MATHEMATICAL MODELING AND COMPUTING, Vol. 11, No. 3, pp. 798–813 (2024)

Big data clustering through fusion of FCM, optimized encoder–decoder CNN, and BiLSTM

Belhabib F.[1], El Moutaouakil K.[1], Rbihou S.[2], Elafaar A.[1]

[1] *Sidi Mohamed Ben Abdellah University, Faculty of Sciences Polydisciplinaire, Taza, Morocco*
[2] *Engineering, Systems and Applications, Sidi Mohamed Ben Abdellah, ENSA, Fes, Morocco*

(Received 22 January 2024; Revised 15 August 2024; Accepted 7 September 2024)

Clustering Big Data, as a fundamental component in the processing and analysis of massive datasets, holds crucial importance in addressing complex challenges inherent in handling extensive data sets. Falling within the realm of unsupervised learning methods, the primary objective of clustering is to efficiently organize substantial datasets into homogeneous clusters without relying on pre-existing labels. Our innovative approach seeks to optimize this process by synergistically combining three techniques: the fuzzy C-Means (FCM) methodology, the optimized encoder–decoder CNN model, and the bidirectional recurrent neural network (BiLSTM). This synergy represents a strategic convergence between supervised and unsupervised paradigms. The introduction of BiLSTM is of significant importance, leveraging its capability to sequentially process data from both sides using LSTM cells. This bidirectional approach enhances the understanding of data sequences, a crucial feature in the demanding context of Big Data clustering. Simultaneously, FCM benefits from substantial improvement through the introduction of a function that calculates the separation between the cluster center and the instance, thereby reinforcing the precision of clustering. To optimize performance and reduce computation time, our methodology advocates for the use of the Optimized Encoder–Decoder CNN model. This refined architecture promotes more efficient extraction of data features, thereby enhancing the intrinsic quality of clustering. The rigorous evaluation of our approach revolves around specific data sources, namely fashion MNIST. Performance criteria such as accuracy, adjusted rand index (ARI), and normalized mutual information (NMI) convincingly attest to the remarkable capability of our methodology. In comparative analyses, our approach significantly outperforms existing models, demonstrating its effectiveness and relevance in the complex domain of Big Data clustering.

Keywords: *fuzzy C-Means (FCM); clustering; optimized encoder–decoder; clustering; bidirectional recurrent neural network (BiLSTM).*

2010 MSC: 68-xx, 90-xx **DOI:** 10.23939/mmc2024.03.798

1. Introduction

The explosion of Big Data [1] has ushered in a transformative era in scientific research, characterized by massive volumes of data, simultaneously offering unprecedented opportunities and complex challenges. At the core of this evolution, clustering emerges as a crucial technique in the exploration and structuring of these vast data sets. As an unsupervised analysis method, clustering reveals underlying structures, freeing the exploration process from the constraint of pre-existing labels and paving the way for the detection of complex and often nonlinear patterns. The clustering of Big Data [2] presents itself as an adaptive and sophisticated response to the diversified needs of researchers, aiming to extract meaningful knowledge from massive data sets without imposing pre-existing constraints. Falling within the category of unsupervised learning methods, clustering aims to identify underlying structures within the data, contributing to a profound understanding of the complex phenomena behind massive data. Our innovative approach relies on a synergistic methodological convergence, establishing a strategic alliance between three cutting-edge techniques, each bringing distinct and complementary characteristics. The

first component of our method is the Fuzzy C-Means (FCM) [3–7] methodology, renowned for its ability to handle massive data sets by assigning fuzzy membership degrees to each data point in different clusters. This fuzzy approach allows for a more flexible representation of relationships within the data, adapting to the inherent complexity of voluminous data sets. Our innovative approach is based on a synergistic methodological convergence, combining three advanced techniques: the Fuzzy C-Means (FCM) [7] methodology, the Optimized Encoder-Decoder CNN model [8, 9], and the Bidirectional Recurrent Neural Network (BiLSTM). This methodological fusion aims to leverage the unique advantages of each approach to address specific challenges related to Big Data [2] clustering. In the context of our innovative approach, we integrate the FCM methodology with the Optimized Encoder-Decoder CNN model, a robust technique dedicated to processing structured information. The Encoder-Decoder CNN model [10, 11] excels in capturing complex and spatial features, providing an in-depth view of inherent patterns in the data. This architecture, designed to efficiently process structured information, is particularly well-suited for the analysis of massive data. The meticulous optimization of this architecture significantly enhances its ability to extract discriminative information from the data, thereby increasing the overall accuracy of the clustering process. Indeed, the Encoder-Decoder CNN model [9, 12] excels in representing and interpreting complex structures, which proves particularly beneficial in the context of Big Data [8, 13, 14] where the variability and diversity of patterns can be substantial. By improving the model's ability to extract relevant and discriminative features, we aim to strengthen the intrinsic quality of clustering, contributing to a better understanding of underlying structures and a more precise interpretation of massive data. This synergy between the FCM methodology and the Optimized Encoder-Decoder CNN model represents a complementary and powerful approach to addressing specific challenges in Big Data [14] analysis. The centerpiece of our approach lies in the use of the Bidirectional Recurrent Neural Network (BiLSTM). Recurrent neural networks, and specifically BiLSTM, stand out for their exceptional ability to model complex temporal sequences. In the context of Big Data [14] clustering, this feature is crucial for detecting evolving trends over time, providing an essential dynamic perspective on the inherent structure of the data. Recurrent neural networks are designed to process sequential data while retaining internal memory, making them particularly suitable for representing complex temporal dependencies. BiLSTM, in particular, extends this capability by processing sequences in both directions through Long Short-Term Memory (LSTM) [15] cells. This bidirectional approach allows for more comprehensive modeling of temporal relationships in the data, which becomes essential in the context of Big Data clustering [14] where trends and temporal structures can be extremely varied and complex. In the realm of Big Data clustering [14], the use of BiLSTM adds significant value by enabling in-depth analysis of temporal sequences, facilitating the detection of evolving structures. This provides a dynamic and nuanced perspective on how data evolves, which is essential for understanding the inherent changes and variations in massive and complex data sets. By integrating BiLSTM as a key component of our approach, we aim to fully exploit this unique capability to enhance the relevance and quality of clustering in the demanding context of Big Data. The goal of our new integrated approach is to address specific challenges associated with Big Data clustering. By combining the flexibility of the FCM methodology, the feature-capturing ability of the Encoder-Decoder CNN model, and the temporal sequence modeling of BiLSTM, our methodology aspires to offer a robust and holistic solution for the analysis of massive data, thereby opening new perspectives for research and innovation in the field. 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 5, and finally a conclusion in Section 6.

2. Related works

In this section, we review various existing strategies for data clustering. First, there is a clustering method based on a dissimilarity matrix [16]. This approach transforms the dissimilarity matrix in a way that highlights distinct groups by representing them as dark blocks along the diagonal. This method

proves to be particularly effective in detecting halo-like structures in dark matter data, although it is primarily suited for large datasets. In our in-depth exploration of current data classification strategies, we highlight specific gaps within our domain and the inherent challenges of processing massive volumes of data. Among existing methods, those based on anomaly identification are frequently employed, but their real-time effectiveness often falls short, thereby exposing the system to potential intrusion risks. Approaches such as threshold-based anomaly detection or traditional statistical models show limitations, particularly in terms of responsiveness to the dynamics of Big Data [2, 17]. Our objective is to overcome these challenges by developing an innovative clustering methodology, judiciously integrating techniques such as Fuzzy C-Means (FCM) [19], Optimized Encoder-Decoder CNN [20], and Bidirectional Long Short-Term Memory (BiLSTM) [15]. This fusion aims to enable more precise and efficient real-time detection, leveraging the complementary strengths of each component. Another major challenge lies in the efficiency and speed of data classification systems, especially in the face of massive datasets. Conventional methods often face bottlenecks due to the high volume of data. In our research, we seek to address this challenge by adopting an innovative fusion approach that capitalizes on the distinct features of each component (FCM, Optimized Encoder-Decoder CNN, and BiLSTM) [16]. This holistic approach aims to significantly improve the speed and efficiency of our clustering methodology for Big Data processing. The issue of missing data is another crucial limitation highlighted in the literature [16]. In the context of our methodology, we focus on advanced techniques to effectively handle missing data, ensuring the quality and reliability of clustering results. The fusion of different approaches aims to create a robust and adaptive methodology that fully addresses this limitation. We conduct a thorough comparison of these existing approaches with our fusion methodology, highlighting the distinctive advantages and significant improvements that our approach brings in terms of anomaly detection, processing speed, and handling of missing data [21]. Our research is committed to transcending the limits of conventional data clustering approaches, adapting proven methods to specifically address the intrinsic challenges of Big Data processing [22]. We strive to present an innovative methodology that exceeds standards in real-time analysis, efficiency, and classification of massive data, aiming for a higher level of accuracy and reliability in the field of Big Data clustering.

3. Material and methods

3.1. The methodology

In this section, we delve into a detailed exploration of an innovative mechanism resulting from the hybridization of the FCM method (Fuzzy C-Means) [18], an optimized CNN encoder–decoder, and bidirectional convolutional neural networks (BiLSTM) to enhance the clustering process. The organization of this section into multiple parts aims to facilitate a thorough understanding of each component of the mechanism. To commence, we undertake a thorough analysis of the overall functioning of Fuzzy C-Means (FCM) [18], highlighting key elements and fundamental mechanisms that form the foundation of our hybrid approach. This initial step is meticulously crafted to establish a robust groundwork, enabling the reader to grasp the underlying theoretical principles of FCM. In the following series, we will introduce FCM-CNN-BiLSTM, a method created expressly to improve the clustering process's performance metrics. In this study, we explore how the optimized CNN encoder–decoder and bidirectional convolutional neural networks are included in the overall mechanism to improve the precision and effectiveness of the clustering process. Toward the end of our analysis, we combine two sub-mechanisms (FCM, or fuzzy C-means) and the hybrid FCM-CNN-BiLSTM technique smoothly. The result of this transparent merger is an improved and united organization. This consolidation aims to demonstrate the interoperability of the various parts and demonstrate how these complementing techniques when thoughtfully combined can result in notable advancements in the field of data clustering. The parameterization function that computes the separation between the data instance and the cluster center is then introduced. By facilitating an accurate assessment of the distance between data points and cluster centers, this function is essential to the clustering process and helps achieve a more refined and nuanced segmentation. From a practical standpoint, this fusion depends on the seamless integration

of the FCM processes, which offer a fuzzy clustering approach, and the FCM-CNN-BiLSTM hybrid approach, which leverages the benefits of both convolutional and bidirectional neural networks. By combining the best features of each technique, this combination seeks to produce a stronger, more effective strategy. Figure 1 provides a visual summary of our new hybridization method to accompany this presentation. It gives a clear visual representation of the final overall strategy and graphically depicts the interactions between the various process steps, emphasizing the excellent integration of the parts. Using this visual aid, we hope to increase the reader's understanding of how our hybrid approach is structured and coherent.

3.2. The proposed model

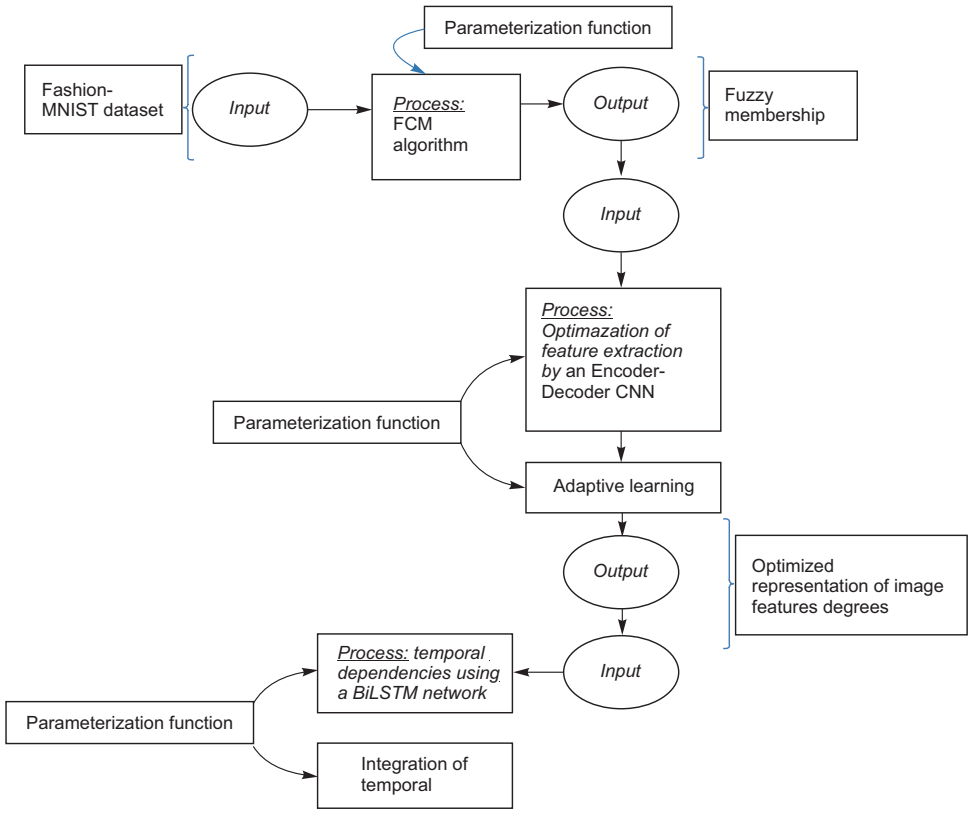

Fig. 1. The proposed model.

4. 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 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 x_i 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:

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

The normalized pixel value of the i-th image is represented as x_i'.

Let x_i represents 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 [20] models like Optimized Encoder-Decoder CNN and BiLSTMs.

4.1. FCM (Fuzzy C-Means)

We have developed a strategy that combines Euclidean distance with FCM (Fuzzy C-Means), a novel hybridization technique. A fuzzy clustering algorithm called FCM is used to group data according to similarities. Concurrently utilizing Euclidean distance and FCM is a widely used method to assess data point similarity. This method uses ordinary Euclidean geometry to calculate the distance between two points. The membership values assigned to each cluster are determined by the Fuzzy C-Means (FCM) algorithm, which takes into account the distances between each data point and the cluster centroids [14]. When working with overlapping data, FCM has a big benefit because it can produce reliable results. A data point may also be assigned to more than one cluster if needed. Some constraints, meanwhile, must be taken into account, including the amount of time needed for computing, accuracy, and the large number of iterations needed to reach convergence. Furthermore, Euclidean distance is a frequent metric used in FCM, which may give the data points varying weights. The FCM learning equation is utilized to update the membership degrees of data points in various clusters iteratively. The sum of the weighted Euclidean distances between data points and cluster centers raised to the power of the fuzziness parameter m is the fuzzy objective function, and its goal is to minimize it. The following is the learning equation for FCM:

$$\mu_{i,j}(t+1) = \frac{1}{\sum_{k=1}^{e} \left(\frac{d_{ij}}{d_{ik}}\right)^{\frac{2}{m-1}}},$$

where $\mu_{i,j}(t+1)$ is the membership degree of data point i to cluster j at iteration $t+1$, $d_{i,j}$ is the Euclidean distance between data point i and the center of cluster j, e is the total number of clusters, m is the fuzziness parameter ($m > 1$), t represents the iteration number.

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 $W = \{w_{kl} \mid 1 \leqslant k \leqslant q, 1 \leqslant l \leqslant p\}$. 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} elements represent the membership degrees of each data point k to each cluster l; FCM can be formulated,

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

$$\sum_{l=1}^{e} w_{KL} = 1, \quad w_{KL} \geqslant 0.$$

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

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

Membership matrix:

$$w_{kl} = \left[(1 + (e_{kl}/\gamma_k))^{-1/(o-1)}\right]^{-1}.$$

4.2. Optimized Encoder-Decoder CNN (Convolutional Neural Network)

In our study, we have incorporated a crucial element into our novel approach an encoder-decoder based on an optimized Convolutional Neural Network (CNN). This deep learning architecture [20] is commonly employed for tasks such as image segmentation and classification. The encoder–decoder model consists of two fundamental parts: an encoder network that compresses input data into a latent representation, and a decoder network that reconstructs the output from this representation. This model unfolds in three distinct phases: Encoder phase: In the context of an optimized Convolutional

Table 1. General fuzzy C-Means (FCM) algorithm.

Setp	Description
Input	Data normalization, max_pixel_value
Output	cluster member and membership FCM vector that has been optimized
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: $y_k = \frac{\sum_{l=1}^{p} w_{kl}^{o} x_l}{\sum_{l=1}^{p} w_{kl}}$
4	For(k=1; k<e; k++) do $y_k = \frac{\sum_{l=1}^{p} w_{kl}^{o} x_l}{\sum_{l=1}^{p} w_{kl}}$ while$(l < p)$
5	For(k=1; k<p; k++) For(l=1; l<p; l++) $w_{kl} = \left[\left(1 + \frac{e_{kl}}{\gamma_k}\right)^{-\frac{1}{o-1}}\right]^{-1}$
6	End of for
7	End of for

Neural Network (CNN) encoder–decoder, the general Encoder phase manipulates input data to extract crucial features, thereby generating a condensed latent representation. This phase refrains from delving into specific details, focusing on the fundamental transformation of data for further processing. Decoder phase: within an optimized Convolutional Neural Network (CNN) encoder–decoder, the general Decoder phase reconstructs the output from the condensed latent representation, avoiding a detailed focus on specific aspects. Optimization Phase: the optimization stage in a Convolutional Neural Network (CNN) encoder–decoder involves a specific set of procedures and techniques aimed at enhancing network performance and efficiency. These methods encompass adjusting hyperparameters, applying regularization techniques, utilizing optimization algorithms, and overall network architecture design. All these approaches are implemented to improve the overall efficiency of the network. The Fuzzy C-Means (FCM) module plays a crucial role as the first step in the model, generating fuzzy membership degrees for each clothing class in the Fashion-MNIST dataset. This fuzzy approach provides a more nuanced representation of the membership relations of images to different classes, reflecting the complexity of shared features among clothing categories. The fuzzy membership degrees generated by the FCM module are then integrated into the optimized CNN encoder–decoder. The Encoder, utilizing convolutional layers, extracts significant features from input images while effectively reducing dimensionality. This combination of convolution and pooling optimizes the feature extraction process by capturing complex patterns and reducing redundant information. The layer in the Encoder is crucial for representing the extracted features in a compressed manner, creating a dense and informative representation. This layer serves as a transition point between feature extraction and the reconstruction phase. The Decoder, composed of deconvolution layers, takes the representation at the layer and reconstructs it into an image preserving essential features. The use of deconvolutions allows the restoration of dimensionality while retaining important details. As illustrated in Figure 2, our optimized Convolutional Neural Network (CNN) encoder–decoder model comprises two CNN layers.

Figure 2 illustrates the conceptual structure we have developed for this optimized neural network model with a CNN encoder–decoder, specifically designed for the classification task. The encoder begins with two convolutional layers (CNN), the first containing 64 filters and the second with 128 filters, followed by max-pooling layers to reduce spatial dimensions. This process allows for the progressive extraction of crucial features from the input data. The connection between the encoder and the decoder is facilitated by a dense layer with 256 neurons and an activation function. This layer plays a crucial role in linking the features extracted by the encoder to the decoding process. As for the decoder, it starts with two additional convolutional layers, the first with 128 filters and an operation to increase spatial dimensions, followed by another layer with 64 filters and a new operation. These decoding operations aim to reconstruct spatial information from the features extracted by the encoder. Finally, the layer of fuzzy membership degrees, located at the model's output, is a dense layer with several

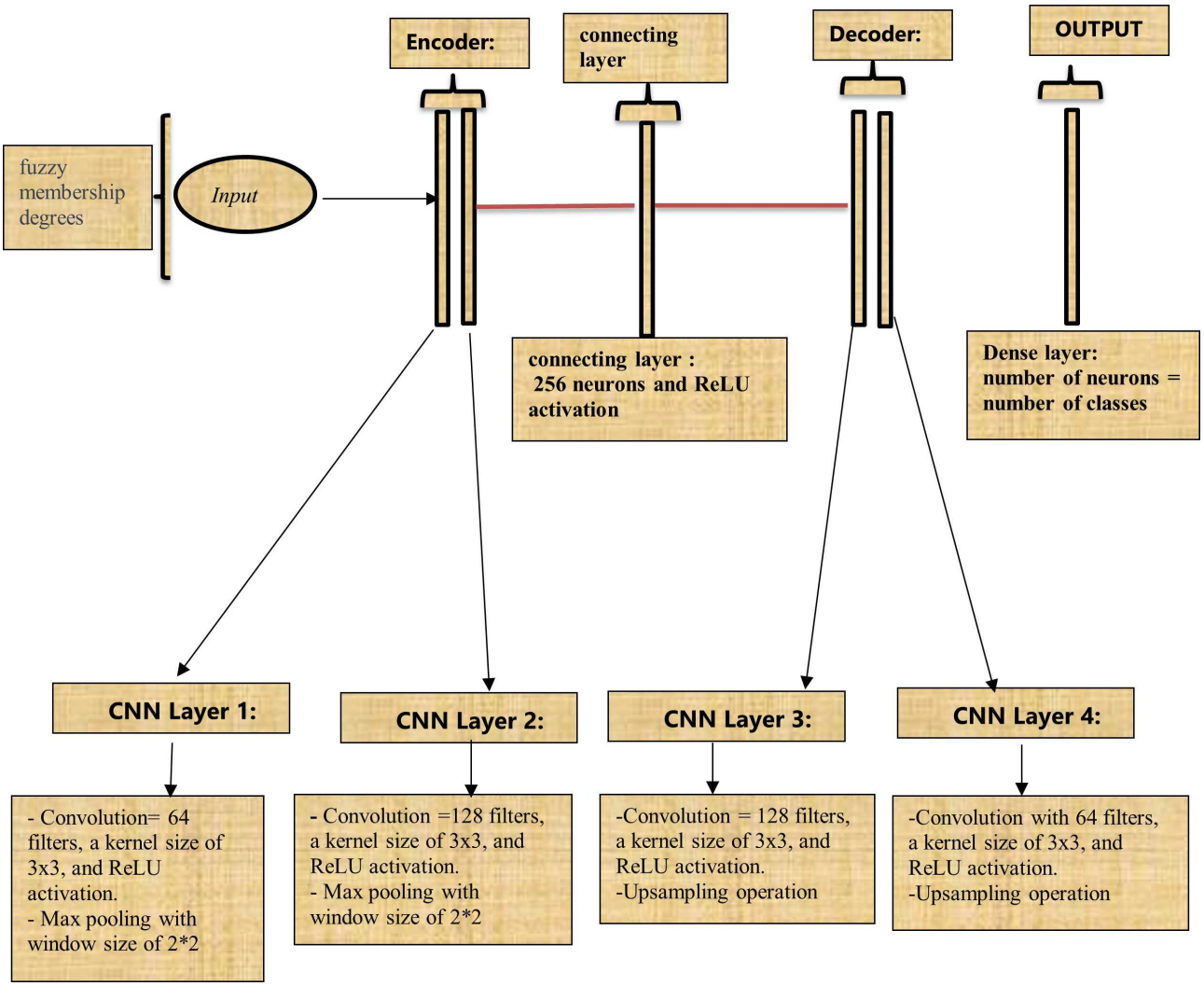

Fig. 2. Our optimized Convolutional Neural Network (CNN) encoder–decoder.

neurons equivalent to the number of classes in the classification task. This layer uses an activation function to generate normalized membership degrees, ranging from 0 to 1, for each class. The model parameters, such as the loss function, optimizer with a learning rate of 0.001, batch size of 32, and several epochs of 50, are chosen to facilitate model learning while avoiding overfitting. It is crucial to note that these parameters can be adjusted based on the specific characteristics of the dataset and task requirements. In summary, this structure aims to create a model capable of efficiently encoding information, representing it concisely, and decoding it accurately for the classification task, while using fuzzy membership degrees for a more nuanced representation of classes.

4.3. BiLSTM (Bidirectional Long Short-Term Memory)

Once the data had undergone preprocessing by the CNN encoder–decoder, we seamlessly integrated the BiLSTM network. BiLSTM, being proficient in modeling sequential dependencies within the data, operates bidirectionally, diligently scrutinizing both past and future sequences. Through the incorporation of BiLSTM after the CNN encoder–decoder process, we ushered in a sequential analysis of the extracted features, an indispensable step toward comprehending the underlying sequential patterns within the data. A BiLSTM layer is an innovation in the field of recurrent neural networks, commonly used in natural language processing (NLP) and sequence modeling. It builds upon the foundation of LSTM (Long Short-Term Memory) by introducing the capability to capture contextual information from both the past and the future of a sequence. This bidirectional approach allows the model to better grasp the overall context of the sequence because it processes data in both directions, from the

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:

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

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:

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

Utilizing Transposed Convolutional Layers
Training Neural Networks:

$$Y = Y - \phi \left(\frac{1}{o} \sum_{k=1}^{o} \phi Y + \delta d_k \right) \tag{2}$$

Understanding Backpropagation:

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

Forward propagation computes input and output values:

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

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

Compute output using Eq. (2)

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

Update parameters the Y

$$Y = Y - \alpha \Delta Y$$

Output: Repeat for Each Cluster Eq. (1)

forward to the backward and vice versa, for each time step. The outputs from both directions are then combined to provide a single output for each time step. BiLSTM layers are particularly valuable for tasks such as sequence classification, sequence labeling, segmentation, and machine translation, as they enhance the model's ability to capture context more comprehensively.

Algorithm 2 Bidirectional LSTM Forward and backward pass

Data: Input sequence of length T, LSTM parameters W_f, W_i, W_C, W_o, b_f, b_i, b_c, b_o
Result: Hidden states h_t and cell states C_t
Initialization:
Initialize input sequence of length T
Initialize LSTM_forward with parameters W_f, b_f
Initialize LSTM_backward with parameters W_f^b, b_f^b
Initialize hidden state h_T^b for backward pass
Forward Pass:
for $t = 1$ to T **do**
    $f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big)$ // Forget gate
    $i_t = \sigma\big(W_i \cdot [h_{t-1}, x_t] + b_i\big)$ // Input gate
    $C_t = \tanh\big(W_C \cdot [h_{t-1}, x_t] + b_c\big)$ // Cell state update
    $C_t = f_t \cdot C_{t-1} + i_t \cdot C_t$ // Updated cell state
    $o_t = \sigma\big(W_o \cdot [h_{t-1}, x_t] + b_o\big)$ // Output gate
    $h_t = o_t \cdot \tanh(C_t)$ //Updated hidden state
Backward Pass:
for $t = T - 1$ to 1 **do**
    $f_t^b = \sigma\big(W_f^b \cdot [h_{t+1}^b, x_t] + b_f^b\big)$ // Forget gate backward
    $i_t^b = \sigma\big(W_i^b \cdot [h_{t+1}^b, x_t] + b_i^b\big)$ // Input gate backward
    $C_t^b = \tanh\big(W_C^b \cdot [h_{t+1}^b, x_t] + b_c^b\big)$ // Cell state update backward
    $C_t^b = f_t^b \cdot C_{t+1}^b + i_t^b \cdot C_t^b$ // Updated cell state backward
    $o_t^b = \sigma\big(W_o^b \cdot [h_{t+1}^b, x_t] + b_o^b\big)$ // Output gate backward
    $h_t^b = o_t^b \cdot \tanh(C_t^b)$ // Updated hidden state backward

5. Performance evaluation

In this section, the focus is on evaluating our new proposed method using datasets specifically dedicated to clustering. A comprehensive comparative analysis is conducted to assess the performance of the mechanism compared to other existing approaches or methods. The aim is to gain precise insights into the effectiveness and relevance of our new method in the context of real data clustering. The results of this evaluation will provide crucial information for assessing the robustness and applicability of our new method in real-world scenarios of data clustering.

5.1. Dataset details

Using the Fashion-MNIST dataset, we chose to evaluate the efficacy of our approach [1, 16]. Ten thousand test photos and sixty thousand training images make up this dataset. These pictures are all 28×28 pixels in size, or 784 pixels in total, represented in grayscale. It includes products like t-shirts, pants, sweaters, skirts, coats, sandals, shirts, sneakers, handbags, and ankle boots, among other clothing that falls into ten different categories. The Fashion-MNIST dataset is frequently used by academics and industry professionals to assess how well machine learning algorithms work, especially when it comes to picture classification. To handle the growing diversity and complexity of real-world applications, this database is essential. It is also commonly used in computer vision and machine learning research, and it is readily accessible through well-known machine learning libraries. We therefore chose this dataset to evaluate our novel method, which integrates learning, data extraction via clustering approaches, and, at the end, classification. For each training period, the performance results for an optimization model with an encoder and a decoder using a first convolutional layer (CNN) are displayed in Table 2 and Figure 3. The "loss" column displays the model's training loss, which quantifies the degree to which the model's predictions deviate from the actual values. The main goal is to increase the accuracy of the model by minimizing this loss throughout epochs. By expressing the percentage of accurate predictions relative to the total number of samples, the "accuracy" column sheds light on the model's performance on the training set. The validation set's corresponding metrics are shown in parallel in the "val-accuracy" and "val-loss" columns. To prevent overfitting of the training

data, a low validation loss indicates that the model generalizes well. The model's capacity to produce accurate predictions on fresh data is confirmed by an increase in validation accuracy. Monitoring the training process across various batches of samples is made possible by batch tracking. The way these metrics have changed throughout epochs is one way to evaluate how well the model has learned. On both the training set and the validation set, the ideal outcome is to see a drop in loss and an improvement in accuracy. These patterns show that a model is learning efficiently and adapting well to fresh input.

Table 2. Performance results for an optimization model consisting of an encoder and a decoder with the first convolutional layer (CNN).

Epoch/10	Batch	Run Time/step	Loss	Accuracy	Val Loss	Val Accuracy
1	3000/3000	22 s 7 ms/step	0.5599	0.7984	0.3973	0.8602
2	3000/3000	21 s 7 ms/step	0.3609	0.8731	0.3397	0.8824
3	3000/3000	20 s 7 ms/step	0.3137	0.8896	0.3345	0.8809
4	3000/3000	21 s 7 ms/step	0.2844	0.9006	0.2993	0.8969
5	3000/3000	22 s 7 ms/step	0.2631	0.9069	0.2864	0.9003
6	3000/3000	22 s 7 ms/step	0.2426	0.9160	0.2880	0.9017
7	3000/3000	22 s 7 ms/step	0.2284	0.9180	0.2841	0.9027
8	3000/3000	21 s 7 ms/step	0.2134	0.9237	0.2983	0.8972
9	3000/3000	21 s 7 ms/step	0.2015	0.9277	0.2773	0.9063
10	3000/3000	20 s 7 ms/step	0.1918	0.9312	0.2887	0.9057

Table 3 and Figure 4 provide a detailed exploration of the performance of an optimization model, consisting of an encoder and a decoder with the integration of the second convolutional layer (CNN) at each training epoch. The "loss" column exposes the measure of the model's training loss, quantifying the gap between the model's predictions and the actual values. The central objective is to reduce this loss over epochs, aiming to refine the model's precision. Special attention is devoted to understanding the performance of our model in this context. The "accuracy" column offers a detailed perspective on the model's precision on the training set, quantifying the proportion of correct predictions relative to the total number of samples. In parallel, the "val loss" and "val accuracy" columns present equivalent metrics specifically for the validation set. A reduced validation loss emphasizes the model's ability to generalize effectively, avoiding overfitting the training data. The confirmation of this generalization through an increase in validation accuracy indicates the model's capability to make accurate predictions on new data. Batch tracking allows for close monitoring of the learning progression across various batches of samples. Observing the evolution of these metrics over epochs permits a thorough evaluation of the model's learning effectiveness. Our attention is particularly drawn to the nuances of performance, ideally seeking a decrease in loss and an increase in accuracy not only on the training set but also on the validation set. These detailed trends serve as crucial indicators revealing the robustness and effectiveness of the model in generalizing to new data.

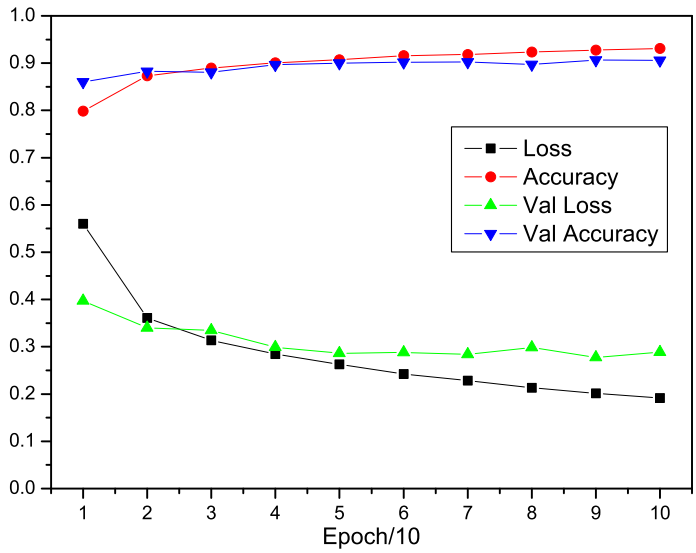

Fig. 3. Performance results for an optimization model consisting of an encoder and a decoder with a first convolutional layer (CNN).

Table 3. Performance results for an optimization model consisting of an encoder and a decoder with a second convolutional layer (CNN).

Epoch/10	Batch	Run Time/step	Loss	Accuracy	Val Loss	Val Accuracy
1	3000/3000	20 s 6 ms/step	0.4578	0.8375	0.3427	0.8788
2	3000/3000	19 s 6 ms/step	0.3114	0.8902	0.3159	0.8857
3	3000/3000	18 s 6 ms/step	0.2694	0.9040	0.2906	0.8956
4	3000/3000	18 s 6 ms/step	0.2411	0.9139	0.2768	0.9013
5	3000/3000	19 s 6 ms/step	0.2188	0.9208	0.2752	0.9018
6	3000/3000	18 s 6 ms/step	0.1991	0.9283	0.2810	0.9011
7	3000/3000	18 s 6 ms/step	0.1805	0.9341	0.2700	0.9068
8	3000/3000	19 s 6 ms/step	0.1667	0.9401	0.2881	0.9066
9	3000/3000	18 s 6 ms/step	0.1520	0.9444	0.3011	0.9051
10	3000/3000	18 s 6 ms/step	0.1410	0.9489	0.3039	0.9047

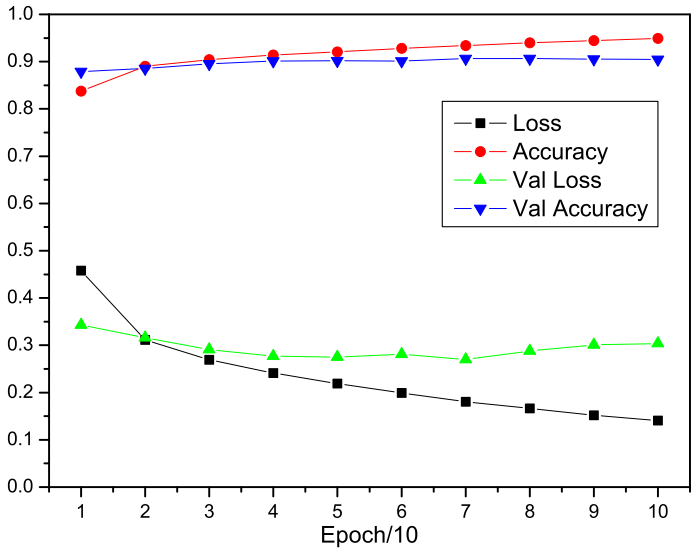

Fig. 4. Performance results for an optimization model consisting of an encoder and a decoder with a second convolutional layer (CNN).

Table 4 and Figure 5 provide detailed insights into the performance metrics specifically associated with the first layer of a BiLSTM model over ten training epochs. Each row corresponds to a specific epoch, presenting key details related to the training and validation of the first layer. The training loss of the first layer decreases from epoch 1 to epoch 10, suggesting a continuous improvement in the first layer's ability to capture patterns within the training data. Additionally, the training accuracy of the first layer increases, indicating the first layer's proficiency in making more precise predictions on the training set.

Table 4. Performance Results for a BiLSTM Model (First Layer).

Epoch/10	Batch	Timer	Loss	Accuracy	Val Loss	Val Accuracy
1	3000/3000	16 s 5 ms/step	0.4274	0.8518	0.3150	0.8905
2	3000/3000	14 s 5 ms/step	0.2891	0.8970	0.2998	0.8932
3	3000/3000	14 s 5 ms/step	0.2470	0.9100	0.2742	0.9010
4	3000/3000	14 s 5 ms/step	0.2183	0.9205	0.2733	0.8989
5	3000/3000	14 s 5 ms/step	0.1934	0.9283	0.2646	0.9074
6	3000/3000	15 s 5 ms/step	0.1745	0.9354	0.2575	0.9105
7	3000/3000	14 s 5 ms/step	0.1572	0.9416	0.2735	0.9048
8	3000/3000	14 s 5 ms/step	0.1414	0.9478	0.2638	0.9130
9	3000/3000	18 s 6 ms/step	0.1297	0.9521	0.2948	0.9080
10	3000/3000	24 s 8 ms/step	0.1164	0.9576	0.2935	0.9093

Table 5 and Figure 6 provide detailed insights into the performance metrics specifically associated with the second layer of a BiLSTM model over ten training epochs. Each row corresponds to a specific epoch, presenting key details related to the training and validation of the first layer. The training loss of the first layer decreases from epoch 1 to epoch 10, suggesting a continuous improvement in the first layer's ability to capture patterns within the training data. Additionally, the training accuracy of the first layer increases, indicating the first layer's proficiency in making more precise predictions on the training set.

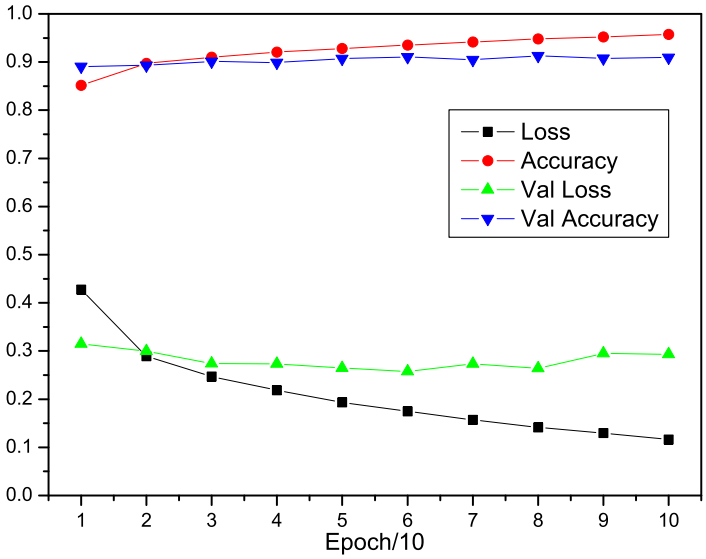

Fig. 5. Performance results for a BiLSTM (Bidirectional Long Short-Term Memory) model for the first layer.

Table 5. Performance results for a BiLSTM model in the second layer.

Epoch/10	Batch	Timer	Loss	Accuracy	Val Loss	Val Accuracy
1	3000/3000	26 s 9 ms/step	0.4559	0.8415	0.3332	0.8807
2	3000/3000	27 s 9 ms/step	0.3111	0.8896	0.3081	0.8897
3	3000/3000	26 s 9 ms/step	0.2726	0.9025	0.2911	0.8962
4	3000/3000	25 s 8 ms/step	0.2472	0.9117	0.2786	0.9013
5	3000/3000	18 s 6 ms/step	0.2238	0.9200	0.2642	0.9047
6	3000/3000	19 s 6 ms/step	0.2061	0.9252	0.3042	0.8921
7	3000/3000	23 s 8 ms/step	0.1925	0.9303	0.2662	0.9112
8	3000/3000	20 s 7 ms/step	0.1753	0.9365	0.2960	0.8993
9	3000/3000	18 s 6 ms/step	0.1648	0.9397	0.2758	0.9071
10	3000/3000	17 s 6 ms/step	0.1552	0.9439	0.2681	0.9116

Normalized mutual information, or NMI, is a statistic that assesses how similar two sets of data are to one another and is typically used in conjunction with cluster analysis. Two sets' mutual information ($I(X;Y)$) and entropies ($H(X)$ for true labels and $H(Y)$ for cluster assignments) are used,

$$\mathrm{NMI}(X,Y) = \frac{2 \cdot I(X;Y)}{H(X) + H(Y)}.$$

It is based on these factors. Within the range of 0 to 1, where 1 denotes complete agreement between two sets of labels, NMI normalizes the result. A statistical measure used to determine how similar two data clusterings are to one another is called the Adjusted Rand Index (ARI). Concordance between data points within the same clusters is taken into account while chance or randomization is also taken into account. The score that ARI produces falls between -1 and 1, where: when two clusterings have a score of 1, there is perfect agreement; when there is a score of 0, it means that two clusters are identical to what would

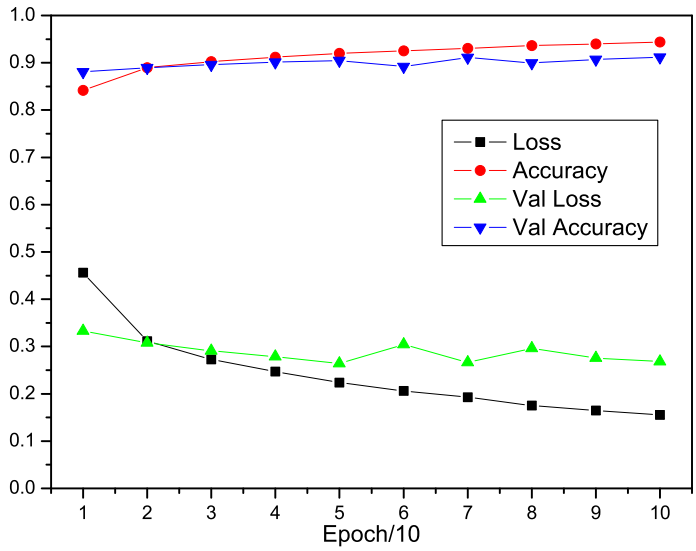

Fig. 6. Performance results for a BiLSTM (Bidirectional Long Short-Term Memory) model for the second layer.

be expected by chance. A significant difference between two clusterings is indicated by a score that is near -1. The following formula is used to calculate ARI mathematically:

$$\text{ARI} = \frac{\text{Rand Index} - \text{True Negative}}{\text{Max(Rand Index)} - E(\text{Rand Index})} - 1.$$

In this section, we subjected our method to a comprehensive evaluation by calculating precision (Accuracy) and recall (Recall) on the Fashion MNIST dataset, well-known for its complexity. Table 6 highlights the effectiveness of our approach.

Table 6. Demonstrating the efficacy of our approach.

N positions		N clusters		Performance metrics			
				Accuracy	Recall	TRUE Positive	TRUE Negative
64	19	92.90	92.31	300	420	30	25
64	20	94.54	94.78	400	500	30	22
64	21	94.71	95.13	450	500	30	23
64	22	94.60	94.66	444	520	30	25
32	19	93.85	94.65	425	400	30	24
32	20	93.67	94.05	380	420	30	24
32	21	94.66	94.93	450	420	25	24
32	22	93.77	93.01	333	420	25	25
16	19	94.96	94.66	444	500	25	25
16	20	94.87	94.44	425	500	25	25
16	21	94.52	94.73	450	500	30	25
16	22	94.71	95.13	450	500	30	23

Table 7. Comparative analysis of existing models on fashion MNIST dataset including our method.

Approaches to Clustering	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

In this section, we subjected our method to rigorous testing, conducting a comparative analysis on the Fashion MNIST dataset, renowned for its complexity. Table 7 summarizes the comparison between various existing mechanisms and our proposed model in terms of accuracy, ARI, and NMI. It is important to note that the Basic Fuzzy C-means model achieved an accuracy of 52.91%, while the K-means model reached 51.07%. However, other approaches such as IDEC, DEC, and DFCM outperformed these accuracy values, although our model maintained respectable performance. Regarding ARI as a comparison measure, it is noteworthy that Fuzzy C-means achieved an ARI of 36.44%, while K-means reached 36.39%. Our exist-

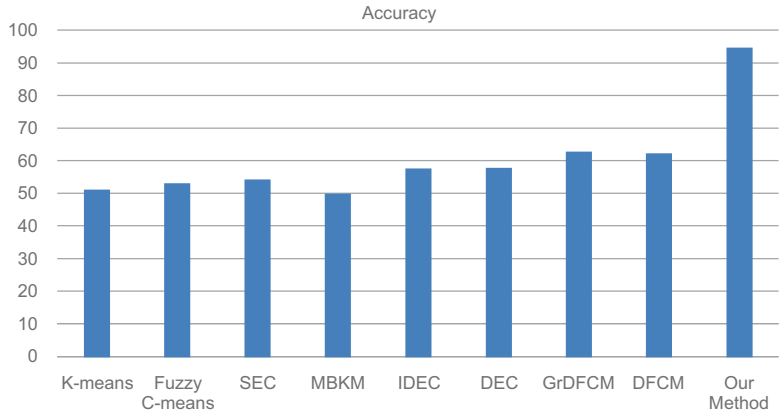

Fig. 7. Comparative evaluation of different preexisting models on the fashion MNIST dataset.

ing model demonstrated significant improvement, particularly with DFCM achieving 48.65% compared to the base model's 50.28%. Similarly, compared to other existing models, our enhanced FCM achieved a respectable ARI of 54.19%. Finally, considering NMI as a comparison metric, Fuzzy C-means reached 51.59%, K-means 51.64%, our existing model 66.09%, while our hybrid method attained 67.35%. Figure 1 below provides a comparison of different existing models on the Fashion MNIST dataset. In Figure 1 below, a comparison is conducted between several models commonly used with the Fashion MNIST dataset, including our new method.

6. Conclusion

During this study, we delved into the fascinating field of big data clustering and introduced an innovative approach that leverages the fusion of three potent techniques: FCM (Fuzzy C-Means), an optimized Encoder-Decoder Convolutional Neural Network (CNN), and a Bidirectional Long Short-Term Memory (BiLSTM) network. Our primary objective was to address the challenges inherent in clustering extensive and complex datasets by harnessing the strengths of these three powerful methods. We observed that integrating FCM allowed for an efficient initial clustering of data, significantly reducing the complexity of the problem. The optimized Encoder-Decoder CNN, trained to extract essential data features, contributed to improving the quality of cluster representations. The inclusion of BiLSTM, with its capacity to model contextual information from past and future sequences, enhanced the overall clustering performance by accounting for temporal dependencies within the data. Our experiments 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 BiLSTM not only yielded superior clustering results but also provided a more explicit and comprehensible representation of data clusters. This research has substantial implications across various domains, including finance, healthcare, marketing, and other sectors where the ability to extract meaningful insights from vast and complex datasets is of paramount importance. The fusion of FCM, the optimized Encoder-Decoder CNN, and BiLSTM offers a promising solution for data scientists and professionals operating in the era of big data analytics.

[1] Han J., Kamber M., Pei J. Mining: Concepts and Techniques. Morgan Kaufmann (2011).

[2] Chandola V., Banerjee A., Kumar V. Anomaly detection: A survey. ACM Computing Surveys. **41** (3), 1–58 (2009).

[3] Yeganejou M., Dick S. Classification via Deep Fuzzy c-Means Clustering. 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). 1–6 (2018).

[4] Rajesh T., Malar R. S. M. Rough set theory and feed-forward neural network-based brain tumor detection in magnetic resonance images. International Conference on Advanced Nanomaterials, Emerging Engineering Technologies. 240–244 (2013).

[5] Kuznietsov S., Chen Q. C., Wang X. L. Semisupervised deep learning for monocular depth map prediction. Preprint arXiv:1702.02706 (2017).

[6] Venkat R., Reddy K. S. Dealing with big data using fuzzy c-means (FCM) clustering and optimizing with gravitational search algorithm (GSA). 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). 465–467 (2019).

[7] Venkat R., Reddy K. S. Clustering of huge data with fuzzy c-means and applying gravitational search algorithm for optimization. International Journal of Recent Technology and Engineering. **8** (5), 3206–3209 (2020).

[8] Siebel N. T., Maybank S. J. Fusion of Multiple Tracking Algorithms for Robust People Tracking. Computer Vision – ECCV 2002. 373–387 (2002).

[9] Riaz S., Arshad A., Jiao L. C. Fuzzy rough C-mean based unsupervised CNN clustering for large-scale image data. Applied Sciences. **8** (10), 1869 (2018).

[10] Zhou S., Chen Q., Wang X. Fuzzy deep belief networks for semi-supervised sentiment classification. Neurocomputing. **131**, 312–322 (2014).

[11] Tarvainen A., Valpola H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. Advances in Neural Information Processing Systems. 1196–1205 (2014)

[12] Aqel F., Alaa K., Alaa N. E., Atounti M. Hybridization of Divide-and-Conquer technique and Neural Network algorithm for better contrast enhancement in medical images. Mathematical Modeling and Computing. **9** (4), 921–935 (2022).

[13] Zhang T., Lu H., Li S. Z. Learning semantic scene models by object classification and trajectory clustering. 2009 IEEE Conference on Computer Vision and Pattern Recognition. 1940–1947 (2009).

[14] 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. Mathematical Modeling and Computing. **10** (2), 338–350 (2023).

[15] Hochreiter S., Schmidhuber J. Long short-term memory. Neural Computation. **9** (8), 1735–1780 (1997).

[16] Little R., Rubin D. Statistical Analysis with Missing Data. Wiley (2019).

[17] Patcha A., Park J.-M. An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks. **51** (12), 3448–3470 (2007).

[18] Bezdek J. C. Fuzzy Algorithms for Perceptual Grouping. Computer Vision for Robots. Academic Press (1984).

[19] Bezdek A.,J. C. Fuzzy mathematics in pattern classification: A critique and some recommendations. Pattern Recognition Letters. **2** (3), 173–183, 3448–3470 (1984).

[20] LeCun Y., Bengio Yo., Hinton G. Deep learning. Nature. **521** (7553), 436–444 (2015).

[21] Hodge V. J., Austin J. A survey of outlier detection methodologies. Artificial Intelligence Review. **22** (2), 85–126 (2004).

[22] Batista G. E., Monard M. C. An analysis of four missing data treatment methods for supervised learning. Applied Artificial Intelligence. **17** (5–6), 519–533 (2003).

Кластеризація великих даних через поєднання FCM, оптимізованого кодера–декодера CNN та BiLSTM

Белхабіб Ф.[1], Ель Мутауакіл К.[1], Рбіхоу С.[2], Елафаар А.[1]

[1] *Університет Сіді Мохамед Бен Абделла, Факультет полідисциплінарних наук, Таза, Марокко*
[2] *Інженерія, системи та застосування, Сіді Мохамед Бен Абделла, ENSA, Фес, Марокко*

Кластеризація великих даних, як фундаментальний компонент обробки та аналізу масивних наборів даних, має вирішальне значення для вирішення складних проблем, пов'язаних із обробкою великих наборів даних. Основна мета кластеризації, яка входить у сферу методів неконтрольованого навчання, полягає в тому, щоб ефективно організувати значні набори даних в однорідні кластери без використання вже існуючих міток. Наш інноваційний підхід спрямований на оптимізацію цього процесу шляхом поєднання трьох методів: методології нечітких С-середніх (FCM), моделі оптимізованого кодера–декодера CNN і двонаправленої рекурентної нейронної мережі (BiLSTM). Це поєднання є стратегічним зближенням між контрольованою та неконтрольованою парадигмами. Впровадження BiLSTM має важливе значення, оскільки використовує його здатність послідовно обробляти дані з обох сторін за допомогою комірок LSTM. Цей двонаправлений підхід покращує розуміння послідовностей даних, що є важливою особливістю у контексті кластеризації великих даних. Водночас FCM отримує переваги від суттєвого вдосконалення завдяки впровадженню функції, яка обчислює відстань між центром кластера та екземпляром, тим самим підвищуючи точність кластеризації. Щоб оптимізувати продуктивність і скоротити час обчислень, запропонована методологія підтримує використання оптимізованої моделі CNN кодера-декодера. Ця вдосконалена архітектура сприяє більш ефективному вилученню функцій даних, тим самим підвищуючи внутрішню якість кластеризації. Строга оцінка запропонованого підходу базується на конкретних джерелах даних, а саме на MNIST моди. Критерії ефективності, такі як точність, скоригований індекс Ренда (ARI) та нормалізована взаємна інформація (NMI), переконливо свідчать про надзвичайні можливості запропонованої методології. У порівняльному аналізі запропонований підхід значно перевершує існуючі моделі, демонструючи свою ефективність і актуальність у складній оюласті кластеризації великих даних.

Ключові слова: *нечіткі С-середні (FCM); кластеризація; оптимізований кодер–декодер; кластеризація; двонаправлена рекурентна нейронна мережа (BiLSTM).*