	0.8601
4	263s 27ms/step	0.3219	0.8825	0.3367	0.8763
5	155s 16ms/step	0.3005	0.8889	0.3390	0.8752

Table 6. Training and Validation Accuracy Over Epochs 4/5

Epoch/5	Run Time	Loss	Accuracy	Validation Loss	Validation Accuracy
1	672s 70ms/step	0.5622	0.7937	0.4623	0.8314
2	201s 21ms/step	0.3844	0.8582	0.3644	0.8653
3	182s 19ms/step	0.3375	0.8766	0.3345	0.8776
4	183s 19ms/step	0.3065	0.8864	0.3298	0.8811
5	184s 19ms/step	0.2864	0.8940	0.3666	0.8668

Table 7. Training and Validation Accuracy Over Epochs 5/5

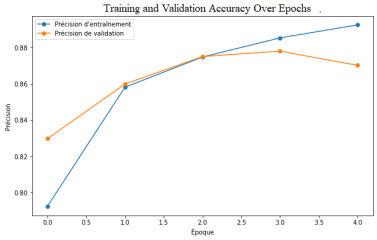
Epoch/5	Run Time	Loss	Accuracy	Validation Loss	Validation Accuracy
1	156s 16ms/step	0.5776	0.7886	0.4505	0.8324
2	145s 15ms/step	0.4046	0.8524	0.4101	0.8502
3	142s 15ms/step	0.3601	0.8674	0.3713	0.8643
4	152s 16ms/step	0.3304	0.8782	0.3441	0.8712
5	153s 16ms/step	0.3134	0.8845	0.3433	0.8791

Through a detailed examination of the Figure 7, tables 3,4, 5, 6, 7 depicting the evolution of training accuracy and validation accuracy over epochs, it is clear that the model consistently demonstrates an improvement in its performance. This steady progression is observed in both training accuracy and validation accuracy, indicating a robust learning capability of the model.

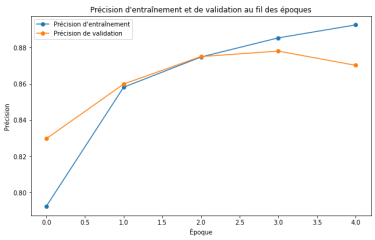
The model's learning efficiency is evident in how it assimilates information from the training data, resulting in a steady increase in accuracy over time. More importantly, this learning skill is reflected in the model's ability to generalize this knowledge beyond the examples on which it was trained, as confirmed by the similar progression of validation accuracy.

The consistent proximity between validation accuracy and training accuracy is an additional indicator of the model's performance. A low divergence between these two measures suggests that the model can generalize its learnings well to data it has not encountered before. This consistency in performance on both training and validation data is crucial to ensure that the model is not overfitting to specific training data but rather capable of making accurate predictions on new data.

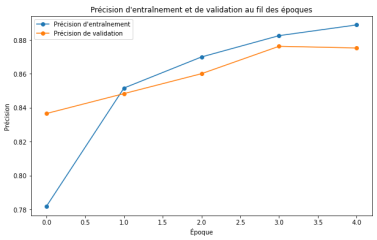
Performance differences between the different tables can be attributed to various factors, such as variations in model hyperparameters or other experimental conditions. These variations can influence the model's learning



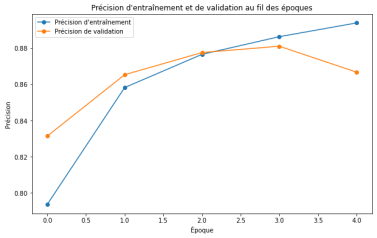
(a) Training and Validation Accuracy Over Epochs 1/5



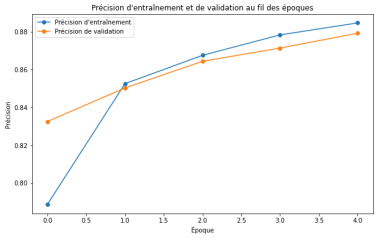
(b) Training and Validation Accuracy Over Epochs 2/5



(c) Training and Validation Accuracy Over Epochs 3/5



(d) Training and Validation Accuracy Over Epochs 4/5



(e) Training and Validation Accuracy Over Epochs 5/5

Figure 7. Training and Validation Accuracy Over Epochs

speed, its ability to capture complex patterns, or its resilience to overfitting. By examining these variations, it becomes possible to identify optimal model configurations for specific tasks.

Table 8 presents the training statistics of a model, organized in rows corresponding to different generations. Each generation is characterized by several parameters. The column ”gen” indicates the generation number, while ”nevals” represents the number of evaluations conducted during the respective generation. The columns ”avg,” ”min,” and ”max” provide crucial information about the model’s performance.

Analyzing the first generation (gen 0), it is observed that the model underwent 5 evaluations (nevals 5). The performances vary, with an average (avg) of 0.96498, a minimum performance (min) of 0.9517, and a maximum performance (max) of 0.9741. Subsequent generations (gen 1, gen 2, and gen 3) exhibit similar trends, with average values fluctuating between 0.97356 and 0.97482. The minimum and maximum performances also show a tendency to improve over the generations. Table 9, along with Figure 8, unveils the performance of our model during the

Table 8. Training Statistics

gen	nevals	avg	min	max
0	5	0.96498	0.9517	0.9741
1	5	0.97356	0.968	0.9797
2	5	0.97364	0.968	0.9753
3	3	0.97482	0.968	0.9802

first generation of the genetic algorithm. These representations provide a detailed overview of key performance metrics, namely the average (avg), minimum (Min), and maximum (Max). In Generation 1, the genetic algorithm demonstrated an average performance of 0.9748, with a minimum of 0.9680 and a maximum of 0.9802.

In addition to these statistics, the table provides essential information about the optimal hyperparameters discovered during this generation. The section dedicated to ”Best Hyperparameters” reveals that the ideal set of hyperparameters to maximize the algorithm’s performance is [62, 3, 22]. These values likely represent specific configuration parameters for the genetic algorithm, such as population size, mutation rate, and crossover points.

Figure 9 visually showcases carefully organized images along with crucial information about the optimal hyperparameters discovered during this generation. The dedicated section on ”Best Hyperparameters” unveils an

Table 9. Genetic Agorithm Performance :Generation 1

avg	Min	Max
0.9748199939727783	0.9679999709129333	0.9802000284194946
	Best Hyperparameters	[62, 3, 22]

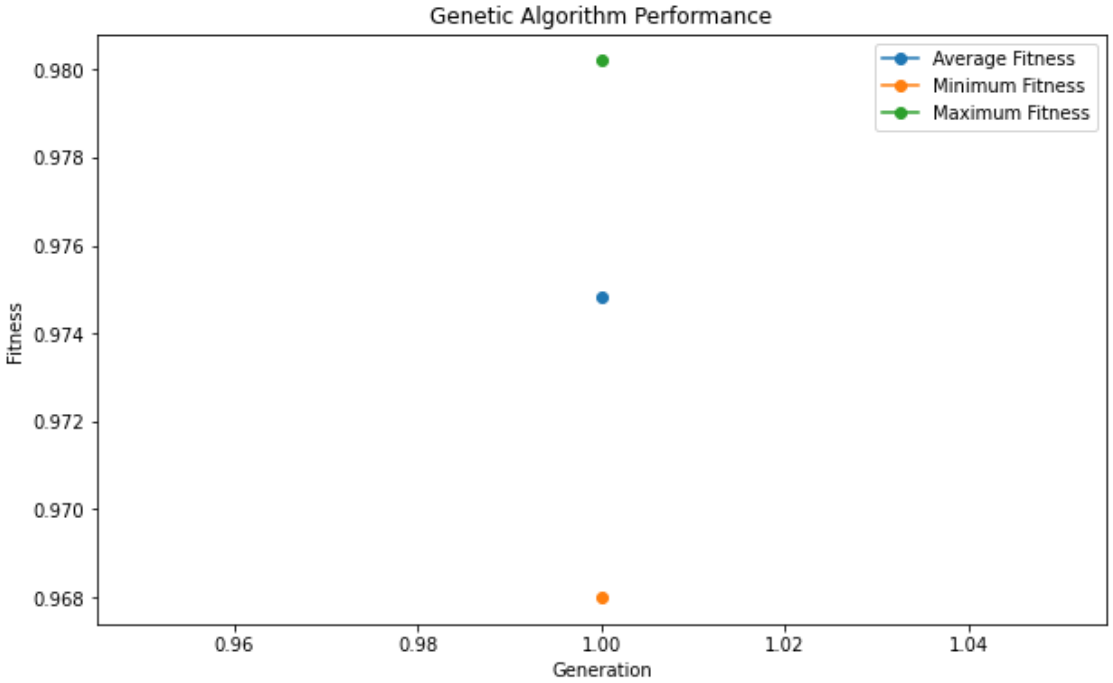


Figure 8. Genetic Algorithm Performance

optimal set, [62, 3, 22], intended to maximize the algorithm's performance. These values likely represent specific configuration parameters for the genetic algorithm, encompassing aspects such as population size, mutation rate, and crossover points. The image and its associated data provide a comprehensive overview of the achieved results, along with key adjustments that contributed to the effectiveness of the genetic algorithm. table 10provides a

Table 10. Genetic Algorithm Performance :Generation 2

avg	Min	Max
0.9748199939727783	0.9679999709129333	0.9802000284194946
	Best Hyperparameters	[62, 3, 22]

summary of the genetic algorithm's performance during its second generation. Key metrics, including average (avg), minimum (Min), and maximum (Max), are detailed. For Generation 2, the genetic algorithm achieved an average performance of 0.9748, with respective minimum and maximum values of 0.9680 and 0.9802. Additionally, the "Best Hyperparameters" section indicates that the optimal hyperparameters for this generation were [62, 3, 22], suggesting specific values for aspects such as population size, mutation rate, and crossover points.

The associated Figure 10, visually complements these data. The image provides a graphical representation of the outcomes during the second generation, allowing for a visual understanding of the genetic algorithm's performance. This combination of table and figure offers a comprehensive presentation of the algorithm's performance and key adjustments at this particular stage of its evolution.

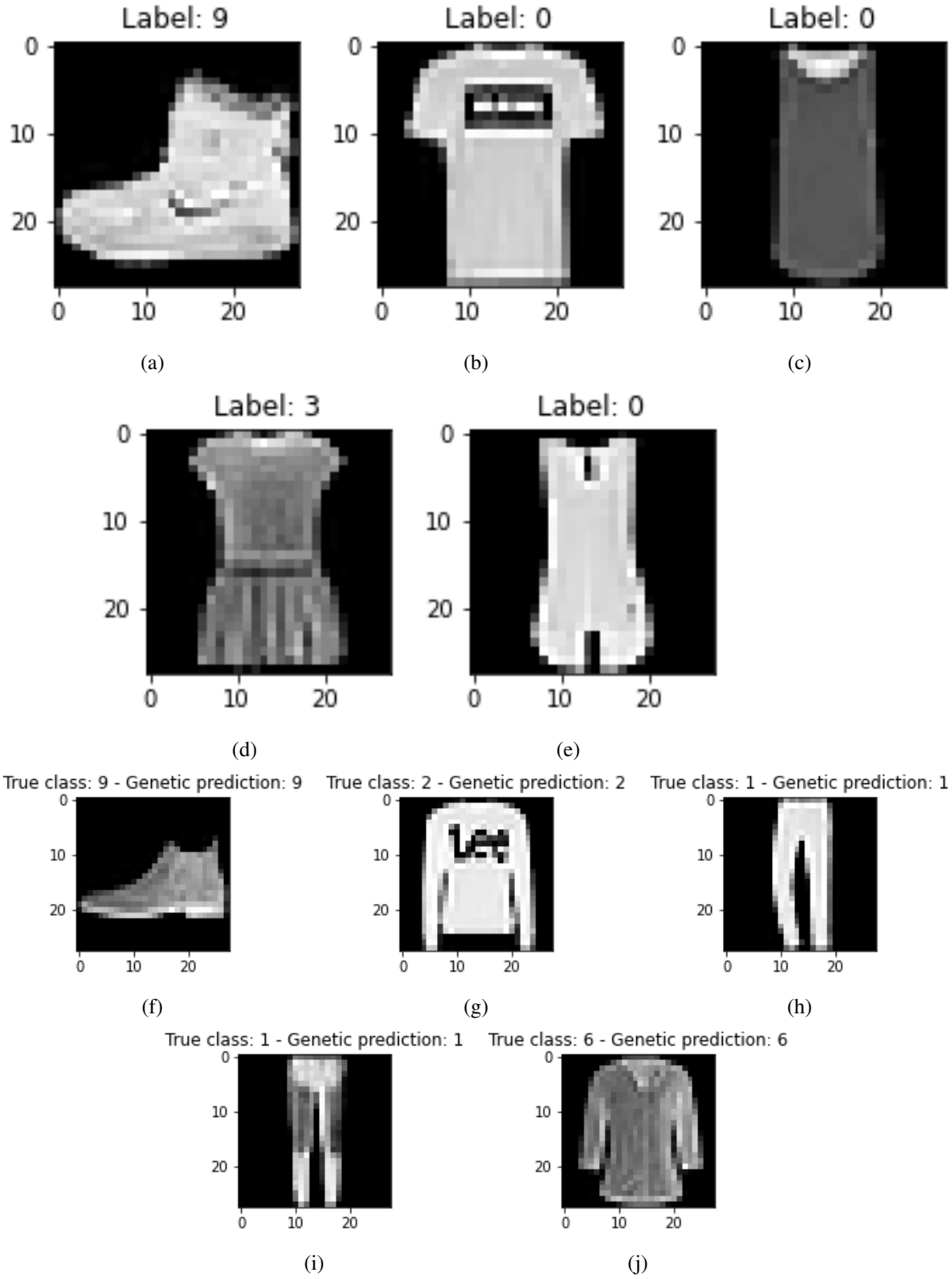


Figure 9. Results of the Images Before and After the Second Generation 1.

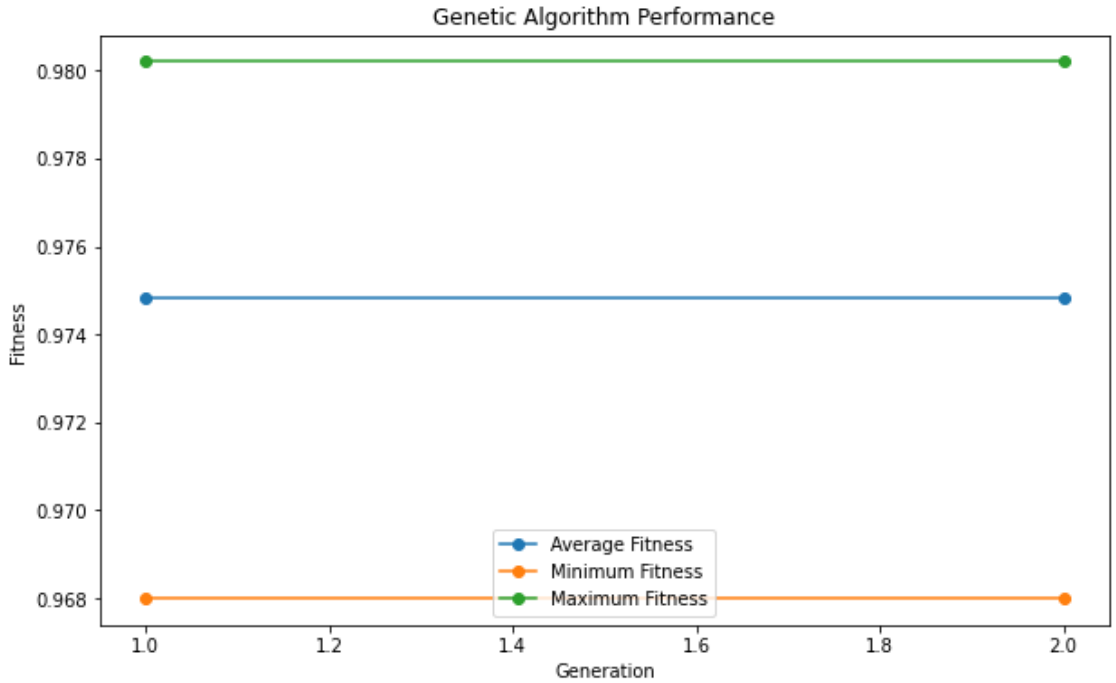


Figure 10. Genetic Algorithm Performance: generation 2

Figure 11 significantly contributes to enhancing the understanding of the data presented in the table. This graphical illustration provides a detailed visual representation of the outcomes during the second generation of the genetic algorithm. Each visual element in the figure, such as trends, variations, and highlights, complements the numerical information provided in the table. The image sheds light on the model's performance in a more tangible manner, allowing for a visual analysis of emerging patterns and performance fluctuations. The combination of the table and the figure creates a comprehensive and holistic presentation of the genetic algorithm's performance at this particular stage of its development. This amalgamation of visual and numerical elements facilitates an in-depth interpretation of the results and key adjustments, providing a thorough perspective on the model's performance.

Table 11 furnishes a concise overview of the genetic algorithm's performance in its second generation. The table meticulously outlines essential metrics, namely average (avg), minimum (Min), and maximum (Max). In Generation 2, the genetic algorithm demonstrated a noteworthy average performance of 0.9748, with corresponding minimum and maximum values of 0.9680 and 0.9802. Furthermore, insights into the optimal configuration are provided in the "Best Hyperparameters" section, revealing that the most effective hyperparameters for this generation were [62, 3, 22]. These values likely correspond to specific settings related to population size, mutation rate, and crossover points.

The accompanying Figure 12 serves as a visual complement to the tabulated data. This graphical representation offers a vivid portrayal of the outcomes observed during the second generation, facilitating a more intuitive understanding of the genetic algorithm's performance. The integration of both table and figure results in a comprehensive presentation that encapsulates the algorithm's achievements and notable adjustments at this particular phase of its development.

The detailed explanation of experimental configurations and adjustments of hyperparameters in the second generation of the genetic algorithm is crucial for understanding its effectiveness. The population size, set at 62, ensures sufficient genetic diversity to efficiently explore the possible solution space while remaining manageable from a computational perspective. This diversity is essential to avoid premature convergence to suboptimal solutions. The mutation rate is set at 3%, which represents a balance between maintaining genetic variability

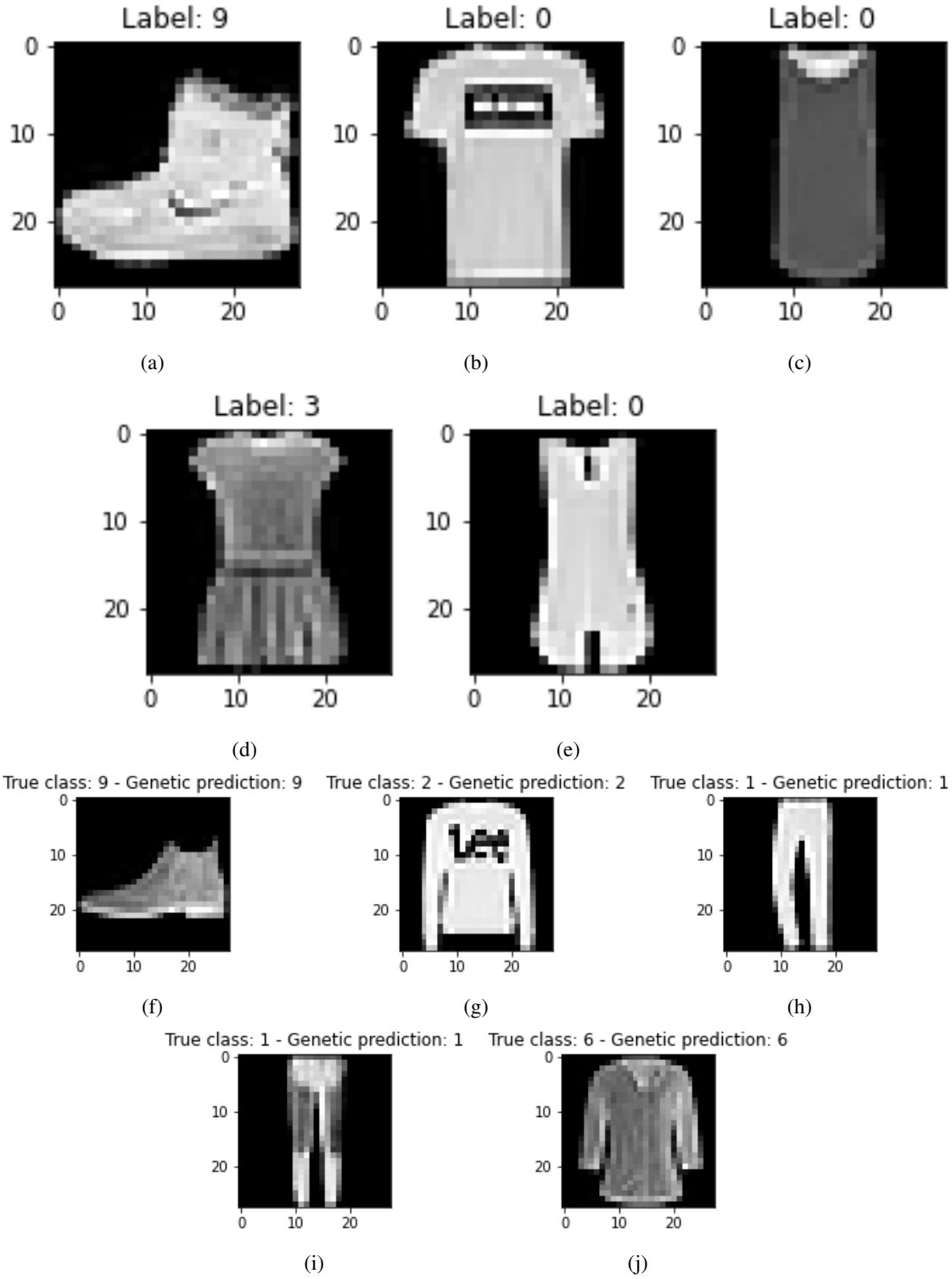


Figure 11. Results of the Images Before and After the Second Generation 2.

and preserving advantageous traits from existing configurations. This rate allows for the necessary variations to adapt to changing environments or new analytical demands without compromising the gains from previous generations. As for the crossover points, their number is fixed at 22. This choice affects how traits are inherited and combined during crossover, facilitating the creation of new configurations that could potentially outperform their predecessors in performance. This parameter is crucial to ensure that the algorithm does not stagnate and continues to evolve towards increasingly effective solutions. The selection method is rigorously applied to ensure that only the most effective configurations are retained for reproduction. This enhances the average quality of the population with each generation, relying on configurations that have demonstrated their efficacy in clustering and classification tasks. Overall, this strategic approach to hyperparameter tuning and population management ensures continuous improvement in the performance of the genetic algorithm, making it a robust and adaptable tool in the complex context of Big Data analysis.

Table 11. Genetic Algorithm Performance: Generation 3

avg	Min	Max
0.9748199939727783	0.9679999709129333	0.9802000284194946
	Best Hyperparameters	[62, 3, 22]

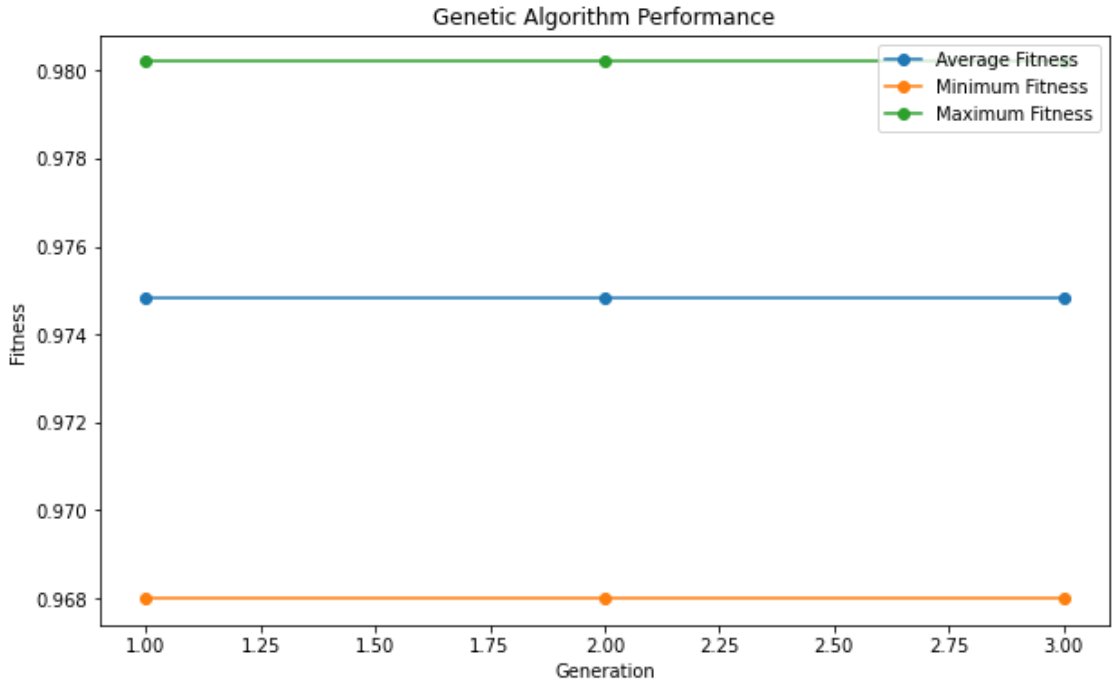


Figure 12. Genetic Algorithm Performance: Generation 3

Table 12. Results at Different Epochs Before and After Optimization

Epoch	1/5	2/5	3/5	4/5	5/5
Result before optimization	0.8075	0.8188	0.8257	0.8357	0.8457
Result after optimization	0.8806	0.9139	0.9208	0.9354	0.9455

Table 12 provides a detailed illustration of the evolution of our model's performance across five distinct epochs. Before optimization, the model exhibited initial results of 0.8075, 0.8188, 0.8257, 0.8357, and 0.8457

at epochs 1, 2, 3, 4, and 5, respectively. Following the application of the second generation of the algorithm, there was a notable and significant increase in performance. Post-optimization results demonstrated substantial improvements, reaching values of 0.8806, 0.9139, 0.9208, 0.9354, and 0.9455 at the corresponding epochs. These figures reveal a positive and ascending trend, highlighting the beneficial impact of the second generation on the model's performance at each stage of training. These results underscore the notion that optimization has led to significant changes, contributing to an overall enhancement in the model's visual outcomes over time. In this dedicated evaluation section, we subjected our method to rigorous testing by conducting a comparative analysis on the Fashion MNIST dataset, renowned for its complexity. Table 13 synthesizes the comparison between various existing mechanisms and our proposed model based on accuracy, ARI (Adjusted Rand Index), and NMI (Normalized Mutual Information). It is crucial to note that the basic Fuzzy C-means model achieved an accuracy of 52.91%, while the K-means model recorded 51.07%. However, other approaches such as IDEC, DEC, and DFCM surpassed these accuracy values, although our model maintained respectable performance. Regarding ARI as a comparison measure, it is remarkable that Fuzzy C-means reached an ARI of 36.44 %, while K-means achieved 36.39%. Our existing model demonstrated a significant improvement, notably with DFCM reaching 48.65% compared to the base model's 50.28%. Similarly, compared to other existing models, our enhanced FCM displayed a respectable ARI of 54.19Finally, considering NMI as a comparison metric, Fuzzy C-means reached 51.59%, K-means 51.64%, our existing model 66.09%, while our hybrid approach attained an impressive 67.35%. Figure 13 below provides a visual representation of the comparison between different existing models on the Fashion MNIST dataset.

Table 13. Clustering and Classification Approaches with Performance Measures

Approach	Accuracy (%)	ARI (%)	NMI (%)
K-means	51.07	36.39	51.64
Fuzzy C-Means	52.91	36.44	51.59
SEC	54.24	38.44	55.8
MBKM	50.00	34.5	50.03
IDEC	57.64	44.09	60.13
DEC	57.81	45.71	62.83
GrDFCM	62.78	50.14	65.78
DFCM	62.29	48.65	64.54
Our Method	94.71	68.66	78.3

The NMI is often used in information theory and data analysis to quantify the similarity between two data sets or the performance of clustering. A higher NMI indicates a stronger relationship or similarity between the variables.

$$\text{NMI}(X, Y) = \frac{H(X) + H(Y)}{2 \times I(X; Y)} \tag{6}$$

the Adjusted Rand Index is a valuable tool for assessing the quality of clustering algorithms by considering both the actual agreement and the agreement expected by chance. Provides a normalized measure that facilitates comparisons between different datasets and clustering methods.

$$\text{ARI}(X, Y) = \frac{\text{RI} - \mathbb{E}[\text{RI}]}{\max(\text{RI}_{\max} - \mathbb{E}[\text{RI}], 0)} \tag{7}$$

In the context of our comparative study between our new method and several widely used models with the Fashion MNIST dataset, we observed significant results in favor of our model. Our approach demonstrated superior performance in terms of data classification Figure 13.

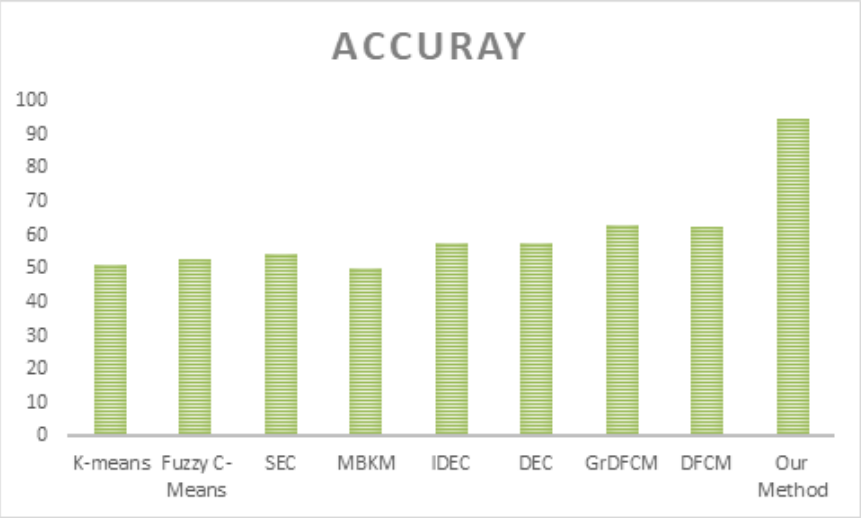


Figure 13. Comparative Evaluation of Different Models on the Fashion MNIST Dataset

5. Conclusion

During this in-depth study, we delved into the captivating realm of clustering and classification of voluminous data (big data). We introduced an innovative approach harnessing the fusion of three powerful techniques: FCM (Fuzzy C-Means), an optimized Encoder-Decoder Convolutional Neural Network (autoEncodeCNN), and genetic algorithm. Our primary objective was to address the inherent challenges in clustering and classifying extensive and complex datasets by leveraging the complementary strengths of these three powerful methods. The integration of FCM proved to be a crucial step, enabling an efficient initial clustering of data and significantly reducing the complexity of the problem. The optimized Encoder-Decoder CNN, specifically trained to extract essential data features, played a pivotal role in enhancing the quality of cluster representations. The inclusion of the genetic algorithm, with its capacity to model contextual information from past and future sequences, contributed significantly to improving the overall performance of clustering and data classification by accounting for temporal dependencies within the data. Experiments conducted with real-world big data datasets validated the effectiveness of our approach. It outperformed traditional clustering methods in terms of accuracy, scalability, and the ability to handle the high dimensionality and noise often present in extensive data. The fusion of FCM, the optimized Encoder-Decoder CNN, and the genetic algorithm not only yielded superior clustering results but also provided a more explicit and comprehensible representation of data clusters.

6. Data availability

I confirm the inclusion of a data availability statement. For your convenience, I have attached the data file. https://www.kaggle.com/datasets/zalando-research/fashionmnist

7. Conflict of Interest Statement

The authors declare no conflicts of interest regarding the publication of this research paper. The research was conducted in an unbiased and impartial manner, and the authors have no financial, professional, or personal

relationships that could be perceived as potentially influencing the research, analysis, or content presented in this paper.

8. Ethical Approval

This research, titled "Optimization of Big Data Analysis through a Hybrid Approach: Fuzzy C-Means, Encoder-Decoder CNN, and Genetic," has received ethical approval from the Universite Sidi Mohamed Ben Abdellah MOROCCO. This approval ensures that the study adheres to ethical considerations, including obtaining informed consent from participants, safeguarding the confidentiality and security of data, promoting beneficence and non-maleficence, maintaining integrity and transparency in reporting, and complying with all relevant regulations. The research team is committed to upholding the highest ethical standards throughout the study, prioritizing the well-being and rights of participants while making valuable contributions to the field of Big Data Analysis.

REFERENCES

1. Elkano, M., Sanz, J.A.A., Barrenechea, E., Bustince, H., Galar, M. CFM-BD: A distributed rule induction algorithm for building Compact fuzzy models in big data classification problems. IEEE Trans. Fuzzy Syst. March **2019**.
2. Wang, X.K., Yang, L.T., Liu, H.Z., Deen, M.J. A big data-as-a-service framework: State-of-the-art and perspectives. IEEE Trans. Big Data, 4(3): 325-340. https://doi.org/10.1109/TBDATA.2017.2757942 **2017**.
3. Chen, C.L.P., Zhang, C.Y. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. Inf. Sci. (Ny)., 275: 314-347.**2014**. https://doi.org/10.1016/j.ins.2014.01.015
4. Zhenhua Y, Guangwen Y, Shanwei L, Qishan Z (2010) A modified immune genetic algorithm for channel assignment problems in cellular radio networks. In: Intelligent system design and engineering application (ISDEA), 2010 International Conference on, vol 2. , pp 823–826
5. Bezdek, J. C., Ehrlich, R., Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. Computers Geosciences, 10(2-3), 191-203.
6. Li, Y., Sun, H., Wu, D. (2018). Clustering algorithms for massive data: A survey. IEEE Transactions on Knowledge and Data Engineering, 30(9), 1630-1648.
7. Johnson, R., Zhang, T. (2019). A survey of clustering algorithms. In M. A. Meiguins and al. (Eds.), Advances in Knowledge Discovery and Data Mining (pp. 29-47). Springer.
8. Zhang, X., Cheng, W., Chen, D. (2017). Big data clustering: A review. In 2017 IEEE Trustcom/BigDataSE/ICESS (Vol. 2, pp. 639-644). IEEE.
9. LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
10. Bas, E. A robust optimization approach to diet problem with overall glycemic load as objective function. *Appl. Math. Modelling* **2014**, *38*, 4926–4940.
11. El Moutaouakil, K.; Ahourag, A.; Chakir, S.; Kabbaj, Z.; Chellack, S.; Cheggour, M.; Baizri, H. Hybrid firefly genetic algorithm and integral fuzzy quadratic programming to an optimal Moroccan diet. *Math. Model.Comput.* **2023**, *10*, 338–350.
12. Ahourag, A.; Chellak, S.; Cheggour, M.; Baizri, H.; Bahri, A. Quadratic Programming and Triangular Numbers Ranking to an Optimal Moroccan Diet with Minimal Glycemic Load. *Stat. Optim. & Inf. Comput.* **2023**, *11*, 85–94.
13. El Moutaouakil, K.; Ahourag, A.; Chellak, S.; Baïzri, H.; Cheggour, M. Fuzzy Deep Daily Nutrients Requirements Representation. *Rev. Intell. Artif.* **2022**, 36 (2).
14. El Moutaouakil, K.; Baizri, H.; Chellak, S. Optimal fuzzy deep daily nutrients requirements representation: Application to optimal Morocco diet problem. *Math. Model. Comput.* **2022**, *9*, 607–615.
15. J. A. Richards, Remote Sensing Digital Image Analysis: An Introduction, New York, NY, USA: Springer, 2013.
16. F. M. Lacar, M. M. Lewis and I. T. Grierson, "Use of hyperspectral imagery for mapping grape varieties in the Barossa Valley South Australia", Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), vol. 6, **2001**, *27*, 2875-2877.
17. Zhang X-Y, Zhang J, Gong Y-J, Zhan Z-H, Chen W-N, Li Y (2016) Kuhn-Munkres parallel genetic algorithm for the set cover problem and its application to large-scale wireless sensor networks. IEEETrans Evol Comput 20(5):695–710
18. M. Volpi and D. Tuia, "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks", IEEE Trans. Geosci. Remote Sens., vol. 55, no. 2, pp. 881-893, 2017.
19. El Moutaouakil, K.; El Ouissari, A.; Palade, V.; Charroud, A.; Olaru, A.; Baïzri, H.; Chellak, S.; Cheggour, M. . Multi-Objective Optimization for Controlling the Dynamics of the Diabetic Population. *Mathematics* **2023**, *11* (13), 2957.
20. El Moutaouakil, K.; Palade, V.; Safouan, S.; Charroud, A. FP-Conv-CM: Fuzzy Probabilistic Convolution C-Means. *Mathematics* **2023**, *11* (8), 1931.
21. Hastie, T., Tibshirani, R., Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction.Springer.
22. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
23. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.
24. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
25. Freund, Y., Schapire, R. E. (1996). Experiments with a new boosting algorithm. In Machine learning: Proceedings of the thirteenth international conference (Vol. 96, pp. 148-156).

26. Chapelle, O., Schölkopf, B., Zien, A. (Eds.). (2009). Semi-supervised learning (Vol. 2). MIT press Cambridge.
27. Settles, B. (2009). Active learning literature survey. University of Wisconsin, Madison, 52(55-66), 11.
28. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).
29. Differential evolution and particle swarm optimization in partitional clustering Comput. Stat. Data Anal.(2006)
30. Caruana, R., Lou, Y., Gehrke, J., Koch, P. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1721-1730).
31. Ren, S., He, K., Girshick, R. Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).
32. Lopez de Prado, M. (2018). Advances in financial machine learning. John Wiley Sons.
33. Ahourag, A.; El Moutaouakil, K.; Cheggour, M.; Chellak, S.; Baizri, H. Multiobjective Optimization to Optimal Moroccan Diet Using Genetic Algorithm. *Int. J. Eng. Modelling* **2023**, *36*(1), 67–79.
34. El Moutaouakil, K.; Saliha, C.; Hicham, B.; Mouna, C. Intelligent Local Search Optimization Methods to Optimal Morocco Regime. *IntechOpen.* 2003, doi: 10.5772/intechopen.105600
35. Nama, S.; Sharma, S.; Saha, A.K.; Gandomi, A.H. A quantum mutation-based backtracking search algorithm. *Artif. Intell. Rev.* **2022**, 1–55.
36. Kim, J. H.; Kim, K.H.; Yoo, S.H. Evaluating and ranking the mining damage prevention programs in South Korea: An application of the fuzzy set theory. *Resour. Policy* **2022**, *78*, 102873.
37. El Moutaouakil, K.; Cheggour, M.; Chellak, S.; Baizri, H. Metaheuristics Optimization Algorithm to an Optimal Moroccan Diet, *7th Annual International Conference on Network and Information Systems for Computers (ICNISC)* **2021**.
38. Bas, E. (2016). A three-step methodology for GI classification, GL estimation of foods by fuzzy c-means classification and fuzzy pattern recognition, and an LP-based diet model for glycaemic control. Food Research International, 83, 1-13.
39. Nitesh, S.; Chawda, B.; Vasant, A. An improved K-medoids clustering approach based on the crow search algorithm. *J. Comput. Math. Data Sci.* **2022**, *3*, 100034.
40. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, 1989.
41. Agushaka, J.O.; Ezugwu, A.E.; Abualigah, L.; Alharbi, S.K.; Khalifa, H.A.E.W. Efficient initialization methods for population-based metaheuristic algorithms: a comparative study. *Arch. Comput. Methods Eng.* **2023**, *30* (3), 1727–1787.
42. Sharma, S., and Kumar, V. (2022). Application of genetic algorithms in healthcare: a review. Next Generation Healthcare Informatics, 75-86.
43. Zhou, J., and Hua, Z. (2022). A correlation guided genetic algorithm and its application to feature selection. Applied Soft Computing, 123, 108964.
44. BACK , T., EIBEN, A. E., AND VAN DER VAART, N. A. L. 2000. An empirical study on gas without parameters. In Proceedings of the 6th International Conference on Parallel Problem Solving from Nature, SpringerVerlag, 315–324.
45. GALVAN -LOPEZ, E., MCDERMOTT, J., O'NEILL, M., AND BRABAZON, A. 2010. Towards an understanding of locality in genetic programming. In Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation. 901–908.
46. DE JONG, K. A. 2002. Evolutionary Computation. MIT Press, Cambridge, MA.
47. Crepinsek, M., Liu, S. H., and Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. ACM computing surveys (CSUR), 45(3), 1-33.
48. M. Sarkar et al. A clustering algorithm using an evolutionary programming-based approach Pattern Recogn. Lett.(1997)
49. Sivanandam, S.N. Principles of soft computing. Deepa, S. N. Wiley: New Delhi, India, 2013. ISBN 978-1-118-54680-2. OCLC 891566849.
50. Zhou, J.; Hua, Z.. A correlation guided genetic algorithm and its application to feature selection. *Appl. Soft Comput.* **2022**, *123*, 108964.
51. Umbarkar, A. J.; Sheth, P.D. Crossover operators in genetic algorithms: a review. *ICTACT J. Soft Comput.* **2015**, *6*(1), 1083–1092.
52. Vasconcelos, J.A.; Ramirez, J.A.; Takahashi, R.H.C.; Saldanha, R.R. Improvements in genetic algorithms. *IEEE Trans. Magn.* **2001**, *37*(5), 3414-3417.
53. Biere, A.; Heule, M.; Van Maaren, H.; Walsh, T. (Eds.). Handbook of satisfiability (Vol. 185). IOS Press, 2009.
54. Watson, D. A practical approach to compiler construction (p. 254). Springer, 2017.
55. https://docplayer.fr/197944976-Programme-al-khawarizmi-liste-des-45-projets-retenus-pour-financement-classes-par-ordre-alphabetique-du-porteur-6-pages.html.
56. Nachate, S.; Kabbaj, Z.; Chakir, S.; Idrissi, A.; El Moutaouakil, K.; Baizri, H.; Cheggour, M.; Chellak, S. Which dietary survey methods for type 2 diabetics?. *Ann. Endocrinol.* **2023**, *84* (1), 192.
57. El Moutaouakil, K.; Yahyaouy, A.; Chellak, S.; Baizri, H. An optimized gradient dynamic-neuro-weighted-fuzzy clustering method: Application in the nutrition field. *Int. J. Fuzzy Syst.* **2022**, *24*, 3731-–3744.
58. A. Ghosh, and S.K. Pal, Object Background classification using Hopfield type neural networks. International Journal of Patten Recognition and Artificial Intelligence, 1992, pp. 989-1008..
59. El Moutaouakil, K.; El Ouissari, A.; Baizri, H.; Chellak, S.; Cheggour, M. Multi-objectives optimization and convolution fuzzy C-means: control of diabetic population dynamic. *RAIRO, Oper. Res.* **2022**, *56* (5), 3245–3256.
60. Asenjo Conrado, F. Variations in the nutritive values of foods. *Am. J. Clin. Nutr.* **1962**, *11* (5), 368–376.
61. Fanzo, J.; McLaren, R.; Davis, C.; Choufani, J. Climate change and variability. What are the Risks for Nutrition, Diets, and Food Systems. *IFPRI* **2017**, Discussion Paper 01645.
62. Donhouedé, J.C.F.; Salako, K.V.; Assogbadjo, A.E.; Ribeiro-Barros, A.I.F.; Ribeiro, N. The relative role of soil, climate, and genotype in the variation of nutritional value of Annona senegalensis fruits and leaves. *Heliyon* **2023**, *9* (8), e19012.