



OPEN ACCESS

EDITED BY

Roshan Martis,
Global Academy of Technology, India

REVIEWED BY

Gokhan Altan,
Iskenderun Technical University, Türkiye
Muzafer Saracevic,
University of Novi Pazar, Serbia

*CORRESPONDENCE

Sunil Kumar Prabhakar
✉ sunilprabhakar22@gmail.com
Dong-Ok Won
✉ dongok.won@hallym.ac.kr

RECEIVED 01 February 2023

ACCEPTED 25 May 2023

PUBLISHED 21 June 2023

CITATION

Prabhakar SK and Won D-O (2023)
Performance comparison of bio-inspired and
learning-based clustering analysis with
machine learning techniques for classification
of EEG signals. *Front. Artif. Intell.* 6:1156269.
doi: 10.3389/frai.2023.1156269

COPYRIGHT

© 2023 Prabhakar and Won. This is an
open-access article distributed under the terms
of the [Creative Commons Attribution License
\(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction
in other forums is permitted, provided the
original author(s) and the copyright owner(s)
are credited and that the original publication in
this journal is cited, in accordance with
accepted academic practice. No use,
distribution or reproduction is permitted which
does not comply with these terms.

Performance comparison of bio-inspired and learning-based clustering analysis with machine learning techniques for classification of EEG signals

Sunil Kumar Prabhakar* and Dong-Ok Won*

Department of Artificial Intelligence Convergence, Hallym University, Chuncheon, Republic of Korea

A comprehensive analysis of an automated system for epileptic seizure detection is explained in this work. When a seizure occurs, it is quite difficult to differentiate the non-stationary patterns from the discharges occurring in a rhythmic manner. The proposed approach deals with it efficiently by clustering it initially for the sake of feature extraction by using six different techniques categorized under two different methods, e.g., bio-inspired clustering and learning-based clustering. Learning-based clustering includes K-means clusters and Fuzzy C-means (FCM) clusters, while bio-inspired clusters include Cuckoo search clusters, Dragonfly clusters, Firefly clusters, and Modified Firefly clusters. Clustered values were then classified with 10 suitable classifiers, and after the performance comparison analysis of the EEG time series, the results proved that this methodology flow achieved a good performance index and a high classification accuracy. A comparatively higher classification accuracy of 99.48% was achieved when Cuckoo search clusters were utilized with linear support vector machines (SVM) for epilepsy detection. A high classification accuracy of 98.96% was obtained when K-means clusters were classified with a naive Bayesian classifier (NBC) and Linear SVM, and similar results were obtained when FCM clusters were classified with Decision Trees yielding the same values. The comparatively lowest classification accuracy, at 75.5%, was obtained when Dragonfly clusters were classified with the K-nearest neighbor (KNN) classifier, and the second lowest classification accuracy of 75.75% was obtained when Firefly clusters were classified with NBC.

KEYWORDS

epilepsy, EEG, K-means clusters, fuzzy C-means clusters, Cuckoo search clusters, Firefly clusters, Dragonfly clusters

1. Introduction

Due to the excessive and abnormal electrical discharges in the brain cells, seizures are produced, which occur suddenly and are quite uncontrollable, making epilepsy a chronic neurological disorder (Gotman, 1982). To monitor the activities of the brain, electroencephalography (EEG) is widely used as it provides a huge amount of both physiological and pathological information, thereby proving its validity for an effective diagnosis (Kim et al., 2014; Jukic et al., 2020). The application of EEG plays a vital role in almost all areas of biomedical engineering, ranging from seizure classification (Rajaguru and Prabhakar, 2016), Alzheimer's disease diagnosis (Zhu et al., 2016), subject-dependent classification in Brain-Computer Interface (BCI) (Lee et al., 2015), classification of steady-state visual evoked potentials (Won et al., 2015), efficient analysis of Event-Related Potentials

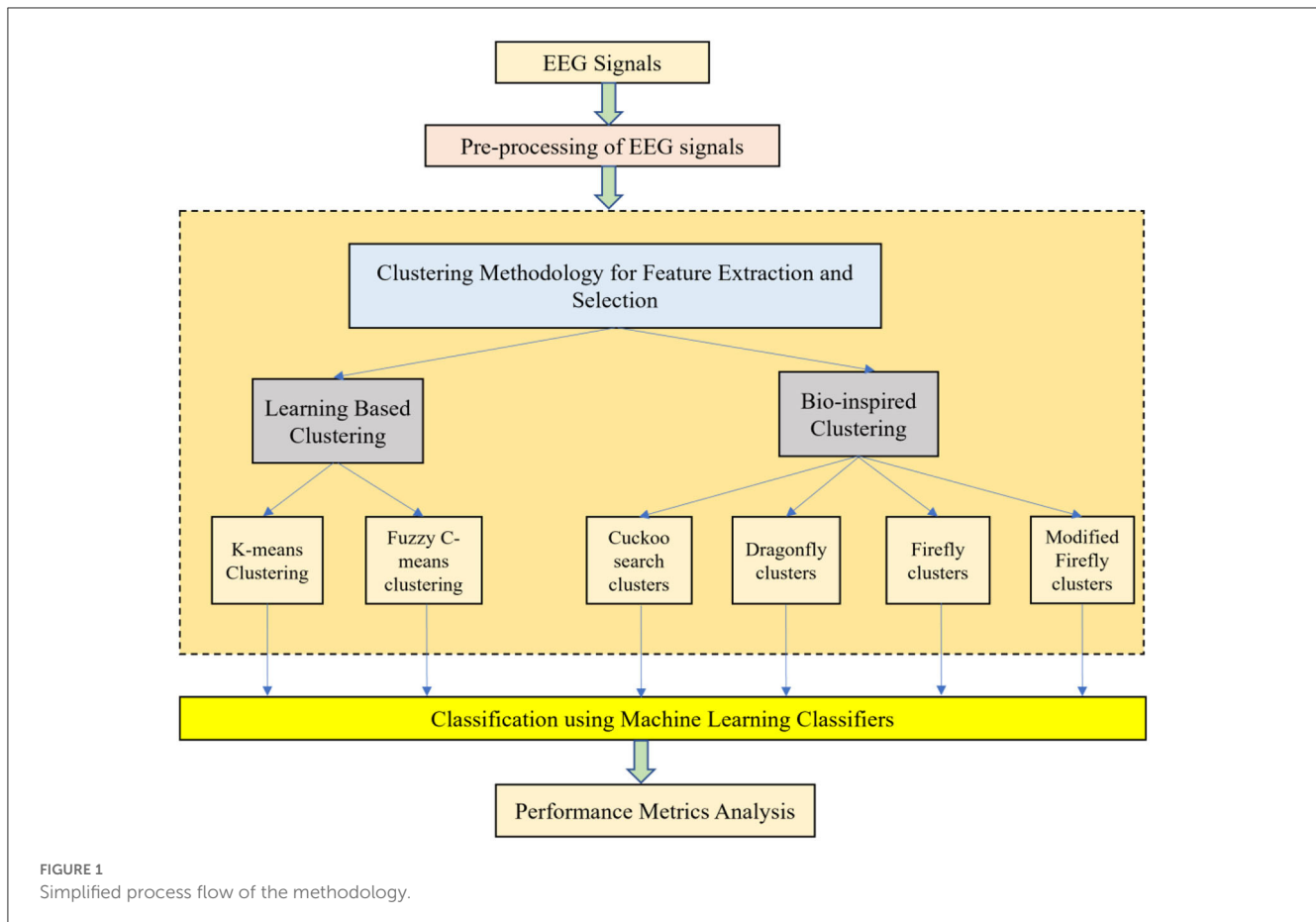
(ERP)-based BCI (Yeom et al., 2014), analysis of deep neural network modeling under an ambulatory environment (Kwak et al., 2017), dementia classification (Jeong, 2004), and iris recognition (Adamovic et al., 2020). Visual analysis of EEG recordings, even by expert neurologists, is very difficult and time-consuming. The over-reliance and subjective judgments of visual encephalographs may result in different diagnoses of the EEG segment (Kumar et al., 2014). As the EEG recordings have a very high margin of noise, separating seizures from artifacts is difficult as they have a similar time-frequency pattern. A plethora of machine-learning algorithms have been proposed for seizure detection and classification (Magosso et al., 2009). Generally, in the initial stages, preprocessing and feature extraction are performed and then classified with supervised or unsupervised classification models. To remove the major artifacts caused by eye blinking, slight body movement, and random muscle activity, preprocessing is performed, which helps in the extraction of the most discriminative features in the time domain (Bulthoff et al., 2003). Thus, studies on seizure detection using epileptic EEG databases are of great significance.

With the help of Artificial Neural Networks (ANN), a classification accuracy of 100% between normal and epileptic EEG signals has been achieved using Approximate Entropy (ApEn) by Srinivasan et al. (2007). For the automated detection of epileptic seizures from EEG signals, many researchers have proposed different feature extraction methodologies, as explained in (Tawfik et al., 2016). With the help of both Bayesian Linear Discriminant Analysis (BLDA) and lacunarity analysis, an algorithm for intracranial EEG seizure detection was proposed by Zhou et al. with a sensitivity of about 96.25% (Zhou et al., 2013). A good review of different entropy methods to differentiate normal, ictal, and interictal EEG signals with seven different classifiers was done by Acharya et al. (2012) where a high classification accuracy of 98.1% was reported. Compressive sensing techniques using Sample entropy, permutation entropy, and Hurst index for differentiating and detecting inter-ictal, pre-ictal, and ictal-based epileptic seizures with four different classifiers were done by Zeng et al. (2016), and a high classification accuracy of 76.7% was obtained. A 13-layer deep Convolutional Neural Network (CNN) was employed to detect normal, pre-ictal, and seizure classes in (Acharya et al., 2018) and reported accuracy, specificity, and sensitivity of 88.67, 90, and 95%, respectively. Automated seizure detection from EEG signals using a deep Convolutional Neural Networks (CNN)-based technique yielding a high classification accuracy of 97.5% was performed in (Zhou et al., 2018). SVM classifiers are one of the most versatile pattern recognition techniques used in epileptic seizure detection due to their most promising ability for generalization (Moghim and Corne, 2014). Constructive Genetic Programming (CGP) was used for classification after feature extraction by decomposing EEG signals using Empirical Mode Decomposition (EMD) by Bhardwaj et al. (2016), and for different validation schemes, the classification accuracy results were reported as ranging from 97 to 100%. ANNs were incorporated with multivariate EMD to classify ictal vs. non-ictal signals with the help of a standard SVM classifier (Riaz et al., 2016). Frequency domain features were reduced using non-linear dimension techniques, and the seizures were then classified using a K-nearest neighbor (KNN) classifier, which reported a classification accuracy of 98.40% (Rivero et al., 2011). Smoothed pseudo-Wigner-Ville distribution was incorporated with ANNs for

seizure classification, as proved in (Tzallas and Tsipouras, 2007), reporting an overall accuracy of 97.72 to 100%. Wavelet transforms also gave good results in the analysis of epilepsy detection studies, as shown in (Chen et al., 2017), where a perfect classification rate of 100% was obtained with all kinds of wavelet filters. Usage of techniques like tunable Q-factor wavelet transform with bootstrapping methodology gave a 100% classification accuracy, especially for A-E seizure classification problems (Haasan and Siuly, 2016), multi-wavelet-based approximate entropy (ApEn), with ANN again reporting a 100% classification accuracy (Kumar et al., 2014), and complex-valued neural network transform with K-fold cross-validation methodology also reporting results from 99 to 100% classification accuracy (Peker and Sen, 2016) are some of the predominantly used methodologies for epilepsy classification. A sparse autoencoder with a swarm-based deep learning technique (Prabhakar and Lee, 2022a), sparse representation-based robust hybrid feature extraction models with transfer and deep learning (Prabhakar and Lee, 2022b), sparse modeling with an ensemble and nature-inclined classification (Prabhakar and Lee, 2022c), and end-to-end deep neural network models (Zhao et al., 2020) were also done for epilepsy classification from EEG signals. Other important studies deal with the application of matrix determinants with machine learning classifiers (Raghu et al., 2019), local mean decomposition (LMD) with genetic algorithm-supported SVM (GA-SVM) (Zhang and Chen, 2017), discrete wavelet transform (DWT) with SVM (Chen et al., 2017), and linear least squares preprocessing with SVM (Zamir, 2016) for the classification of epilepsy from EEG signals. Other prominent studies include the usage of orthogonal wavelet analysis with linear SVM (Sharma et al., 2018), weighted complex networks with SVM (Diykh et al., 2017), Fourier transform with neural networks (Samiee et al., 2015), and binary pattern generation with Bayesian networks (Kaya et al., 2014) for the classification of epilepsy from EEG signals. In this study, the A-E classification problem of the Bonn data set was thoroughly studied through feature extraction, initially through clustering techniques, followed by categorization using appropriate classifiers for epilepsy from EEG signals. The organization of the work is as follows: Section 2 presents the materials and methods, followed by feature extraction through the clustering method in Section 3. The classification process is discussed in Section 4. It is followed by results and discussion in Section 5 and concludes in Section 6. The block diagram of the work is shown in Figure 1. The EEG signals taken from the Bonn data set were first preprocessed using the Independent Component Analysis (ICA) method and then clustered using two methodologies. Learning-based clustering includes k-means clustering and FCM clustering, while bio-inspired clustering involves cuckoo search clustering, dragonfly clustering, firefly clustering, and modified firefly clustering. The most important works in this field with interesting results in the last 6 years are shown in Table 1.

2. Materials and methods

The data used in this study is a publicly available data from the Epilepsy Research Center at Bonn University in Germany (Andrzejak et al., 2001). It consists of five subsets indicated as A-E. Each subset has 100 single-channel segments, and each



segment comprises an EEG signal of 23.6 s duration with a sampling frequency of ~ 173.61 Hz. The EEG signals present in this public data set were obtained from a 128-channel amplification system with an average reference electrode. In this study, the A-E classification type was studied in detail using clustering techniques and machine learning procedures. Further details about the EEG data set are given in Table 2. As the presence of abnormalities in the EEG signals is quite high due to its non-stationary and non-linearity properties, clustering is an effective methodology to extract useful features.

3. Clustering techniques

Clustering is basically the primary task of grouping or assimilating a set of specific objects in such a manner that the objects in the same group are very similar to each other. It is a vital task in exploratory data mining and is commonly used for all kinds of statistical data analysis in the fields of machine learning, pattern recognition, data analysis, and more (Zhao et al., 2020). Cluster analysis cannot be limited to one specific algorithm. Different algorithms can be used for cluster analyses, which differ significantly in their basic understanding of the constitution of the cluster. The formulation of a clustering problem is generally done as a multi-objective optimization problem, as most popular clusters include groups with dense areas of data space and small distances between the cluster members.

When working with large amounts of data, classification is generally difficult. Therefore, data reduction or data partitioning methods are quite useful for this purpose. After the basic pre-processing of EEG signals using Independent Component Analysis (ICA), data reduction is achieved by various methods like feature extraction, feature selection, and clustering techniques. Clustering is an unsupervised method used for data reduction. Selecting the number of clusters that represent the original data is important for the classification process. This will reduce the burden on the classifier and thereby increase classification accuracy. Clustering (unsupervised methods) can be defined as the process of organizing data into groups whose members are similar in some way. Without using training data, they normally work as an optimizer. To compensate for the lack of training data, this alternative method efficiently analyzes the partitioned data, thereby characterizing the properties of each class in a very distinctive way. The EEG signals are clustered as K-means clusters, FCM clusters, Cuckoo search clusters, Dragonfly clusters, Firefly clusters, and modified Firefly clusters.

3.1. K-means clustering

The problem of clustering a set of m objects $J = \{1, \dots, m\}$ objects into K clusters is initially considered in K-means clustering (Kanungo et al., 2004). For every subject $j \in J$, we have a set of

TABLE 1 A few famous and important epilepsy classification studies that have used the Bonn data set in the past.

Year	References	Classification type	Technique adopted
2022	Prabhakar and Lee (2022a)	A-E, B-E, C-E, D-E, AB-E, CD-E, AC-E, ACD-E, ABCD-E, BCD-E	Sparse autoencoder with swarm-based deep learning and reinforcement-based Q-learning
2022	Prabhakar and Lee (2022b)	B-E, AB-E, D-E, CD-E, A-C-E, AB-CD-E	Sparse representation-based robust hybrid feature extraction models with transfer and deep learning
2022	Prabhakar and Lee (2022c)	A-E, B-E, C-E, D-E, AB-E	Sparse modeling with ensemble- and nature-inclined classification
2020	Zhao et al. (2020)	A-E, B-E, C-E, D-E, AB-E, CD-E, ABCD-E	End-to-end deep neural network model
2019	Raghu et al. (2019)	A-E, B-E, C-E, D-E, AB-E, AC-E, CD-E, ACD-E, ABCD-E, A-C-E, AB-CD-E	SVM, KNN, Multi-layer perceptron (MLP), CNN
2018	Zhou et al. (2018)	A-E, A-C	CNN
2017	Zhang and Chen (2017)	A-E, D-E, ABCD-E, A-D-E, A-B-C-D-E	Local mean decomposition (LMD)-based hybrid features and GA-SVM
2017	Chen et al. (2017)	A-E, AB-E, CD-E, A-CDE, ACD-E, AB-CDE, ABCD-E	Discrete wavelet transform (DWT) and SVM
2016	Bhardwaj et al. (2016)	A-E, ACD-E, ABCD-E, AB-CD-E, A-B-C-D-E	EMD and CGP
2016	Zamir (2016)	A-E, B-E, ABCD-E, ACD-E	Linear least squares pre-processing and library SVM

TABLE 2 Bonn data set description details.

Subset	A	B	C	D	E
Subjects	Five healthy volunteers		Five epileptic patients		
Subject state	Eyes open	Eyes closed	Inter-ictal		Ictal
Recorded period	Normal		Seizure-free intervals		Seizure activity
Electrode types	Non-invasive surfaces		Intracranial		
Placement of electrodes	International 10–20 standard system		From hippocampal formation	Epileptogenic zone	All epileptic seizure areas

n features $[z_{ji} : i \in I]$, where z_{ji} explains the i^{th} features of the object j in a quantitative manner. Assuming $z_j = (z_{j1}, \dots, z_{jm})^T$ is the feature vector of the object j and $Z = (z_1, \dots, z_m)$ is the data set or the feature matrix, this clustering task is nothing but a reformulation as an optimization problem, thereby minimizing the clustering objective function as follows:

$$\min J(A, B) = \sum_{k=1}^K \sum_{j \in J} a_{jk} \|z_j - b_k\|^q \tag{1}$$

Under the following condition $\sum_{k=1}^K a_{jk} = 1, a_{jk} \in \{0, 1\}, \forall j \in J, k = 1, \dots, K$, where $q = 1, 2$. For $k = 1, \dots, K, b_k \in \mathfrak{R}^m$ is the k^{th} cluster prototype, and for any $j \in J, a_{jk}$ denotes whether the object j belongs to the k^{th} cluster.

To solve the clustering problems for $q = 1$ and $q = 2$, K-median and K-means are quite effective algorithms. The cluster prototype matrix $B = [b_1, \dots, b_K] \in \mathfrak{R}^{n \times K}$ and the membership matrix $A = [a_1, \dots, a_m] \in \mathfrak{R}^{K \times m}$, where $b_j = (b_{j1}, \dots, b_{jm})^T$ and $a_j = (a_{j1}, \dots, a_{jK})^T$. In an iterative manner, the clustering problem is solved by the Algorithm 1 as follows:

3.1.1. Algorithm 1: K-means clustering

Step 1: The iteration index $t = 0$ is set first, and then a random selection of K different objects is depicted as the initial cluster prototype and is represented as:

$$\{b_k^t : k = 1, \dots, K\} \tag{2}$$

Step 2: Assuming that $t = t + 1$, then the membership matrix A^t is updated by fixing the cluster prototype matrix B^{t-1} . For any $j \in J$, the random selection of $k^* \in \arg \min \{\|z_j - b_k^{t-1}\|_q : k = 1, \dots, K\}$ and set $a_{jk^*}^t = 1$ and for any $k \neq k^*$, set $a_{jk}^t = 0$

Step 3: The cluster prototype matrix B^t is updated by means of fixing the membership matrix A^t . When $q = 1$, for any $k = 1, \dots, K$ and $i \in I$, set b_{ki}^t as the median of the i^{th} feature values of these objects in the cluster k , when $q = 2$, for any $k = 1, \dots, K$, set b_k^t as the centroid of these objects in the cluster k , (i.e.,):

$$b_k^t = (1/\sum_{j \in J} a_{jk}) \sum_{j \in J} a_{jk} z_j \tag{3}$$

Step 4: For any $j \in J$ and $k = 1, \dots, K$, we have $a_{jk}^t = a_{jk}^{t-1}$, then the process is stopped and returned to A and B ; otherwise we return to step 2 of Algorithm 1.

3.2. Fuzzy C-means clustering

While dealing with fuzzy C-means clustering, $Y = \{y_1, y_2, \dots, y_n\}$ is a finite data set considered for our analysis where $y_i = (y_{i1}, y_{i2}, \dots, y_{ig})$ is a g -dimensional object and y_{ig} is the g^{th} property of the i^{th} object. $K = \{K_1, K_2, \dots, K_k\}$ denotes ' k ' clusters. $W = \{w_1, w_2, \dots, w_k\}$ represents k 1-dimensional cluster centroids, where $w_i = (w_{i1}, w_{i2}, \dots, w_{ig})$. $Z = (z_{iq})_{(n \times k)}$ is a fuzzy partition matrix and z_{iq} denotes the degree of membership of the i^{th} object in the q^{th} cluster where $\sum_{q=1}^k z_{iq} = 1, \forall i = 1, \dots, n$. The quadratic sum of weighted clusters to the cluster centroid from the samples in each cluster is the objective function and is denoted as:

$$H_m(Z, W) = \sum_{q=1}^k \sum_{i=1}^n z_{iq}^p d_{iq}^2 \quad (4)$$

where $d_{iq} = \|y_i - w_q\|$ depicts the Euclidean distance between the i^{th} object and the q^{th} cluster point. ($p \in [1, \infty]$) denotes a fuzziness index that helps in fuzzy membership control (Zhang et al., 2016). The resulting membership will be fuzzier if the value of p is higher. Based on the clustering idea, the appropriate fuzzy partition matrix Z and cluster centroid W are obtained (to minimize the objective function H_m). Depending on the Lagrange multiplier technique Z and W it is calculated as:

$$z_{iq} = \frac{1}{\sum_{h=1}^k (d_{iq}/d_{hq})^{2/(p-1)}} \quad (5)$$

$$w_q = \frac{\sum_{i=1}^n z_{iq}^p y_i}{\sum_{i=1}^n z_{iq}^p}$$

The FCM algorithm is implemented by means of minimizing the objective function H_p with the updates of Z and W . The steps are as follows as mentioned in Algorithm 2.

3.2.1. Algorithm 2: fuzzy C-means clustering

- 1) The initial value of the total number of clusters ' k ', fuzziness index p , threshold ξ , and maximum iterations I_{\max} is assigned.
- 2) The fuzzy partition $Z^{(0)}$ is randomly initialized based on the degree of membership.
- 3) The k cluster centroid $W^{(t)}$ is calculated at the t -step.
- 4) The objective function $H_p^{(t)}$ is calculated. If $|H_p^{(t)} - H_p^{(t-1)}| < \xi$ or $t > I_{\max}$, we stop or continue to the next step.
- 5) $Z^{(t+1)}$ is computed according to step 2 of this algorithm and then we proceed to step 3 of Algorithm 2.

Thus, it is quite a simple algorithm and its simplicity can be extended because of its quick convergence. Then, the conventional method to assess the optimal number of FCM clusters consists of the following steps.

Step 1: The search range $[k_{\min}, k_{\max}]$ must be fed or input; generally, $k_{\min} = 2$ and $k_{\max} = \lceil \sqrt{n} \rceil$

Step 2: For every integer $qn \in [k_{\min}, k_{\max}]$

Step 3: FCM is executed

Step 4: The clustering validity index is calculated

Step 5: The value of the clustering validity index is compared

Step 6: k_{opt} is obtained, which gives the clustering result.

3.3. Cuckoo clustering

Cuckoo search is a bio-inspired algorithm in which some species of cuckoo birds with obligate brood parasitism lay their eggs in the nests of some other bird species (Abd Elazim and Ali, 2016). Using the general rules, the Cuckoo search algorithm can be described as follows.

- 1) Each cuckoo lays one egg at a time and then it deposits it in a randomly selected nest.
- 2) The best nests with very good quality eggs will be passed on to the next generation.
- 3) The total number of available host nests is fixed. The host bird discovers the egg laid by a cuckoo with a probability of $P_a \in [0, 1]$. In such a case, the host bird can abandon the nest and build a new one, or easily get rid of the egg. The method is explained as follows in Algorithm 3.

Objective function: $f(y) = y = (y_1, y_2, \dots, y_m)$

Generate: an initial population of q host nests

While ($t < \text{Maximum Generation}$) or (Criteria to Stop)

Select a random cuckoo (say, j) and solution replacement by performing Levy flights

Fitness F_i evaluation

Randomly choose a nest among m (say, i)

if $F_j < F_i$ then

Replace i by the new solution

End if

A fraction p_a of worse nests are abandoned and new ones are built

The best ones are retained and kept

Ranking of nests is done to find the current best

The current best solution is passed to the next generation

End while

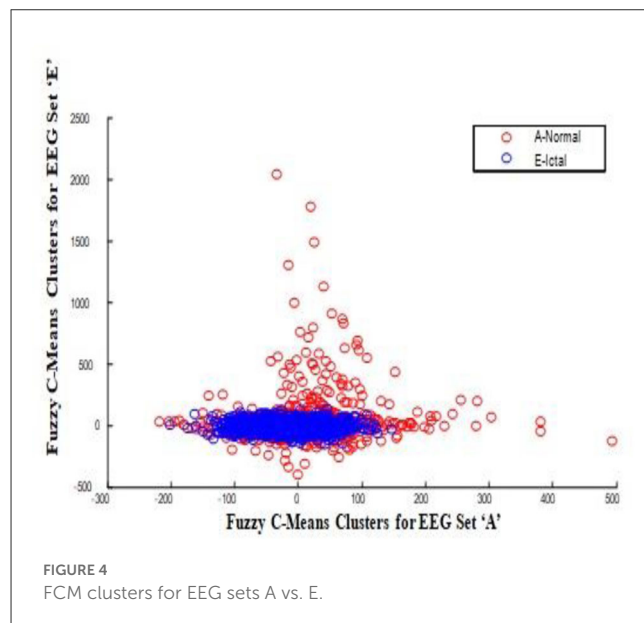
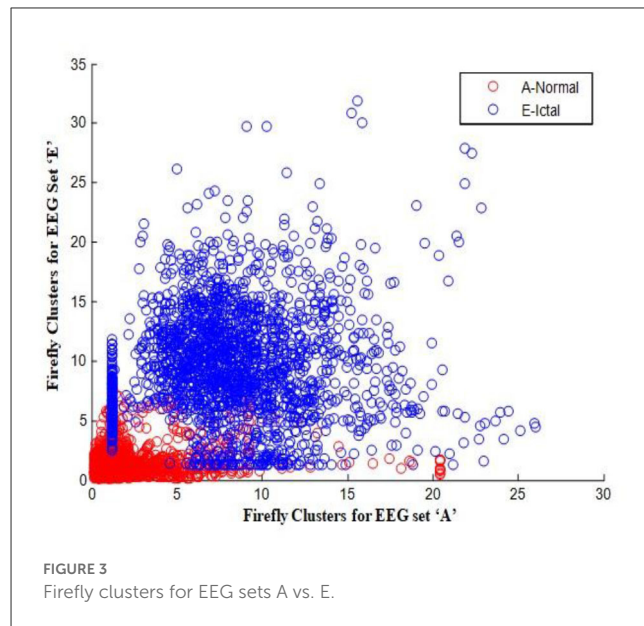
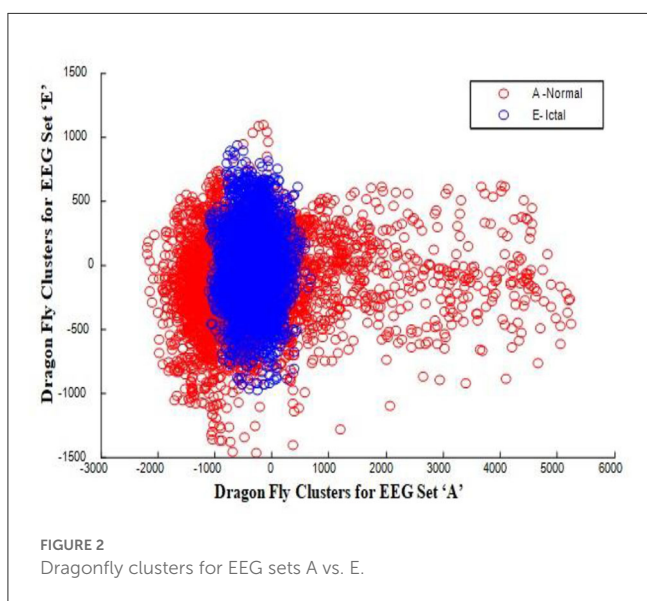
Algorithm 3. Cuckoo clustering.

3.4. Dragonfly clustering

Dragonflies are assumed to be little predators that naturally hunt other little bugs (Sree Ranjini and Murugan, 2017). Even before the marine bugs and little fishes originated, fairy dragonflies came into existence. The unusual swarming behavior is the most fascinating certainty about dragonflies. The two basic swarming methods are the same techniques of streamlining using metaheuristics, namely investigation and misuse. Some default steps like separation/split, alignment/join, cohesion, and attraction to a food source are common in this algorithm and are explained in Algorithm 4.

TABLE 3 Average entropy values.

Clusters	Approximate entropy		Shannon entropy		Sample entropy	
	A	E	A	E	A	E
EEG sets	A	E	A	E	A	E
K-means	1.37	1.19	1.46	1.24	5.429	6.15
FCM	1.42	1.15	1.58	1.27	5.23	6.34
Cuckoo	1.86	1.07	1.92	0.864	4.1774	6.82
Dragonfly	1.79	1.246	1.83	1.12	4.637	5.96
Firefly	1.65	1.058	1.75	1.034	5.407	6.12
Modified Firefly	1.39	1.217	1.86	1.17	4.8019	6.45
Average	1.58	1.155	1.733	1.16	4.95	6.32



3.4.1. Algorithm 4: dragonfly clustering

Step 1: Dragonfly initialization:

The population of dragonflies is initialized first and expressed as Q_j and its representation is done as:

$$H_{e1} = [H_{e1}, H_{e2}, H_{e3}, \dots, H_{e_n}] \tag{6}$$

The four important phases in Dragonfly initialization are as follows.

- i) Separation phase: To neglect the static collision of the people from various people in the neighborhood, the separation process is utilized and expressed as:

$$Sep_{je} = \sum_{l=1}^M H_{e_j} - H_{e_l} \tag{7}$$

where Sep_{e_j} denotes the separation of the j^{th} individual, H_e denotes the present position of the individual, and H_{e_l}

is the position of the l^{th} individual, and M denotes the total number of individuals in an adjacent nearer area present in the search space.

- ii) Alignment phase: This happens depending on the velocity matching of each individual to other respective individuals in the neighborhood. It is calculated as:

$$Ali_j = \frac{\sum_{l=1}^M E_l}{M} \tag{8}$$

where Ali_j denotes the alignment of the j^{th} neighboring individual, E_l is the velocity of the l^{th} individual, and M denotes the total number of neighboring individuals in the entire search space.

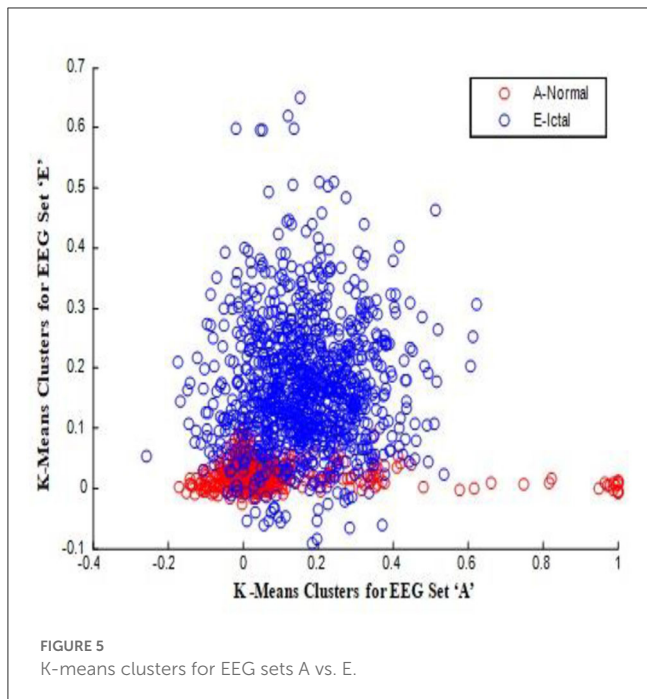


FIGURE 5 K-means clusters for EEG sets A vs. E.

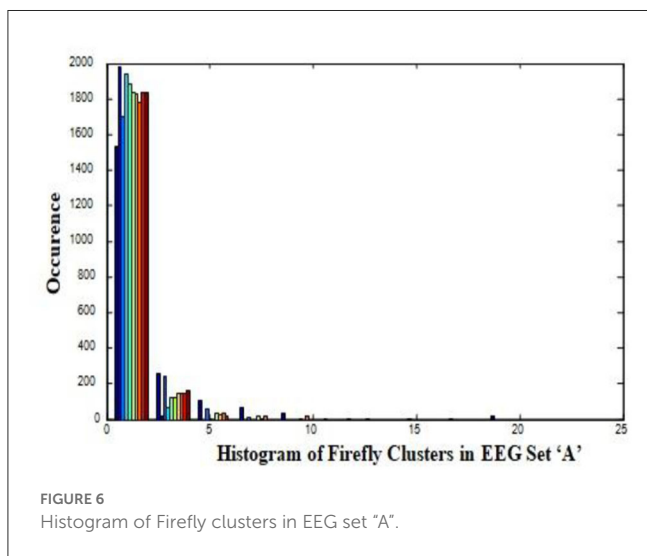


FIGURE 6 Histogram of Firefly clusters in EEG set "A".

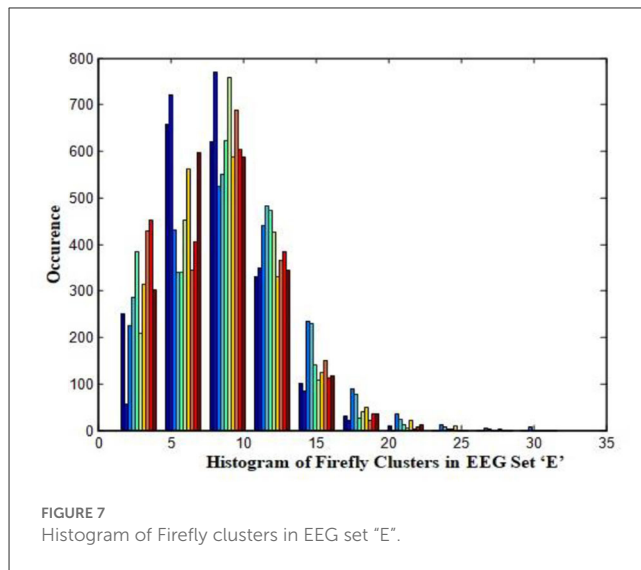


FIGURE 7 Histogram of Firefly clusters in EEG set "E".

where H_e^- represents the position of the enemy and H_e^+ denotes the position of the food source.

Step 2: Update process

$$\Delta Feature_set = (v_s Sep_j + v_a Ali_j + v_c Co_j + v_h Food_j + v_e Ene_j) + v \Delta Feature_t \tag{12}$$

Two vectors, ($\Delta Feature$) and position ($Feature$) vector are considered to update the position of artificial dragonflies in the search space, v_s, v_a , where it represents the weights of the technique, such as detachment, arrangement, and union. From the result of the calculation, the advanced positions are obtained, and thus, after checking the score estimates, additional information can be extracted.

3.5. Firefly clustering

It is a bio-inspired metaheuristic algorithm predominantly used for solving optimization problems. It is based on and observed in the flashing behavior of fireflies at night (Yang, 2010). The three main rules used in the construction of this algorithm are that, first, fireflies are unisexual in nature. Second, the brightness of the firefly is understood from the objective function. Third, there is a direct proportional relationship between attractiveness and brightness. A firefly generally moves toward the brighter one, and if there is no brighter one, then there is random movement. It is a well-known fact that there is an inversely proportional relationship between the intensity of light and the square of the distance, for instance, q from the source. Moreover, when light passes through a particular medium with a light absorption coefficient of λ , the intensity of light J varies with distance as follows:

$$J(q) = J_0 e^{-\lambda q} \tag{13}$$

iii) Cohesion phase: The dependency of the individuals to move toward the center of the mass in the neighborhood is called cohesion. The cohesion Co_j for the j^{th} individual is calculated as:

$$Co_j = \frac{\sum_{l=1}^M H_{e_l}}{M} - H_e \tag{9}$$

iv) Progressing toward a food source phase: Among the dragonflies, this is the attractive phase toward a particular food source, and the outward distance of each dragonfly is expressed as:

$$Food_j = H_e^+ - H_e^- \tag{10}$$

$$Ene_j = Q^- + D \tag{11}$$

where J_0 denotes the intensity at the starting point. It can be combined as:

$$J(q) = J_0 e^{-\lambda q^2} \quad (14)$$

Computing $e^{-\lambda q^2}$ is slightly difficult, but computing $\frac{1}{1+\lambda q^2}$ is comparatively easier. Therefore, the calculation of the intensity is done as follows:

$$J(q) = \frac{J_0}{1 + \lambda q^2} \quad (15)$$

The definition of firefly attractiveness is expressed as follows:

$$B(q) = \frac{B_0}{1 + \lambda q^2} \quad (16)$$

where B_0 denotes the attractiveness of $q = 0$. If a firefly situated at $y' = (y'_1, y'_2, \dots, y'_n)$ is brighter than another firefly situated at $y = (y_1, y_2, \dots, y_n)$, then the firefly located at y moves toward y' . The update of the firefly located at y is expressed as:

$$y := y + A_0 e^{-\lambda q^2} (y' - y) + \alpha \varepsilon \quad (17)$$

$\alpha \varepsilon$ denotes the randomization term with α represented as the randomization parameter denoted within the range of 0 to 1, i.e., $0 \leq \alpha \leq 1$, and ε denotes the random vector numbers. The second term is because of the attractiveness of y toward y' . For the sake of practicality, A_0 can be assumed to be 1; $A_0 = 1$.

The summary of the algorithm is given in Algorithm 5 as follows.

3.5.1. Algorithm 5: firefly clustering

- 1) A random solution set $\{y_1, y_2, \dots, y_k\}$ is generated.
- 2) The intensity for each solution member is computed as $\{J_1, J_2, \dots, J_k\}$.
- 3) Each firefly ' j ' moves toward other bright fireflies. If there are no bright fireflies, then each firefly moves in a random manner.
- 4) The solution set is updated.
- 5) If the termination criteria are fulfilled, then we terminate, otherwise, we go to step 2 of this algorithm.

3.6. Modified firefly clustering

The firefly with the current best global solution is the brightest. The random movement of this brightest firefly may decrease its brightness depending on the direction. Therefore, in that particular direction, it leads to the performance degradation of this algorithm. If the brightest firefly only moves in a direction that increases its brightness, then the algorithm can perform better. The modification made here is implemented in (Tilahun and Ong, 2012), and the idea is incorporated into our work as follows. To assess the direction of the brightest

TABLE 4 Performance of MSE with different numbers of iterations for KNN classifiers at different K values.

Number of iterations	MSE at different K values	
	K = 2	K = 4
50	0.000475	0.000001
100	0.0004	4.00E-03
150	0.000204	7.84E-04
200	4.49E-05	0.002025
225	3.48E-05	3.60E-04
250	0.000025	0.000122
300	2.3E-05	0.0001
350	2.12E-05	0.00059
375	1.68E-05	0.000128
400	1.09E-05	0.000292
500	4.41E-06	6.25E-04
600	4E-06	0.00049
800	2.25E-06	1.00E-04
1000	1.96E-06	0.000202
1200	0.000001	5.76E-04
1400	3.6E-07	0.000302

firefly, ' s ' unit vectors are randomly generated, e.g., v_1, v_2, \dots, v_s . Among the randomly generated ' s ' directions, a direction V is chosen in such a way that the brightness of the brightest firefly always increases when the firefly is in that particular direction. Therefore, the movement of the brightest firefly can be expressed as; $y := y + \alpha V$, where α denotes the random step length.

The brightest firefly will stay in its current position if such a direction does not exist in any of the randomly generated solutions. Moreover, rather than assuming $B_0^j = 1$ for each firefly j , it is a good idea to assign some attractiveness that is dependent on the firefly intensity, which is in turn highly dependent on the objective function. One way to do this is to assign the ratio of the intensities of the fireflies. If a firefly j situated at y' is brighter than a firefly i located at y , then the firefly located at y moves toward the firefly j , and B_0 is expressed as:

$$B_0 = \frac{J'_0}{J_0} \quad (18)$$

where J'_0 is the intensity at $q = 0$ for firefly j , J_0 is the intensity at $q = 0$ for firefly i and $J_0 \neq 0$. For convenience, we reject the singularity case when $J_0 = 0$, B_0 is expressed as $e^{J'_0 - J_0}$. If we consider $B_0 = J'_0$ and when the intensity is large, then the movement of the firefly i toward j is very long. However, B_0 can be adjusted depending on the solution space. Thus, the solution space here should be directly proportional to the intensity at the source J'_0 .

The main reason for the clustering in this work is the data reduction through the process of optimization, as stated in the

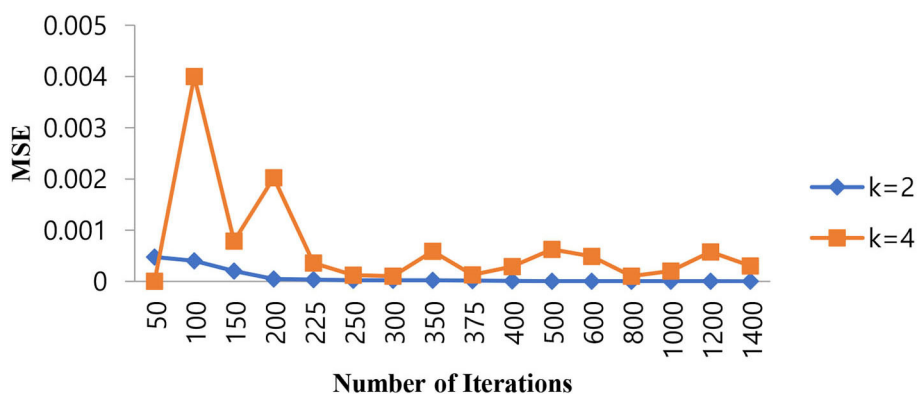


FIGURE 8 Performance of the MSE in the number of iterations for the KNN classifier at different K values.

TABLE 5 Performance of the MSE in the number of iterations for SVM-RBF at different gamma values.

Number of iterations	MSE at different gamma values								
	G = 0.2	G = 0.4	G = 0.6	G = 0.8	G = 1.0	G = 1.2	G = 1.5	G = 2.0	G = 2.2
50	0.00012321	0.00014884	0.00020164	7.744E-05	5.184E-05	0.000001	0.000328	1.37E-05	0.000511
100	5.476E-05	3.6E-05	9.025E-05	0.00017689	1.44E-06	0.000094249	0.000161	2.3E-05	2.81E-05
150	0.00012996	0.00011236	1.1664E-05	0.00018496	0.000051529	2.401E-05	1.96E-06	5.18E-05	8.10E-05
200	1.849E-05	7.921E-05	7.921E-05	1.849E-05	0.00002304	1.68921E-05	0.000801	2.92E-05	9.00E-05
225	7.569E-05	6.4E-05	0.000169	5.184E-05	0.00023409	0.00097969	0.002275	1.40E-07	6.40E-05
250	6.084E-05	0.000144	0.00019044	4.489E-05	0.00206116	0.000225	1.02E-05	4.00E-08	6.40E-06
300	5.929E-05	6.4E-05	0.00031684	0.00012996	0.00178929	6.724E-05	7.57E-05	3.51E-08	6.40E-06
350	0.0001	8.649E-05	0.000196	0.00011881	0.00147456	8.1E-05	0.000174	1.00E-08	7.92E-05
375	2.916E-05	0.00016129	4.096E-05	4.9E-05	1.024E-05	0.00042025	1.94E-05	5.00E-09	3.03E-05
400	4.9E-05	0.00022201	3.721E-05	0.0000169	1.66E-05	2.916E-05	0.000154	4.10E-09	8.10E-05
500	6.4E-05	0.00011025	0.000018496	0.00010609	0.000063504	2.304E-05	5.93E-05	4.00E-09	3.60E-05
600	0.00020736	6.889E-05	8.464E-05	2.704E-05	2.52004E-06	0.00018769	5.78E-05	2.70E-09	4.90E-05
800	0.00014161	1.0609E-05	6.724E-05	2.209E-05	0.00002025	0.00040401	0.000243	1.90E-09	8.41E-05
1000	6.561E-05	2.6569E-05	0.0000121	0.00022801	0.00000121	0.00037636	6.24E-05	1.60E-09	1.96E-05
1200	5.776E-05	0.000053361	2.304E-05	1.764E-05	1.7956E-06	3.31776E-05	0.000973	1.40E-09	6.40E-05
1400	7.14025E-05	2.43049E-05	4.489E-05	2.07E-05	5.9049E-06	1.47456E-05	5.76E-06	1.03E-09	6.64E-05

above algorithms. The clusters are validated by two parameters, compactness and separability. In this study, the compactness of the clusters is analyzed by the scatter plots among the classes of EEG data. When it comes to separability, these particular types of indices are used to differentiate between two clusters. The distance between the two cluster centroids is a commonly used measure of separability. This measure is easy to compute and can detect even hyperspherical clusters. Three types of entropy, e.g., Approximate Entropy, Shannon Entropy, and Sample Entropy were used to identify the separability of the clusters among the classes of the EEG data set (A-E), which is shown in Table 3.

In this work, a total of $(4,097 \times 100)$ A-E EEG data are clustered based on the cluster center features as $(4,097 \times 10)$ A-E EEG data. Therefore, an effective data reduction of ten times is achieved. The reduced features must be analyzed for the presence of non-linearity and dynamic states of the EEG signal. It is a two-step process consisting of scattering with histogram plots and entropy extraction. Scatter plots and histograms can easily help identify the presence of non-linearity in the features. Figures 2, 3 depict the scatter plot of Dragonfly clusters and Firefly clusters for EEG set A vs. E. The shape of the cluster in the figures obtained shows the presence of non-linearity in the features. Figures 4, 5 depict the scatter plot of FCM clusters and K-means clusters for EEG

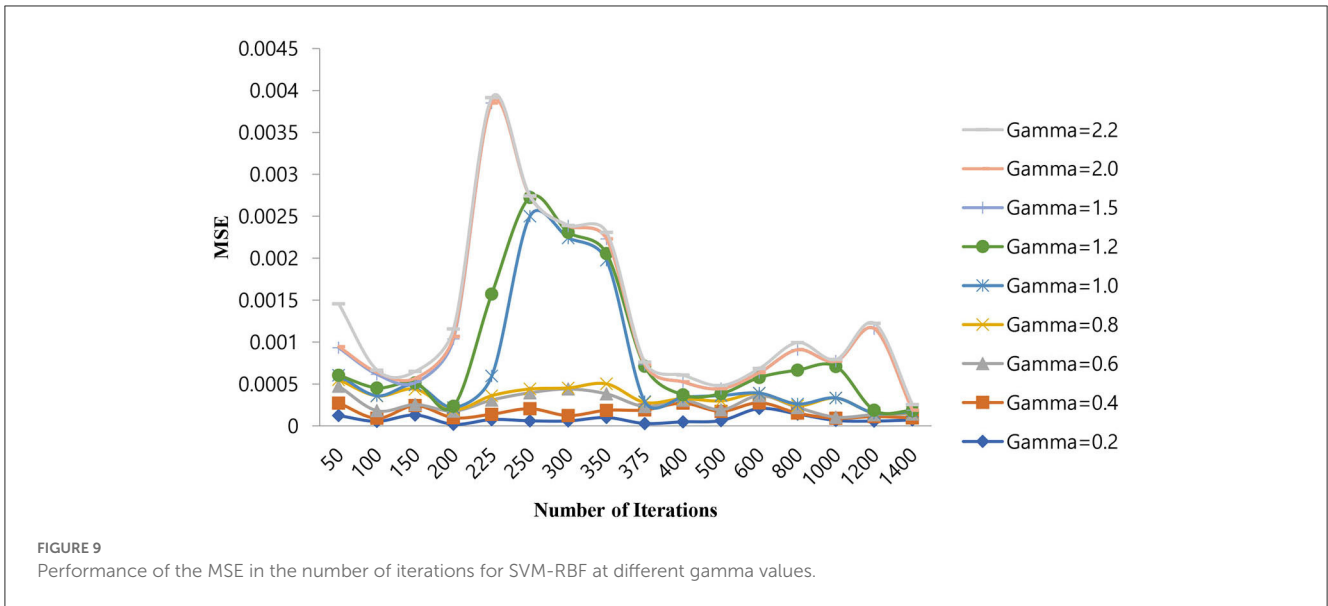


TABLE 6 Performance of the MSE in terms of the number of iterations for the Polynomial SVM classifier at different polynomial orders.

Number of iterations	MSE at different polynomial orders					
	$P = 2$	$P = 3$	$P = 4$	$P = 5$	$P = 6$	$P = 8$
50	4.90E-06	0.000001	0.000053824	8.649E-05	0.00073984	0.00016129
100	1.44E-06	4.225E-05	0.000048841	9.409E-05	0.00061009	0.000144
150	4.9E-07	5.184E-05	0.00073984	1.00E-05	0.0007569	0.000018769
200	6.4E-07	5.184E-05	0.00003984	1.1664E-05	0.000784	8.70489E-05
225	3.6E-07	5.184E-05	4.225E-05	0.000073984	0.00047089	1.369E-05
250	2.20E-07	1.089E-05	3.96E-05	7.056E-05	0.00055696	1.024E-05
300	1.10E-07	0.000073984	6.40E-06	1.96E-05	0.00022201	2.6244E-06
350	1.51E-07	1.77E-05	7.744E-06	5.184E-05	0.00036481	0.001681
375	1.40E-07	5.776E-05	7.84E-06	2.704E-05	0.00010201	5.76E-06
400	1.90E-07	0.000026244	5.184E-06	4.90E-06	2.03E-04	2.63169E-05
500	1.18E-07	0.000073984	1.1236E-06	1.69E-06	6.5536E-05	7.29E-05
600	1.40E-07	5.184E-05	0.00000196	5.1984E-06	0.00018225	6.67489E-05
800	9E-08	1.444E-05	4.9729E-06	5.184E-05	0.00013456	1.12225E-05
1000	6.40E-08	8.836E-05	7.3984E-06	2.809E-05	0.00058564	2.304E-05
1200	3.60E-08	4.00E-06	2.401E-05	0.00000025	8.10E-05	0.000022801
1400	1.60E-08	2.61E-06	5.184E-05	4.00E-06	2.2801E-06	2.01601E-05

A vs. E. Figure 6 shows the histogram of Firefly clusters in EEG set A. This histogram plot is a right-skewed one, which again establishes the presence of non-linearity in the feature. Figure 7 illustrates the histogram of Firefly clusters in EEG set E. Figure 7 exhibits the flatness in the histogram, which is different from the histogram of set A. The profound technique of extracting three types of entropy, e.g., Approximate Entropy (Pincus, 1991), Shannon Entropy (Shannon, 1948), and Sample Entropy (Richman and Moorman, 2000) for the clustered features of EEG data shows the intactness of residuals and variation as in the original EEG data.

Shannon Entropy extracts the uncertainty associated with a more specific event and its outlier characteristics. Approximate Entropy shows the presence of non-linear and complicated features in irregular data. Sample Entropy is obtained by the removal of self-matches in the Approximate Entropy. Sample Entropy is independent of data length and depicts consistency. The average entropies, which in turn represent the lower and upper bounds of the clusters, are presented in Table 3. The average value of entropy among the clusters in Table 3 indicates the availability of a distinct classification in the A-E EEG data set.

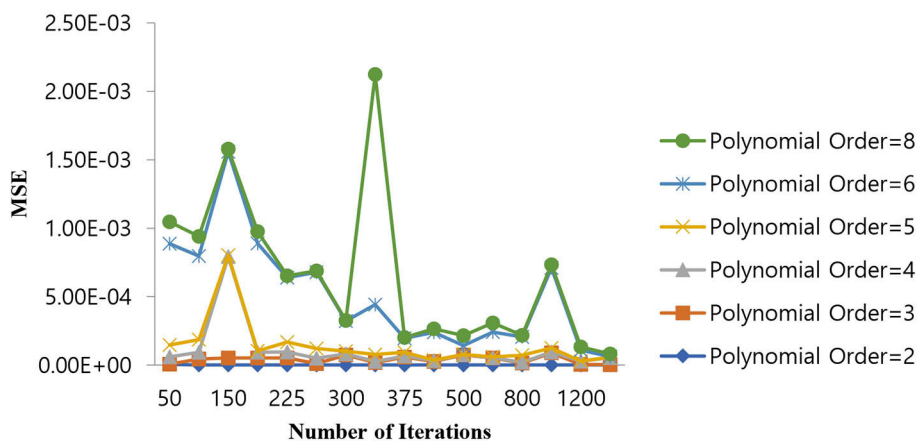


FIGURE 10 Performance of MSE in the number of iterations for the Polynomial SVM classifier at different orders.

TABLE 7 Performance of the MSE in the number of iterations for the Linear SVM classifier for the SGD algorithm.

Number of iterations	MSE
50	2.30E-05
100	0.00061504
150	1.024E-05
200	0.000225
225	7.84E-05
250	3.31E-05
300	3.24E-05
350	6.40E-05
375	1.69E-06
400	4.97E-05
500	1.51E-04
600	2.34E-04
800	0.00021025
1,000	1.23E-06
1,200	1.32E-06
1,400	1.24E-06
1,600	2.3104E-07
2,000	1.69E-08

4. Classification of the clustered values

The clustered values are then classified with the help of 10 different classifiers such as ANN, KNN, Incremental LDA (I-LDA), NBC, Quadratic Discriminant Analysis (QDA), Decision Trees, and Random Forest (RF), as explained in the following subsections.

4.1. ANN classifier

Motivated and inspired by the basic functional ideas of biological neural networks, ANN is a simple learning-based classifier (Sezer et al., 2012). The posterior probabilities can be easily estimated by ANN, which in turn helps to establish classification rules and then perform the statistical analysis. There are various ANN parameters here, the ANN configuration uses training cycles = 800, momentum decay = 0.5, and learning rate = 0.3. The ANN architecture used in this work is (128-32-2). The learning algorithm used here is the Levenberg-Marquardt (LM) algorithm with a sigmoidal function.

4.2. KNN classifier

A given test sample is compared with the training samples that are similar, where the k parameter is a small positive and odd integer value. There are two important steps in this algorithm (Song et al., 2007). First, the k training samples that are closest to the invisible sample are found. Second, the common classification for the k samples is taken, and then the average of the values of its KNN is found out in the regression stage. Using a distance metric called normalized Euclidean distance, it can be defined. Between the two points $Z_1 = (z_{11}, z_{12}, \dots, z_{1m})$ and $Z_2 = (z_{21}, z_{22}, \dots, z_{2m})$, the distance is analyzed as:

$$dist(Z_1, Z_2) = \sqrt{\sum_{i=1}^m (z_{1i} - z_{2i})^2} \tag{19}$$

In this work, the KNN uses a value of $k = 2$, and the measure types have been selected as mixed measures, thus allowing the mixed Euclidean distance as the best option.

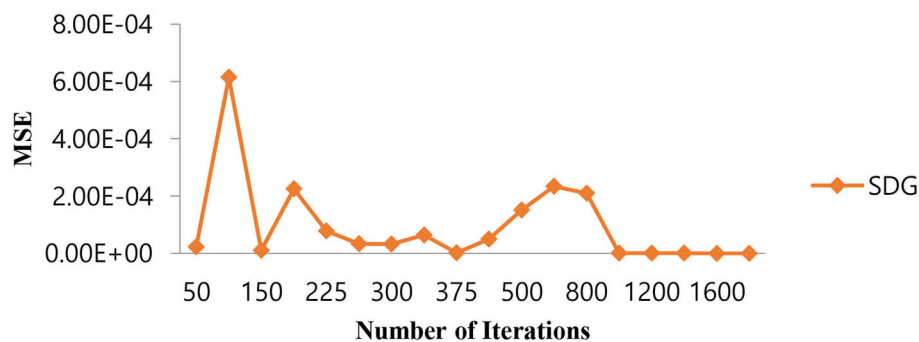


FIGURE 11 Performance of the MSE in the number of iterations for the Linear SVM classifier for the SGD algorithm.

4.3. I-LDA classifier

It is a kind of LDA classification technique that can easily update and classify the cluster features through the simple observation of new samples (Chu et al., 2015). The I-LDA classifier was trained with pre-ictal and ictal feature vectors. In the training set, the sampling strategy was used to randomly balance the number of pre-ictal and ictal segments. During the testing phase, the trained classifier was tasked with categorizing the incoming epoch as a pre-ictal or an ictal state. For the pre-ictal state, the binary result of the classifier was zero, and for the ictal state, it was one. To smooth the results, a median filter was used. The prediction alarm would be raised if $\Sigma T_j = \alpha_j$, where T_j is consecutive "1 s", with a progressive window of 1 s and α_j is a patient dependent threshold. The value of α_j is obtained from the training data set. If the range falls within the prediction zone, then the alarm is positive; otherwise, it is considered as a false alarm.

4.4. NB classifier

This simple statistical probabilistic classifier was developed based on the application of Bayes' theorem (Islam et al., 2007). The naive Bayesian technique assumes that the calculation of the NBC is easier than the exponential complexity. Here, it proves its efficacy by analyzing the fact that certain features of a class are irrelevant to other features. Each feature is considered independently to calculate its respective probability, which helps to calculate the probability of a certain class, which will be the classification outcome.

4.5. SVM classifiers

Based on the principles of structural risk minimization and statistical learning theory, SVM classification was developed (Swami et al., 2014). The main idea of the SVM is to map the input data into a higher-dimensional space. Once mapped, an optimal separating hyperplane between the two data classes of the transformed space is determined. SVMs can easily map the inseparable data into a high-dimensional space by means of

constructing a linear kernel function. When the data are not linearly separable, which is the case with nonlinear classifier models, SVMs can clearly provide a better fit of the hyperplane to the input data set. Originally designed as a two-class classifier, its application was later extended to multiclass classification. A set of pairwise classifiers is employed based on one-against-one decomposition. For the binary SVM classifier, the decision function is expressed as:

$$f(y) = \text{sgn} \left(\sum_{j=1}^s z_j \alpha_j k(y_j, y) + w \right); \quad 0 < \alpha_j < Q \quad (20)$$

where sgn denotes a sigma function; $k(y_j, y)$ denotes a kernel function; and w denotes the bias of the training samples. Several kernel functions, such as the Linear kernel, Polynomial kernel, and Radial Basis Function (RBF) kernel, are used in this work. Between the model complexity and the training error, to control the trade-off, the regularization parameter Q is utilized and is calculated as follows:

$$Q = \frac{N}{\sum_{j=1}^N K(y_j, y)} \quad (21)$$

where N denotes the size of the training set.

By utilizing a set of decision functions f_{kh} , a binary or multiclass classification problem arises. With the help of the following formula, the class decision is obtained as follows:

$$f_k(y) = \sum_{j=1}^n \text{sgn}(f_{kh}(y)) \quad (22)$$

where kh denotes each pair of classes extracted from separate target classes, n denotes the number of separate target classes. Initially, a label is assigned to the class as follows:

$$\arg \max f_k(y), \quad (k = 1, 2, \dots, n) \quad (23)$$

TABLE 8 Consolidated result analysis of K-means clusters with the different classifiers.

Classifiers	Sensitivity (%)	Specificity (%)	Accuracy (%)
ANN	56.57656	100	78.28828
KNN	71.875	100	85.9375
I-LDA	61.46	100	80.73
NBC	100	97.92	98.96
Linear SVM	97.92	100	98.96
Polynomial SVM	94.79	100	97.395
SVM-RBF	92.71	100	96.355
QDA	91.41	100	95.705
Decision Trees	69.795	100	84.8975
RF	100	80.225	90.1125

TABLE 9 Consolidated result analysis of FCM clusters with the different classifiers.

Classifiers	Sensitivity (%)	Specificity (%)	Accuracy (%)
ANN	64.19	100	82.095
KNN	100	89.59	94.795
I-LDA	100	85.42	92.71
NBC	100	89.0675	94.53375
Linear SVM	90.11	100	95.055
Polynomial SVM	95.31	100	97.655
SVM-RBF	100	77.625	88.8125
QDA	100	96.875	98.4375
Decision Trees	97.92	100	98.96
RF	73.4375	100	86.71875

The conversion from the n -class classification problem to the $(n(n-1)/2)$ two-class problem is generally done by pairwise classification, which helps to cover all the pairs of classes.

4.6. QDA classifier

Among statistics, signal processing, and pattern recognition, it is widely used to seek a quadratic combination of features that are responsible for analyzing an example into two or more types of categorizations (Heijden et al., 2005). The process of discriminating quadratic multiplication factors is used for both classification and dimensionality reduction, but in our work it has been used only for classification.

4.7. Decision tree classifier

Decision Trees utilize the top-down construction method to recursively split the data set into smaller subsets (Wang et al.,

TABLE 10 Consolidated result analysis of Cuckoo search clusters with the different classifiers.

Classifiers	Sensitivity (%)	Specificity (%)	Accuracy (%)
ANN	87.5	100	93.75
K-NN	82.94813	100	91.47406
I-LDA	75	100	87.5
NBC	100	96.3525	98.17625
Linear SVM	98.96	100	99.48
Polynomial SVM	91.67	100	95.835
SVM-RBF	67.969	100	83.9845
QDA	70.05625	100	85.02813
Decision Trees	65.232	100	82.616
RF	100	90.11	95.055

TABLE 11 Consolidated result analysis of Dragonfly clusters with the different classifiers.

Classifiers	Sensitivity (%)	Specificity (%)	Accuracy (%)
ANN	80.7375	100	90.36875
KNN	51	100	75.5
I-LDA	65.884	100	82.942
NBC	62.5	100	81.25
Linear SVM	80.225	100	90.1125
Polynomial SVM	100	81.25	90.625
SVM-RBF	94.79	100	97.395
QDA	93.75	100	96.875
Decision Trees	91.15	100	95.575
RF	77.625	100	88.8125

2012). Utilizing the concept of information entropy, a decision tree is built by the classifier for the data set. By splitting the data into a smaller number of subsets, a decision can be made for each attribute in a decision tree. The attribute that gives the highest information gain is easily evaluated by this algorithm. Once an attribute is selected, the data set is divided into further subsets. As a result of the tree structure, each inner node corresponds to input attributes, each branch indicates a range of values within that attribute, and each leaf accounts for a good classification.

4.8. RF classifier

To solve the classification problem, one of the successful ensemble techniques used in the machine-learning approach is RF (Mursalin et al., 2017). A collection of Decision Trees is presented here that could act as a single classifier, also enabling multi-classification models and tasks. To achieve the most stable

TABLE 12 Consolidated result analysis of Firefly clusters with the different classifiers.

Classifiers	Sensitivity (%)	Specificity (%)	Accuracy (%)
ANN	100	68.75	84.375
KNN	59.38	100	79.69
I-LDA	100	83.86	91.93
NBC	51.5	100	75.75
Linear SVM	100	84.38	92.19
Polynomial SVM	66.699	100	83.3495
SVM-RBF	79.7125	100	89.85625
QDA	71.09875	100	85.54938
Decision Trees	89.59	100	94.795
RF	100	78.4125	89.20625

TABLE 13 Consolidated result analysis of modified Firefly clusters with the different classifiers.

Classifiers	Sensitivity (%)	Specificity (%)	Accuracy (%)
ANN	57.42563	100	78.71281
KNN	100	89.59	94.795
I-LDA	100	96.3525	98.17625
NBC	100	91.41	95.705
Linear SVM	95.31	100	97.655
Polynomial SVM	95.83	100	97.915
SVM-RBF	100	65.884	82.942
QDA	100	80.225	90.1125
Decision Trees	56.7725	100	78.38625
RF	71.68094	100	85.84047

tree classification, various subsets of the training data are fed to each tree, thereby resulting in a generalized experience for the classifier. The data set is divided into two parts. First, each tree is trained using the bootstrapping method, and the second part is utilized to assess the classification accuracy. To obtain a high-variance classifier, each tree is allowed to reach its maximum depth. The splitting process continues until the pre-defined termination condition is achieved. Once the forest is established, the total number of subsets remains a constant value. From the root node to the leaf node, the journey is made for the unlabeled instances in the classification. The determination of each class that has the maximum votes from all the decision trees is analyzed for a final decision.

5. Results and discussion

The effectiveness of a classifier can be evaluated with many performance evaluation formulas, such as

Sensitivity, Specificity, and Accuracy, and is represented as follows:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (24)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (25)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (26)$$

True Positive (TP) and True Negative (TN) indicate the correct classification of the number of epileptic seizures and seizure-free signals. False Positive (FP) and False Negative (FN) denote the incorrect classification of the number of epileptic seizures and seizure-free signals.

The representation of the Mean Square Error (MSE) is expressed as follows:

$$\text{MSE} = \frac{1}{Z} \sum_{i=1}^Z (V_i - W_j)^2 \quad (27)$$

where the observed value at a given instant of time is expressed as V_i , W_j represents the target value at the model j ; $j=1$ to 10, and Z represents the total number of significant observations per patient - in our case, 4097. The training was carried out in a regressive manner, and the MSE value of the classifiers was drastically reduced to the maximum possible extent. The training of all the classifiers was done to achieve a zero-training MSE. The selection of the hyper parameters in the classifiers is of two types; they are attaining the minimum MSE and the number of iterations with reasonable accuracy. As in the case of the classifier (SVM), the maximum number of iterations is fixed at 450 for a minimum MSE of (10^{-05}) , and apart from this, another constraint of obtaining continuous achievement of the minimum MSE for at least three consecutive iterations is done. In order to avoid the local minima problem, multiple runs of the algorithm are performed. Under this condition, the classifiers are trained to obtain valuable parametric values with reasonable accuracy.

To ensure the effectiveness and reliability of the classifiers, cross-validation is used. For both training and testing, the available data is split up into subsets of equal size. The first subset is chosen as the test set, and the other $K-1$ subsets are combined to form the training and validation sets. After training the classifier using these subsets, the classification performance of the test set is recorded. The process is then repeated so that each of the $K-1$ subsets acts as a test set in turn. The final classification performance is the average of the K test set results. In this paper, a value of 10 was used for K , and the experiment was repeated ten-fold.

5.1. Selection of KNN hyper parameters

Since the problem is of two classes in nature, that is, normal and epileptic sets of EEG signals, the K value for the KNN classifier has to be chosen as two or more than that. In order to select the hyper parameters for the KNN classifier, we identified the penalty function as Euclidean distance, the similarity index error as MSE, and the number of iterations as the cost function. As in the case

of KNN classifiers, the same procedure is repeated for different values of $K = 2, 4$ to identify the optimal number of iterations in terms of a low MSE value. Table 4 shows the performance of MSE with a different number of iterations for KNN classifiers at different K values.

As shown in Table 4, when the value of K is 2, the KNN classifier converges at a low number of iterations (300) at an MSE value of $2.3E-05$, and any further increases in the number of iterations further decrease the MSE value. For $K = 4$, the KNN classifier is either impacted by local minima or has a flattened MSE effect. Therefore, for the KNN classifier, the K value is selected as 2. The same performance of the MSE is also depicted in Figure 8.

5.2. Selection of SVM hyper parameters

The selection of hyperparameters in an SVM classifier is done by two functions, i.e., penalty and cost functions. In this paper, we are discussing three types of SVM classifiers, namely SVM-RBF, Polynomial SVM, and Linear SVM. In the case of SVM-RBF and Polynomial SVM classifiers, the hyper parameters are selected by the grid-based search method. For the SVM-RBF classifier, the gamma value of the RBF kernel must be chosen. For this, MSE will be the penalty function, and the number of iterations will be the cost function as it attains a good gamma value for low MSE at a lower number of iterations. By varying the gamma value of the RBF kernel as (0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.5, 2.0, and 2.2) and increasing the number of iterations in suitable steps, we calculated the MSE of the above mentioned SVM-RBF classifier. Table 5 shows the performance of the MSE in terms of the number of iterations for SVM-RBF at different gamma values.

From Table 5, it can be seen that for a gamma value of 2.0, a lower MSE of $4.00E-08$ is obtained at a number of iterations of 250. Furthermore, in this case, an increase in the number of iterations is shown as a decrease in the MSE value. For all other gamma values, the SVM-RBF classifier is either impacted by local minima or has a flattened MSE effect. Therefore, in the case of the SVM-RBF classifier, the gamma value is selected as 2.0. The same performance of the MSE is also depicted in Figure 9.

As shown in Figure 9, for a gamma value of 2.0, a lower MSE value of $4.00E-08$ is attained for a number of iterations of 250.

5.3. Selection of hyper parameters for the polynomial SVM classifier

The selection of hyper parameters for the Polynomial SVM classifier is performed by the grid search method, just as for the SVM-RBF classifier. The MSE is considered the penalty function, and the number of iterations is the cost function, which controls the order of the polynomial (P). Table 6 shows the performance of the MSE in terms of the number of iterations for the Polynomial SVM classifier at different polynomial orders.

As shown in Table 6, the polynomial of order 2 converges at a lower number of iterations (300) at an MSE value of $1.10E-07$, and any further increases in the number of iterations further decrease

the MSE value. For all other polynomial orders, the Polynomial SVM classifier is either impacted by local minima or has a flattened MSE effect. Therefore, for the Polynomial SVM classifier, the order is selected as 2.0. The same performance of MSE is also depicted in Figure 10.

As shown in Figure 10, for a polynomial order of 2.0, a lower MSE value of $1.10E-07$ is attained for a number of iterations of 300. The MSE value increases as we increase the order of the polynomial beyond 4 and the number of iterations beyond 400.

5.4. Selection of hyper parameters for the SVM linear classifier

The selection of hyper parameters for the Linear SVM classifier incorporates a random search procedure. The Stochastic Gradient Decendent (SGD) algorithm was used to identify the better regression function with a low MSE along with the optimal number of iterations. The operation of the SGD algorithm is controlled by the MSE as the penalty function and the number of iterations as the cost function. Table 7 shows the performance of the MSE in terms of the number of iterations for the Linear SVM classifier for the SGD algorithm.

As shown in Table 7, in the case of the SGD algorithm, the lowest MSE value is attained only after 1,000 (one thousand) iterations. Further, if we increase the number of iterations beyond 1,000, the MSE becomes a constant. Figure 11 shows the performance of the MSE in terms of the number of iterations for the Linear SVM classifier for the SGD algorithm. It is observed from Figure 11 that the number of iterations from 50 to 800 shows that the MSE value of the Linear SVM classifier follows many ups and downs and is also trapped in the local minima case.

Table 8 shows the consolidated result analysis of K-means clusters with the different classifiers. Table 9 shows the consolidated result analysis of FCM clusters with the different classifiers. Table 10 shows the consolidated result analysis of Cuckoo search clusters with the different classifiers. Table 11 shows the consolidated result analysis of Dragonfly clusters with the different classifiers. Table 12 shows the consolidated result analysis of Firefly clusters with the different classifiers. Table 13 shows the consolidated result analysis of Modified Firefly clusters with the different classifiers. Table 14 shows the consolidated MSE analysis with K-means, FCM, Cuckoo search, Dragonfly, Firefly, and Modified Firefly clusters. In this study, we used a 10-fold training and testing strategy.

The result of the analysis of K-means clusters when treated with the different classification techniques shows a high classification accuracy of 98.96% when NBC and Linear SVM are utilized. A low classification accuracy of 78.28% is obtained when ANN is used because of the high false alarm rate. When the K-means clusters are analyzed with the classification techniques, then the performance of the NBC and RF classifiers is impacted by the missed classification.

The result of the analysis of FCM clusters when treated with the different classification schemes shows a high classification accuracy of 98.96% when classified with Decision Trees and a high classification accuracy of 98.43% when classified with QDA. A low classification accuracy of 82.095% is obtained with a high false alarm rate of 35.8% when implemented with ANN.

TABLE 14 Consolidated MSE result analysis with K-means, FCM, Cuckoo search, Dragonfly, Firefly, and Modified Firefly clusters, with classifiers.

Consolidated analysis	K-means clustering	FCM clustering	Cuckoo search clustering	Dragonfly clustering	Firefly clustering	Modified firefly clustering
Classifiers	MSE	MSE	MSE	MSE	MSE	MSE
ANN	9.41E-05	5.04E-05	4E-06	1.37E-05	3.6E-05	7.92E-05
KNN	2.5E-05	2.25E-06	9.61E-06	0.000529	6.89E-05	2.25E-06
I-LDA	6.08E-05	6.25E-06	2.02E-05	4.62E-05	8.41E-06	1.6E-07
NBC	4E-08	2.89E-06	1.6E-07	5.62E-05	0.0004	1.21E-06
Linear SVM	4E-08	1.96E-06	1E-08	1.44E-05	7.84E-06	3.6E-07
Polynomial SVM	4.9E-07	3.6E-07	0.000001	1.23E-05	4.36E-05	2.5E-07
SVM-RBF	8.1E-07	1.76E-05	3.97E-05	4.9E-07	1.52E-05	4.62E-05
QDA	1.21E-06	9E-08	3.25E-05	6.4E-07	2.92E-05	1.44E-05
Decision Trees	3.36E-05	4E-08	4.76E-05	1.44E-06	2.25E-06	9.02E-05
RF	1.44E-05	2.21E-05	1.96E-06	1.76E-05	1.68E-05	2.6E-05

TABLE 15 Performance comparison with other prominent works for the A vs. E classification problem.

Year	References	Technique	Classification accuracy
2022	Prabhakar and Lee (2022a)	Sparse autoencoder with Swarm-based deep learning and reinforcement-based Q-learning	98.55%
2022	Prabhakar and Lee (2022c)	Sparse modeling with an ensemble and nature-inclined classification	98.15%
2020	Zhao et al. (2020)	End-to-end deep learning model	99.52%
2019	Raghu et al. (2019)	Matrix determinant analysis with MLP	99.45%
2018	Sharma et al. (2018)	Orthogonal wavelet analysis with Linear SVM	100%
2017	Diykh et al. (2017)	Weighted complex networks with SVM	100%
2016	Bhardwaj et al. (2016)	Empirical decomposition analysis with soft computing	98.64%
2015	Samiee et al. (2015)	Fourier transform analysis with neural networks	99.80%
2014	Kaya et al. (2014)	Binary patterns generation with Bayesian networks	99%
Proposed performance comparison analysis of clustering with machine learning (2023)	Prabhakar and Won	Clustering through Cuckoo search with Linear SVM	99.48%
Proposed performance comparison analysis of clustering with machine learning (2023)	Prabhakar and Won	a) Clustering through K-means and classifying with NBC and Linear SVM b) FCM clustering and classifying with Decision Trees	98.96%
Proposed performance comparison analysis of clustering with machine learning (2023)	Prabhakar and Won	FCM clustering and classifying with QDA	98.43%

The result of the analysis of Cuckoo search clusters when treated with the different classification techniques shows a high classification accuracy of 99.48% when classified with Linear SVM, while 98.17% is obtained when NBC is utilized as a classification technique. A low classification accuracy of 82.61% is obtained when Decision Trees are implemented with Cuckoo clusters.

The result of the analysis of Dragonfly clusters when treated with the different classification techniques reports a classification accuracy of 97.39% when utilized with the SVM-RBF kernel. A low classification accuracy of 75.5% is obtained when the KNN classifier is implemented.

The result of the analysis of Firefly clusters when treated with the different classification techniques reports a classification accuracy of 94.79% when classified with Decision Trees. A low classification accuracy of 75.75% is obtained when it is classified with NBC.

The result of the analysis of Modified Firefly clusters when treated with the different classification techniques reports a high classification accuracy of 98.17% when classified with I-LDA, and a classification accuracy of 97.91% is obtained when classified with Polynomial SVM. A low classification accuracy of 78.38% is obtained when classified with Decision Trees.

5.5. Comparison with other works

For the classification problem A vs. E, the most important works are compared with our study and tabulated in Table 15.

Even though the proposed work produces slightly lower classification accuracy than the previous study, the main intention of the present study is to show that the combination of clustering with machine learning can prove to be well-suited for the classification of EEG signals for analyzing neurological disorders.

6. Conclusions and future work

Utilizing an EEG to detect epileptic seizures is quite a challenging task that demands a very high level of skill from doctors. The advent of computer-aided detection is a great asset to physicians for the interpretation of EEGs. This study used the concept of feature extraction through various clustering methodologies and then categorized them with the help of suitable post-classifiers. The process is simple and easy to implement. Comparatively higher classification accuracy of 99.48% was achieved when Cuckoo search clusters were utilized with Linear SVM for epilepsy detection. The second highest classification accuracy of 98.96% was obtained when K-Means clusters were utilized with NBC and Linear SVM classifiers. FCM clusters, used with a Decision Trees classifier, also gave the same classification accuracy of 98.96%. The third-best classification accuracy of 98.43% was obtained when FCM clusters were classified with QDA. The lowest classification accuracy, at 75.5%, was obtained when Dragonfly clusters were used with the KNN classifier, and the second lowest classification accuracy, at 75.75%, was obtained when Firefly clusters were classified with NBC. Future works intend to work with various other clustering techniques, such as the Whale optimization algorithm, the Moth Flame optimization algorithm, the Artificial Algae optimization algorithm, the Genetic Bee optimization algorithm, the Gray Wolf optimization algorithm, and the Fish Swarm optimization algorithm, along with different pattern recognition techniques and advanced machine learning methodologies, for effective classification of epileptic seizures in other combinations of the Bonn data set. Future works also aim to incorporate the concept of clustering and machine learning for the efficient classification of other neurological disorders. Future works, finally, aim to include a variety of transfer and deep learning techniques for the efficient classification of epilepsy from EEG signals.

References

- Abd Elazim, S. M., and Ali, E. S. (2016). Optimal power system stabilizers design via cuckoo search algorithm. *Int. J. Electrical Power Energy Syst.* 75, 99–107. doi: 10.1016/j.ijepes.2015.08.018
- Acharya, U. R., Molinari, F., Vinitha, S., and Chattopadhyay, S. (2012). Automatic diagnosis of epileptic EEG using entropies. *Biomed. Signal Process Control* 7, 401–408. doi: 10.1016/j.bspc.2011.07.007
- Acharya, U. R., Oh, S. O., Hagiwara, Y., Tan, J. H., and Adeli, H. (2018). Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Comput. Biol. Med.* 100, 270–278. doi: 10.1016/j.compbiomed.2017.09.017
- Adamovic, S., Miskovic, V., Macek, N., Milosavljevic, M., Sarac, M., Saračević, M., et al. (2020). An efficient novel approach to iris recognition based on stylometric features and machine learning techniques. *Future Generat. Computer Syst.* 107, 144–157. doi: 10.1016/j.future.2020.01.056
- Andrzejak, R. G., Lehnertz, K. C., Rieke, F., and Mormann, P. (2001). Indications of non linear deterministic and finite dimensional structures in time series of brain electrical activity: dependence on recording region and brain state. *Phys. Rev. E Statist. Nonlinear Soft Matter Phys.* 64, 061907. doi: 10.1103/PhysRevE.64.061907

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://www.upf.edu/web/ntsa/downloads/-/asset_publisher.

Author contributions

SP: concept, methods, implementation, visualization, and writing draft. D-OW: visualization, critical analysis, revision, supervision, and funding acquisition. Both authors contributed to the article and approved the submitted version.

Funding

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00451, Development of BCI based Brain and Cognitive Computing Technology for Recognizing User's Intentions using Deep Learning) and partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (Nos. 2022R1A5A8019303 and 2022R1F1A1074640).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- Bhardwaj, A., Tiwari, A., Krishna, R., and Varma, V. (2016). A novel genetic programming approach for epileptic seizure detection. *Comput. Methods Programs Med.* 124, 2–18. doi: 10.1016/j.cmpb.2015.10.001
- Bulthoff, H. H., Lee, S. W., Poggio, T. A., and Wallraven, C. (2003). *Biologically Motivated Computer Vision*. Tübingen: Springer-Verlag.
- Chen, G., Xie, W., Bui, T. D., and Krzyzak, A. (2017). Automatic epileptic seizure detection in EEG using nonsubsampling wavelet Fourier features. *J. Med. Biol. Eng.* 37, 123–131. doi: 10.1007/s40846-016-0214-0
- Chu, D., Liao, L. Z., Ng, M. K. P., and Wang, X. (2015). Incremental linear discriminant analysis: a fast algorithm and comparisons. *IEEE Trans. Neural Networks Learn. Syst.* 26, 2716–2735. doi: 10.1109/TNNLS.2015.2391201
- Diyyk, M., Li, Y., and Peng, W. (2017). Classify epileptic EEG signals using weighted complex networks based community structure detection. *Expert Syst. Appl.* 90, 87–100. doi: 10.1016/j.eswa.2017.08.012
- Gotman, J. (1982). Automatic recognition of epileptic seizures in the EEG. *Electroencephalogr. Clin. Neurophysiol.* 54, 530–540. doi: 10.1016/0013-4694(82)90038-4
- Haasan, A. R., and Siuly, S. (2016). Epileptic seizure detection in EEG signals using tunable-q factor wavelet transform and bootstrap aggregating. *Comput. Methods Programs Biomed.* 137, 247–259. doi: 10.1016/j.cmpb.2016.09.008
- Heijden, F. V. D., Duin, R. P. W., Ridder, D. D., and Tax, D. M. J. (2005). *Classification, Parameter Estimation and State Estimation*. Chichester: John Wiley & Sons.
- Islam, M. J., Wu, Q. J., Ahmadi, M., and Sid-Ahmed, M. (2007). “Investigating the performance of naive-Bayes classifiers and k-nearest neighbor classifiers,” in *2007 International Conference on Convergence Information Technology (ICCIT 2007)*, Gyeongju, 1541–1546.
- Jeong, J. (2004). EEG dynamics in patients with Alzheimer’s disease. *Clin. Neurophysiol.* 115, 1490–1505. doi: 10.1016/j.clinph.2004.01.001
- Jukic, S., Saračević, M., Subasi, A., and Kevric, J. (2020). Comparison of ensemble machine learning methods for automated classification of focal and non-focal epileptic EEG signals. *Mathematics* 8, 1–16. doi: 10.3390/math8091481
- Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., and Wu, A. (2004). A local search approximation algorithm for k-means clustering. *Comput. Geom.* 28, 89–112. doi: 10.1016/j.comgeo.2004.03.003
- Kaya, Y., Uyar, M., Tekin, R., and Yildirim, S. (2014). 1D-local binary pattern based feature extraction for classification of epileptic EEG signals. *Appl. Math. Comput.* 243, 209–219. doi: 10.1016/j.amc.2014.05.128
- Kim, J. H., Bießmann, F., and Lee, S. W. (2014). Decoding three-dimensional trajectory of executed and imagined arm movements from electroencephalogram signals. *IEEE Trans. Neural Syst. Rehabil. Eng.* 23, 867–876. doi: 10.1109/TNSRE.2014.2375879
- Kumar, Y., Dewal, M., and Anand, R. (2014). Epileptic seizures detection in EEG using DWT based ApeEn and artificial neural network. *Signal Image Video Process* 8, 1323–1334. doi: 10.1007/s11760-012-0362-9
- Kwak, N. S., Müller, K. R., and Lee, S. W. (2017). A convolutional neural network for steady state visual evoked potential classification under ambulatory environment. *PLoS ONE* 12, e0172578. doi: 10.1371/journal.pone.0172578
- Lee, M. H., Fazli, S., Mehnert, J., and Lee, S. W. (2015). Subject-dependent classification for robust idle state detection using multi-modal neuroimaging and data-fusion techniques in BCI. *Pattern Recognit.* 48, 2725–2737. doi: 10.1016/j.patcog.2015.03.010
- Magosso, E., Ursino, M., Zaniboni, A., and Gardella, E. (2009). A wavelet-based energetic approach for the analysis of biomedical signals: application to the electroencephalogram and electrooculogram. *Appl. Math. Comput.* 207, 42–62. doi: 10.1016/j.amc.2007.10.069
- Moghim, N., and Corne, D. W. (2014). Predicting epileptic seizures in advance. *PLoS ONE* 9, e99334. doi: 10.1371/journal.pone.0099334
- Mursalin, M., Zhang, Y., Chen, Y., and Chawla, N. V. (2017). Automated epileptic seizure detection using improved correlation-based feature selection with random forest classifier. *Neurocomputing* 241, 204–214. doi: 10.1016/j.neucom.2017.02.053
- Peker, M., and Sen, B. (2016). A novel method for automated diagnosis of epilepsy using complex-valued classifiers. *IEEE J. Biomed. Health Inform.* 20, 108–118. doi: 10.1109/JBHI.2014.2387795
- Pincus, S. M. (1991). Approximate entropy as a measure of system complexity. *Proc. Natl. Acad. Sci. U. S. A.* 88, 2297–2301. doi: 10.1073/pnas.88.6.2297
- Prabhakar, S. K., and Lee, S. W. (2022a). SASDL and RBATQ: sparse autoencoder with swarm based deep learning and reinforcement based Q-learning for EEG classification. *IEEE Open J. Eng. Med. Biol.* 3, 58–68. doi: 10.1109/OJEMB.2022.3161837
- Prabhakar, S. K., and Lee, S. W. (2022b). Improved sparse representation with robust hybrid feature extraction models and deep learning for EEG classification. *Expert Syst. Appl.* 198, 116783. doi: 10.1016/j.eswa.2022.116783
- Prabhakar, S. K., and Lee, S. W. (2022c). ENIC: ensemble and nature inclined classification with sparse depiction based deep and transfer learning for biosignal classification. *Appl. Soft Comput.* 117, 108416. doi: 10.1016/j.asoc.2022.108416
- Raghu, S., Sriraam, N., Hegde, A. S., and Kubben, P. L. (2019). A novel approach for classification of epileptic seizures using matrix determinant. *Expert Syst. Appl.* 127, 323–341. doi: 10.1016/j.eswa.2019.03.021
- Rajaguru, H., and Prabhakar, S. K. (2016). A framework for epilepsy classification using modified sparse representation classifiers and native bayesian classifier from EEG signals. *J. Med. Imaging Health Inform.* 6, 1829–1837. doi: 10.1166/jmhi.2016.1935
- Riaz, F., Hassan, A., Rehman, S., Niazi, I. K., and Dremstrup, K. (2016). EMD based temporal and spectral features for the classification of EEG signals using supervised learning. *IEEE Trans. Neural Syst. Rehabil. Eng.* 24, 28–35. doi: 10.1109/TNSRE.2015.2441835
- Richman, J. S., and Moorman, J. R. (2000). Physiological time-series analysis using approximate entropy and sample entropy. *Am. J. Physiol. Heart Circ. Physiol.* 278, H2039–H2049. doi: 10.1152/ajpheart.2000.278.6.H2039
- Rivero, D., Blanco, E. F., Dorado, J., and Pazos, A. (2011). “A new signal classification technique by means of genetic algorithms and kNN,” in *Proceedings of the IEEE Congress of Evolutionary Computation (CEC ’11)*, New Orleans, LA, 581–586.
- Samiee, K., Kovacs, P., and Gabbouj, M. (2015). Epileptic seizure classification of EEG time-series using rational discrete short-time Fourier transform. *IEEE Trans. Biomed. Eng.* 62, 541–555. doi: 10.1109/TBME.2014.2360101
- Sezer, E., Işık, H., and Saracoglu, E. (2012). Employment and comparison of different artificial neural networks for epilepsy diagnosis from EEG signals. *J. Med. Syst.* 36, 347–362. doi: 10.1007/s10916-010-9480-5
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell Syst. Tech. J.* 27, 379–423. doi: 10.1002/j.1538-7305.1948.tb01338.x
- Sharma, M., Bhurane, A. A., and Acharya, U. R. (2018). MMSFL-OWFB: A novel class of orthogonal wavelet filters for epileptic seizure detection. *Knowledge Based Syst.* 160, 265–277. doi: 10.1016/j.knsys.2018.07.019
- Song, Y., Huang, J., Zhou, D., Zha, H., and Giles, C. L. (2007). “Iknn: Informative k-nearest neighbor pattern classification,” in *Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery (Springer)*, 248–264.
- Sree Ranjini, K. S., and Murugan, S. (2017). Memory based hybrid dragon?y algorithm for numerical optimization problems. *Expert Syst. Appl.* 83, 63–78. doi: 10.1016/j.eswa.2017.04.033
- Srinivasan, V., Eswaran, C., and Sriraam, N. (2007). Approximate entropy-based epileptic EEG detection using artificial neural networks. *IEEE Trans. Inf. Technol. Biomed.* 11, 288–295. doi: 10.1109/ITTB.2006.884369
- Swami, P., Godiyal, A. K., Santhosh, J., Panigrahi, B. K., Bhatia, M., Anand, S., et al. (2014). “Robust expert system design for automated detection of epileptic seizures using SVM classifier,” in *Proceedings of the 2014 3rd IEEE International Conference on Parallel, Distributed and Grid Computing, PDGC 2014, India*, 219–222.
- Tawfik, N. S., Youssef, S. M., and Kholief, M. (2016). A hybrid automated detection of epileptic seizures in EEG records. *Comput. Electr. Eng.* 53, 177–190. doi: 10.1016/j.compeleceng.2015.09.001
- Tilahun, S. L., and Ong, H. C. (2012). Modified firefly algorithm. *J. Appl. Math.* 2012, 1–12. doi: 10.1155/2012/467631
- Tzallas, A. T., and Tsipouras, M. G. (2007). Automatic seizure detection based on time-frequency analysis and artificial networks. *Comput. Intell. Neurosci.* 2007, 80510. doi: 10.1155/2007/80510
- Wang, Y., Simon, M., Bonde, P., Harris, B. U., Teuteberg, J. J., Kormos, R. L., et al. (2012). Prognosis of right ventricular failure in patients with left ventricular assist device based on decision tree with SMOTE. *IEEE Trans. Inform. Technol. Biomed.* 16, 383–390. doi: 10.1109/ITTB.2012.2187458
- Won, D. O., Hwang, H. J., Dähne, S., Müller, K. R., and Lee, S. W. (2015). Effect of higher frequency on the classification of steady-state visual evoked potentials. *J. Neural Eng.* 13, 016014. doi: 10.1088/1741-2560/13/1/016014
- Yang, X. S. (2010). *Nature-Inspired Metaheuristic Algorithm*, 2nd Edn. Beckington: Luniver Press.
- Yeom, S. K., Fazli, S., Müller, K. R., and Lee, S. W. (2014). An efficient ERP-based brain-computer interface using random set presentation and face familiarity. *PLoS ONE* 9, e111157. doi: 10.1371/journal.pone.0111157
- Zamir, Z. R. (2016). Detection of epileptic seizure in EEG signals using linear least square preprocessing. *Comput. Methods Programs Biomed.* 133, 95–109. doi: 10.1016/j.cmpb.2016.05.002
- Zeng, K., Yan, J., Wang, Y., Sik, A., and Ouyang, G. (2016). Automatic detection of absence seizures with compressive sensing EEG. *Neurocomputing* 171, 497–502. doi: 10.1016/j.neucom.2015.06.076

Zhang, L., Lu, W., Liu, X., Pedrycz, W., and Zhong, C. (2016). Fuzzy c-means clustering of incomplete data based on probabilistic information granules of missing values. *Knowledge Based Syst.* 99, 51–70. doi: 10.1016/j.knosys.2016.01.048

Zhang, T., and Chen, W. (2017). LMD based features for the automatic seizure detection of EEG signals using SVM. *IEEE Trans. Neural Syst. Rehabil. Eng.* 28, 1100–1108. doi: 10.1109/TNSRE.2016.2611601

Zhao, W., Zhao, W., Wang, W., Jiang, X., Zhang, X., Peng, Y., et al. (2020). A novel deep neural network for robust detection of seizures using EEG signals. *Comput. Math. Models Med.* 2020, 968982. doi: 10.1155/2020/9689821

Zhou, M., Tian, C., Rui, C., Wang, B., Niu, Y., Hu, T., et al. (2018). Epileptic seizure detection based on EEG signals and CNN. *Front. Neuroinform.* 12, 95. doi: 10.3389/fninf.2018.00095

Zhou, W., Liu, Y., Yuan, Q., and Li, X. (2013). Epileptic seizure detection using lacunarity and Bayesian linear discriminant analysis in intracranial EEG. *IEEE Trans. Biomed. Eng.* 60, 3375–3381. doi: 10.1109/TBME.2013.2254486

Zhu, X., Suk, H. I., Lee, S. W., and Shen, D. (2016). Canonical feature selection for joint regression and multi-class identification in Alzheimer's disease diagnosis. *Brain Imaging Behav.* 10, 818–828. doi: 10.1007/s11682-015-9430-4