Hybrid Optimized Techniques for EEG Based Mild Cognitive Impairment Detection Using Time Domain Feature Extraction

Saroja Pathapati[*], N. J. Nalini

Department of Computer Science and Engineering, Faculty of Engineering and Technology, Annamalai University, Chidambaram 608002, India

Corresponding Author Email: pathapati.saroja@gmail.com

ABSTRACT

Mild Cognitive Impairment (MCI) represents early cognitive changes that can signal the potential development of more serious memory and thinking problems. The EEG data is preprocessed by applying band pass filter and segmenting it into epochs of 5 secs. Subsequently, time domain feature extraction techniques including Kurtosis, Zero Crossing Rate (ZCR) and Hjorth parameters are explored and applied to the EEG signals. The investigation includes the integration of these techniques with 1D deep learning techniques like Convolutional Neural Networks (CNN) and Convolutional Recurrent Neural Networks (CRNN) and hybrid 1D deep optimized models like PBCNN (Population Based CNN) and PBCRNN (Population-Based CRNN). The impact of feature extraction on MCI detection accuracy is evaluated by comparing the results obtained with and without feature extraction. Additionally, the influence of epoch duration, considering 5-second epochs with 1 second overlap, is examined to determine the optimal duration for precise MCI classification using EEG data. The findings contribute to advancing the understanding of EEG data analysis techniques for early MCI detection. The proposed methods have significant clinical application value in the early screening and diagnosis of Mild Cognitive Impairment (MCI). Among the proposed models 1DPBCRNN works well with 90.01 accuracy.

1. INTRODUCTION

MCI is a state where there is a noticeable decline in cognitive abilities greater than what would typically be anticipated for an individual's age, yet it does not lead to significant impairment in their day-to-day activities. Early detection of MCI is crucial for timely intervention and management of cognitive decline and if not identified earlier may lead to Dementia or Alzheimer's disease [1, 2]. Electroencephalography (EEG) is becoming increasingly recognized as a valuable tool in the detection of MCI due to its nonintrusive characteristics and ability to capture neural activity in real-time [3].

The removal of Electrooculogram (EOG) artifacts in diverse EEG signals is achieved using a novel approach that combines Independent Component Analysis (ICA) and Ensemble Empirical Mode Decomposition (EEMD) is introduced to demonstrate improved performance in EOG artifact rejection, making it a promising method for EEG signal processing and analysis [4]. A comprehensive artifact removal methods like Principal Component Analysis (PCA), Canonical correlation analysis (CCA), Empirical Mode Decomposition (EMD), and Filtering methods for EEG signals were discussed for effectively removing artifacts from EEG signals [5].

For detecting cognitive impairments and neurological disorders, the application of Permutation Entropy (PE) and Statistical Complexity (SC) in the case of MEG recordings from patients detected with MCI and Alzheimer's disease (AD) examines broadband signals and breaks them down into frequency bands to identify changes in each band linked to MCI and AD [6]. The feature extraction scheme based on discrete wavelet transform demonstrates the potential of relative wavelet energy features to categorize EEG signals obtained during intricate cognitive tasks and rest conditions. The performance evaluation with different classifiers highlights the ability of this approach to achieve high accuracy rates in EEG signal classification [7].

The effectiveness of various machine learning algorithms by feature extraction of input EEG samples belonging to motor movements is conducted and shown that the Medium-ANN model emerges as the top performer, suggesting the applicability of the approach in scenarios like brain-computer interfaces and neural prostheses [8]. Regarding neural decoding and brain computer interfaces, Saeidi et al. [9] illustrate the latest developments in the decoding and categorization of EEG signals made possible by supervised ML and DL models. This study explores the increasing importance of deep learning in the analysis of EEG data through a range of tasks, such as finding emotions, mental workload assessment, detection of seizures, program-based potential identification, and sleep scoring [10]. Furthermore, the research delves into the input formulations employed for deep network training and investigates distinct network

architectures tailored to various task categories. CNN and RNN consistently demonstrate superior classification accuracy compared to alternative models.

Existing methods for processing EEG signals and classifying MCI are helpful but still have many issues that need fixing. Techniques like PCA, CCA, EMD, and filtering often have trouble separating noise from real brain signals, especially when the data is messy or comes from different sources. Even though ICA and EEMD perform better, they are too complicated and slow to work well in real-time situations.

For feature extraction, methods like Permutation Entropy (PE) and Statistical Complexity (SC) are promising but aren't fully designed to work well with EEG data. They miss out on important patterns in the signal. Wavelet-based methods also depend on choosing the right settings, which might not work for all tasks. Machine learning and deep learning models like CNNs and RNNs often do well but can struggle with overfitting, require a lot of labeled data, and might not work as well on new datasets. Most studies focus on simpler tasks like detecting movements or emotions, so they don't work well for complex conditions like MCI. There's also not enough research comparing these models across bigger and more varied datasets to figure out which ones work best.

The proposed model tries to solve these problems by improving artifact removal, using features from time-domain analysis, and comparing advanced models like 1D CNN, 1D CRNN, 1D PBCNN, and 1D PBCRNN on different EEG datasets. The goal is to make it easier to detect MCI early and create better tools to understand and diagnose this condition.

2. RELATED WORK

Cognitive impairment, particularly Mild Cognitive Impairment (MCI) and its progression to Alzheimer's disease, has been a significant focus of research. Richardson et al. [1] conducted a longitudinal study that highlighted a two-decade change in the prevalence of cognitive impairment in the UK, emphasizing the need for early detection and intervention strategies. Batum et al. explored the neurocognitive clues linking MCI to Alzheimer's disease, providing insights into the cognitive deficits that characterize these conditions.

EEG Signal Processing
Electroencephalography (EEG) has emerged as a vital tool for understanding brain activity related to cognitive functions. Sur and Sinha [11] provided a foundational overview of EEG and event-related potentials (ERPs), which are crucial for studying cognitive processes. The removal of artifacts from EEG signals is essential for accurate analysis, as highlighted by Jiang et al. [5], who reviewed various techniques for artifact removal, and Al-Baddai et al. [12], who proposed a novel method combining independent component analysis and ensemble empirical mode decomposition.

Machine Learning in EEG Analysis
The application of machine learning techniques to EEG data has gained traction, particularly for diagnosing cognitive impairments. Prochazka et al. [13] demonstrated the effectiveness of wavelet transform and machine learning for feature extraction and classification of EEG signals. Ramírez-Arias et al. [8] evaluated various machine learning algorithms for EEG classification, while Glaser et al. [14] provided a systematic review of neural decoding methods using machine learning. Kotowski et al. [15] further explored deep learning approaches for EEG classification tasks, indicating a shift towards more sophisticated models in this domain.

Feature Extraction and Complexity Analysis
Feature extraction plays a crucial role in enhancing the performance of machine learning models. Movahed et al. [16] focused on the automatic diagnosis of MCI using spectral features from EEG signals and supervised dictionary learning techniques, respectively. Timothy et al. [17] analyzed permutation entropy in MCI, contributing to the understanding of EEG signal characteristics in these conditions.

Real-time Applications and Future Directions
The integration of EEG signal processing methods into real-time applications, such as brain-computer interfaces (BCIs), is an emerging area of research. EEG signal processing methods specifically for brain-controlled robots, while applied a 1D convolutional neural network to classify voluntary eye blinks in EEG signals for BCI applications. Introduced a CNN-LSTM model for epileptic seizure recognition, showcasing the potential of deep learning in real-time EEG analysis.

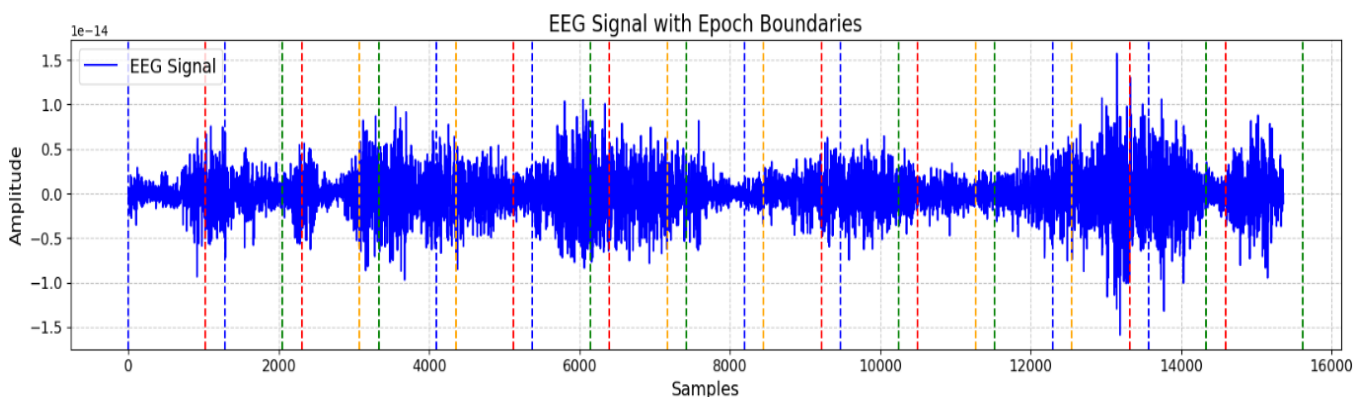## 3. METHODS

3.1 Preprocessing techniques

Figure 1. Sample EEG signal shown for 1 minute

In recent times, there has been a rising interest in the examination of EEG data for the detection of MCI. However, the raw EEG signals are often noisy and contain various artifacts that can hinder accurate interpretation. To address these challenges, preprocessing techniques, such as filtering and epoch segmentation, are applied to enhance the signal

quality and facilitate subsequent analysis [18]. The EEG data is preprocessed to enhance its quality and prepare it for further analysis. Firstly, the EEG data is read from EDF files using the MNE library. Then the data undergoes re-referencing with a common average reference to reduce the presence of noise. A bandpass filter is applied to retain frequencies between 0.5 Hz and 45 Hz, eliminating unwanted artifacts. Subsequently, the data is segmented into fixed-length epochs of 5 seconds with a 1-second overlap, facilitating easier analysis. Finally, the preprocessed EEG data is organized into epochs and can be utilized for various tasks like feature extraction or classification. Figure 1 displays a one-minute duration sample EEG signal.

3.2 Feature extraction techniques

In addition to preprocessing, feature extraction plays a vital role in capturing appropriate data from EEG signals. Time domain features, including Kurtosis and Hjorth parameters and Zero Crossing Rate (ZCR) have shown promise in quantifying signal characteristics associated with cognitive impairment. These features provide insights into the amplitude, variability, and distribution of the EEG signals, which can be indicative of underlying cognitive dysfunction.

3.2.1 Time domain features
A time domain feature is a statistical measure that characterizes the amplitude or waveform of an EEG signal at a specific point in time [19].
a) Kurtosis:
Kurtosis is a valuable time domain feature for EEG signal analysis, as it provides insights into the shape and distribution of EEG data. Extracting kurtosis features allow researchers and clinicians to capture and analyze important characteristics of EEG signals related to brain activity. Kurtosis helps us to identify the peakedness or flatness of the distribution of brain activity [20]. If kurtosis is high, it indicates that the brain signal has high spikes. These sharp peaks can happen during very strong brain activity or when something is wrong, like seizures, memory problems, or diseases like Alzheimer's. Low kurtosis shows that brain activity is normal.

By quantifying the degree of these peaks and tails, kurtosis allows the detection of irregularities in neural activity, which are often associated with conditions like MCI or Alzheimer's. The kurtosis formula is shown in Eq. (1).

$$Kurtosis = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s}\right)^4 - 3 \qquad (1)$$

where, n stands for the total number of data points within the dataset, i takes on values 1, 2, 3, ..., n, x_i represents each specific data point in your dataset, $\bar{x}$ represents the sample mean or average of your dataset.

This is computed by adding together all the data points $\sum x_i$ and then dividing the sum by the total number of data points n. s represents the sample standard deviation of your dataset. $\left(\frac{x_i - \bar{x}}{s}\right)^4$ calculates the fourth s power of the standardized deviation of each data point. It measures how each data point deviates from the mean, scaled by the standard deviation. The final step in the formula is subtracting 3. This is done to make the kurtosis of a normal distribution equal to 0. In other words, if the calculated Kurtosis is greater than 0, it indicates heavier tails than a normal distribution, and if it's less than 0, it

indicates lighter tails.
The input provided for the kurtosis_features function is a 3-dimensional array with dimensions (number of epochs, number of channels, number of data points per channel). This array contains EEG data for two groups: the control group and the patient group.
The kurtosis_features function is designed to calculate the kurtosis value for each epoch and each channel separately. As the function iterates through each epoch and channel, it calculates the kurtosis value for the data recorded in that specific epoch and channel using the "kurtosis" function from the scipy. stats module. The resulting kurtosis values are stored in a new array called "kurtosis_features." After the function has processed all the epochs and channels, it returns the "kurtosis_features" array. The shape of this array is (num_epochs, num_channels), where "num_epochs" initializes the number of epochs, and "num_channels" signifies the count of electrodes.In the input data, there are 14757 epochs in MCI group and 13645 epochs in HC group, and each epoch contains 19 channels. Therefore, the resulting "Patient kurtosis features" and "Control kurtosis features" arrays have shapes (14757, 19) and (13645, 19), respectively.
b) Hjorth:
Hjorth features are a group of time-domain attributes utilized for the characterization of signals, particularly in the context of EEG signals. They were introduced by Dr. Bjorn Hjorth to quantify the activity, mobility and complexity of signals. In EEG signal processing, Hjorth features provide valuable information about the signal's mobility and complexity. Mobility represents the rate of change or activity in the signal, while complexity measures how irregular or diverse the signal's behavior is over time.
Hjorth Mobility shows us how often the EEG signal varies over time. It shows whether the signal is constant or fluctuating. The mobility with a higher score means that the signal is varying at a faster rate and this can occur during increased brain activity or even brain pathology. A low mobility number indicates that the signal is much more constant and smoother. This is useful since it allows one to determine changes in brain activity for instance when the brain is in function or if there are any functional abnormalities. The mobility evaluation is shown in Eq. (2).

$$Mobility = \sqrt{\frac{\text{var}\left(\frac{dx(t)}{dt}\right)}{\text{var}(x(t))}} \qquad (2)$$

where, $\left(\frac{dx(t)}{dt}\right)$ represents the first derivative of the function $x(t)$ over time t, $var\left(\frac{dx(t)}{dt}\right)$ calculates the variance of $\left(\frac{dx(t)}{dt}\right)$. $var(x(t))$ calculates the variance of the position values $x(t)$ over time. $\sqrt{\frac{var\left(\frac{dx(t)}{dt}\right)}{var(x(t))}}$ calculates the square root of the ratio between the variance of the signal's derivative and the variance of the signal.
Hjorth Complexity measures the signal's irregularity or variability, capturing how unpredictable or difficult the signal's behavior is higher complexity means the brain activity is more complicated and has lots of different patterns. This can happen when the brain is solving problems or when something is wrong. Lower complexity means the brain activity is simpler, more repetitive, or not as busy. The calculation of hjorth complexity is shown in Eq. (3).

$$Complexity = \sqrt{\frac{Mobility\left(\frac{dy(t)}{dt}\right)}{Mobility(y(t))}} \qquad (3)$$

where, $Mobility\left(\frac{dy(t)}{dt}\right)$ calculates the first derivative of Hjorth Mobility and $Mobility(y(t))$ calculates the Hjorth Mobility of the original signal, measuring its overall mobility [21].

The input to the hjorth_features function is a 3D array called data_array, which contains the EEG signal data. To calculate the Hjorth mobility and complexity features, the function iterates over each epoch and channel in the data_array. For each epoch and channel, It computes the signal's first derivative to determine mobility, and then further computes the first derivative of the first derivative to calculate complexity.

In total, the function extracts two sets of features for each epoch and channel: the Hjorth mobility features and the Hjorth complexity features. Since there are num_epochs epochs and num_channels channels, the resulting hjorth_mobility_features and hjorth_complexity_features arrays have the shape (num_epochs, num_channels). To combine the mobility and complexity features into a single feature vector, the function uses np.column_stack to stack the two feature arrays horizontally. As a result, the final Hjorth features arrays, control_hjorth_features, and patient_hjorth_features, have the shape (num_epochs, 38). Each row in these arrays represents a different epoch, and the 38 columns contain the combined mobility and complexity features for each channel of the EEG signal.

c) Zero Crossing Rate (ZCR):

ZCR counts how many times the EEG signal crosses the zero line. This shows how much the signal goes up and down. A high ZCR means the signal changes quickly, like during bursts of brain activity. A low ZCR means the signal is smoother and changes less, like when the brain is resting or calm. This is important for finding problems with thinking or memory because the brain's patterns might look different from normal.

These features help us understand EEG signals better by looking at how they work. They make it easier to find small but important changes in the signals that might show problems with thinking or memory. This helps with studying and finding out what might be wrong. It helps identify sharp waveforms, bursts of activity, or the onset and offset of specific events in the EEG signal [22]. The ZCR evaluation is shown in Eq. (4).

$$ZCR = \frac{1}{N-1}\sum_{i=1}^{N-1}\begin{cases}1 & \text{if } x[i].x[i-1] < 0 \\ 0 & \text{otherwise}\end{cases} \qquad (4)$$

where, $x[i]$ is the sample at index i in the signal, $x[i-1]$ is the sample immediately preceding $x[i]$. $x[i].$ $x[i-1]<0$ checks if the product of $x[i]$ and $x[i-1]$ is less than zero, indicating a sign change (crossing). The summation runs over all pairs of consecutive samples, and 1 is added to the sum if a sign change is detected; otherwise, 0 is added.

The input to the zcr_features function is a 3D array representing EEG data, comprising the number of epochs, channels (electrodes), and data points per channel. The function computes the ZCR value for each epoch and channel separately, Tallying the occurrences when the EEG signal intersects the zero baseline. and dividing it by the total length of the epoch.

As a result, the function extracts a ZCR feature for each epoch and channel in the EEG data. Ultimately, the ZCR features matrix provides valuable information about the rate of zero crossings, with each row representing an epoch and each column representing a specific EEG channel. For the control group, there are 13645 epochs, each containing 19 channels resulting in "Control ZCR features" array has a shape of (13645, 19) and for the patient group, there are 14757 epochs, each containing 19 channels resulting in "Patient ZCR features" array has a shape of (14757, 19).

3.2.2 Deep learning techniques

To improve the effectiveness of MCI detection, more advanced machine learning techniques have to be utilized. Specifically, 1D deep learning models like CNN and CRNNs, along with hybrid models such as Population Based CNN (PBCNN) and Population Based CRNN (PBCRNN), have shown remarkable capabilities in capturing spatial and temporal dependencies within EEG data.

a) 1D Convolutional Neural Network:

1D CNN is a popular neural-network architecture which is used for various signal processing tasks, including EEG signal analysis [23]. The temporal aspect of EEG data is crucial for understanding brain activity, and 1D CNN is well-suited for handling this dimension. Furthermore, the network is able to extract both local and global features of the EEG signal, thus making it an effective and robust tool for EEG signal analysis.
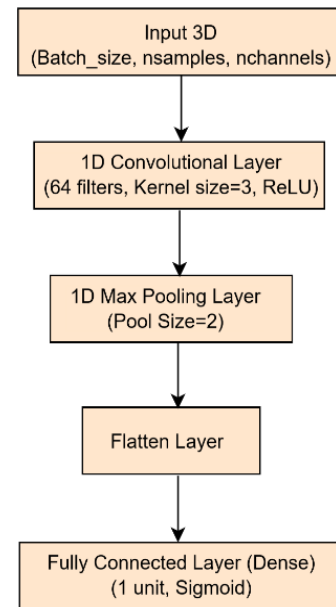

Figure 2. 1D CNN architecture

Figure 2 shows the architecture of 1D CNN which is used for EEG signal analysis. The 1D Convolutional Neural Network is used to study EEG signals and find patterns in the data. The input to the model is like a 3D tensor with parts that include the batch size set to 32 which is how many examples the model looks at one time, epochs set to 30 and features set to 76. The first layer of the model is a 1D convolutional layer that uses 32 filters to capture important details, kernel size of 3 to find small patterns in the EEG signals and the ReLU activation function, which helps the model learn better and faster. After this, the data goes through a 1D max pooling layer with a pool size of 2, which makes the data smaller by keeping

only the most important parts, so the model can work more efficiently.

Next, the output from the pooling layer is flattened into a single line, which is sent to the fully connected layers. The first dense layer has 64 units and uses ReLU activation, helping it find bigger patterns in the data. Finally, the model gives one result using a sigmoid activation function, which gives a value between 0 and 1. A value close to 0 means the data is from the healthy control group, and a value close to 1 means it's from the mild cognitive impairment group. The model learns using binary cross-entropy loss, which is used for binary classification, and the Adam optimizer, which helps the model learn quickly and accurately. This design makes the CNN good at studying EEG data and classifying it correctly.

b) 1D Convolutional Recurrent Neural Network (1D CRNN)

The CRNN architecture is useful for tasks that involve sequential data with long-term dependencies, such as speech, signal, and language processing [24]. The convolutional layers can identify spatial patterns present in the input data, including the frequency features within a signal., while the recurrent layers can capture the temporal patterns. The 1D CRNN architecture is shown in Figure 3.
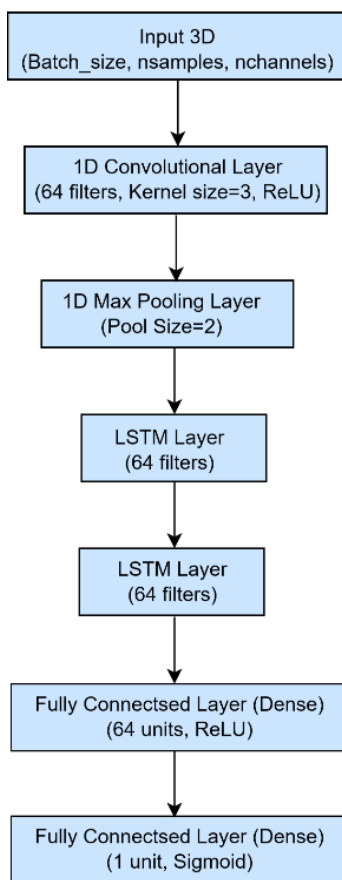

Figure 3. 1D CRNN architecture

The model starts with a normal 1D convolutional layer which has 64 filters, kernel size of 3 in order to capture the basic and frequency characteristics of the EEG signals and a ReLU activation function which assist in the detection of complex patterns. Then max pooling to applied to convolutional layer with pool size 2 through which the data size can be reduced while keeping the most important information from the previous layer. Then the model has two

LSTM layers each with 64 units each. These layers help in comprehending the variations in the data at a certain interval and identifying trends in the data over a long period. Following the LSTM layers the data is supplied to a dense layer with 64 units with ReLU activation function. This layer assists in enhancing the features identified in the prior layers. Last, the model contains one output layer with one unit and uses the sigmoid activation function. This layer outputs a value between 0 with and binary 1 cross for entropy the loss classification function of and the optimizer signal Adam as is control used or which patient. helps the in-model training is the trained model faster and more accurately.

3.2.3 Hybrid deep optimized techniques

a) Population Based Training (PBT)

PBT is an optimization technique within machine learning and deep learning that aims to optimize hyperparameter tuning efficiency and elevate the performance of trained models. It utilizes the principles of genetic algorithms to iteratively evolve a population of models over multiple training epochs. The PBT process typically involves the steps like- Population Initialization, Model Evaluation, Exploitation and Exploration, Hyperparameter Exploration, Population Update. By dynamically allocating more resources (e.g., computation time, memory) to promising models and discarding fewer promising ones early in the training process, PBT converges to better hyperparameter settings more efficiently than traditional hyperparameter optimization methods like grid or random search. This makes PBT particularly advantageous when training resource-intensive deep learning models. PBT has gained popularity for its ability to efficiently explore the hyperparameter space and find optimal configurations, enabling faster and more effective training of machine learning models [25].

Population Based Training (PBT) Algorithm

(1) Start with a population of 5 models and choose the hyperparameters of each model at random. Filters are selected as any number between 32 and 128, the kernel size is chosen from the set of 2 to 5 and the number of dense units can also be between 32 and 128. Make the weights of each model to be any random value.

(2) For each model in the population, fine tune each model for every epoch using the training data set.

(3) To determine the validation accuracy of each model, test each model on the validation dataset

(4) Sort the models in the decreasing order of validation accuracy and retain the best models

(5) Select hyperparameters 20% of the population. Modify hyperparameters by changing kernel size, filter size and dense units for the selected models within the specified ranges.

(6) To transfer the hyperparameters and weights from the top performing models to the low performing models in order to enhance the whole population.

(7) Repeat the training, evaluating and updating the models for a total of 10 generations

(8) Choose and return the model with the best validation accuracy from the last generation.

b) 1D PBCNN:

In the hybrid model, the 1D CNN serves as the core architecture for processing one-dimensional sequential data. The PBT algorithm is integrated with 1D CNN model in the training process to optimize its hyperparameters effectively. During the training process, PBT dynamically adjusts hyperparameters depending on the evaluation of different

models in the population, allowing the models to explore the hyperparameter space efficiently. This integration of PBT with the 1D CNN model results in faster convergence to optimal hyperparameter settings, ultimately leading to a more accurate and efficient model and its architecture is shown in Figure 4.

c) 1D PBCRNN (Population Based Convolutional Recurrent Neural Network)

In the hybrid model, the1D CRNN plays a central role in processing sequential data, enabling it to capture both local and long-range patterns present in the input. To efficiently optimize its performance, the Population-Based Training (PBT) algorithm is seamlessly integrated into the training process of the 1D CRNN model. PBT effectively manages a diverse population of 1D CRNN models, each with distinct hyperparameter configurations. During training, these models undergo evaluation on a validation set, and the best-performing models serve as a foundation for exploitation. Through this exploitation process, their hyperparameters are dynamically updated to potentially improve their performance. This cycle of evaluation, exploitation, and exploration continues over multiple epochs. By dynamically updating the hyperparameters based on the models' performance within the population, PBT allows the hybrid model to explore the hyperparameter space more efficiently. Consequently, this optimization approach empowers the hybrid 1D CRNN model to discover more effective hyperparameter configurations, ultimately leading to a more accurate and efficient model for processing sequential data. The architecture of 1D PBCRNN is shown in Figure 5.
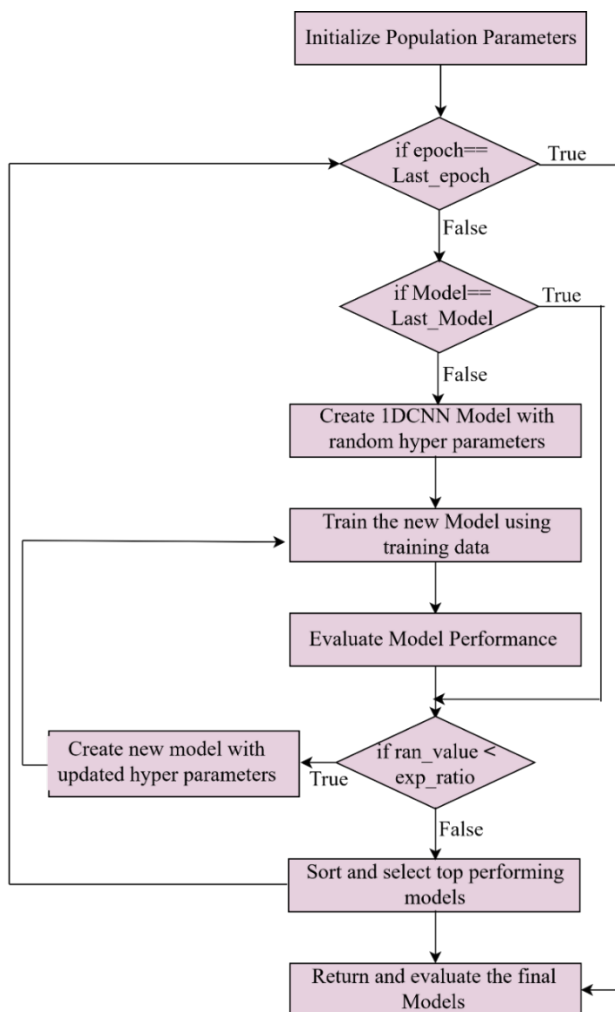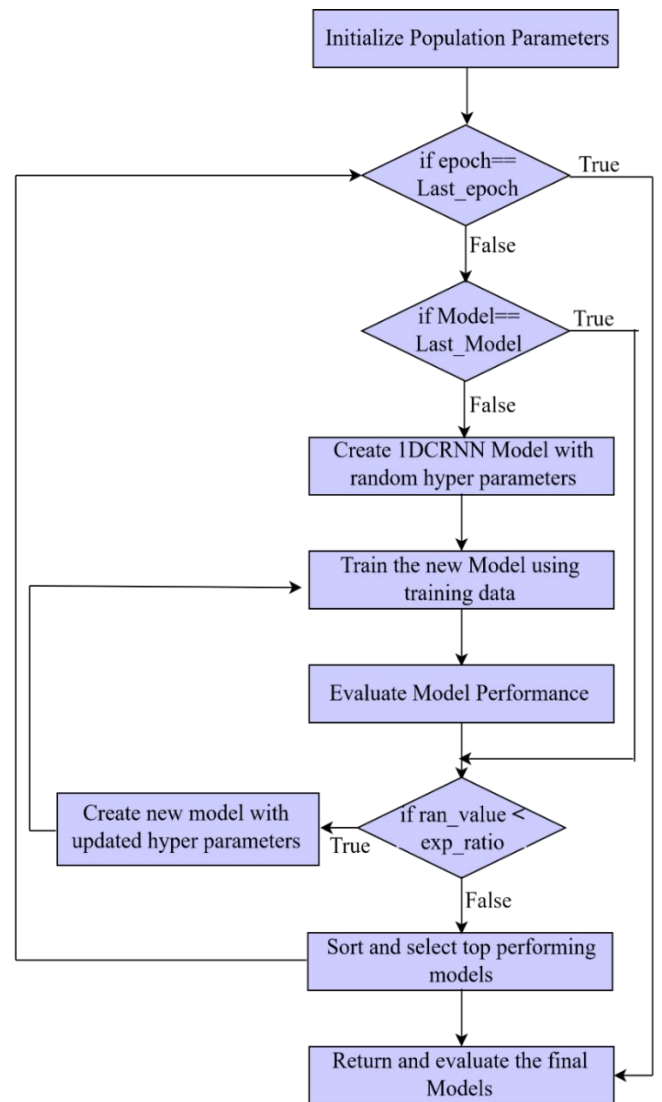

Figure 4. Architecture of 1D PBCNN

Figure 5. Architecture of 1D PBCRNN

4. EXPERIMENTAL RESULTS

The study employed EEG signals Datasets from the Isfahan MISP database, involving 61 participants aged 55, categorized into Healthy Control (29) and Mild Cognitive Impairment (32) groups. EEG recordings were made during morning sessions with closed eyes, utilizing a Galileo NT device with 19 electrodes (C3, Cz, C4, Fp1, Fp2, F7, F3, Fz, F4, F8, O1, O2, P3, Pz, P4, T3, T4, T5 and T6). Data was saved in EDF format [26, 27].

A 5-second epoch duration was employed with an overlap of 1 second. Each epoch contained N = 1280 (5 × 256) samples, resulting in a total of 28,402 input EEG data points, which included data from 14,757 MCI subjects and 13,645 HC subjects.

4.1 Performance metrics

Table 1. Confusion matrix

	MCI	HC
MCI	True Positive (TP)	False Positive (FP)
HC	False Negative (FP)	True Negative (TN)

Performance metrics are important because they allow us to

objectively assess the model's performance and make comparisons with other models or benchmarks function is shown in the Table 1.

- Accuracy: It is a performance metric that calculates the ratio of accurate predictions to the total number of predictions generated by a model. The Accuracy formula is shown in Eq. (5).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (5)$$

where, TP stands for true positives, TN represents true negatives, FP signifies false positives, and FN denotes false negatives.

- Precision: This metric is employed to assess the precision of positive predictions generated by a classification model. The precision calculation is shown in Eq. (6).

$$Precision = \frac{TP}{TP + FP} \qquad (6)$$

- Recall: It signifies the ratio of actual positive instances that the model correctly classifies as positive. Recall formula is shown in Eq. (7).

$$Recall = \frac{TP}{TP + FP} \qquad (7)$$

- F1 score: It provides a balance between precision and recall. F1 score calculation is shown in Eq. (8).

$$F1\ score = \frac{2 * Precision * Recall}{TP + FN} \qquad (8)$$

4.2 1D CNN for feature extraction and classification

The dataset mainly contains training and testing subsets using the "train_test_split" method with 75% and 25% respectively. Initially, the model extracts important patterns or features from the EEG data through convolutional and pooling layers. Then, it uses these learned features to classify the EEG data into control or patient groups through the dense layers. The model is constructed with the 'adam' optimizer, responsible for adapting the model's internal parameters during the training process to minimize the 'binary_crossentropy' loss function. The model's prediction accuracy against the real labels (0 or 1) is gauged by the loss function. The accuracy that we achieved using 1DCNN is 79.43%.

4.3 1D CRNN for feature extraction and classification

The input data is divided into training with 25% and testing 75%. This modified architecture (1D CRNN) incorporates LSTM layers alongside convolutional layers. The CNN part extracts features, and the LSTM part captures temporal dependencies in the EEG data. The model is configured with 'adam' as the optimizer and 'binary_crossentropy' as the loss function. As a result, it achieves an accuracy of 81.12%.

4.4 Time-domain features with 1D CNN

Time-domain features like Kurtosis, Zero Crossing Rate

(ZCR), and Hjorth parameters are very important in studying EEG signals. They capture key details that help to differentiate Healthy Controls (HC) and people with Mild Cognitive Impairment (MCI). When these features are used as inputs for different models—such as 1D CNN, 1D CRNN, 1D PBCNN and 1D PBCRNN—the results show how these features affect the accuracy of classification. The analysis compares how these features contribute are useful in detecting MCI.
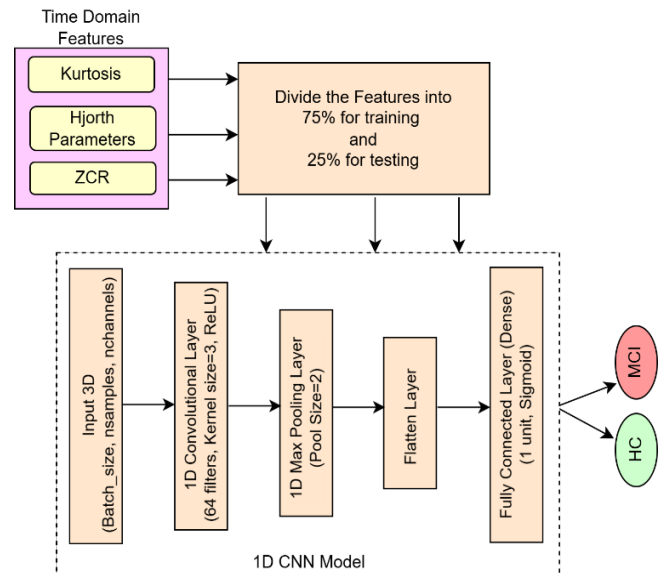

Figure 6. Utilizing time domain features as input for 1D CNN to classify HC and MCI classes

When each feature Kurtosis, ZCR, and Hjorth was tested separately with the 1D CNN, the Hjorth parameter had the highest validation accuracy of 89.70%, when compared to both Kurtosis and ZCR. This shows that Hjorth features, which measure mobility and complexity in EEG signals, are better at telling the difference between Healthy Controls (HC) and people with Mild Cognitive Impairment (MCI) when using the 1D CNN. Although Kurtosis and ZCR are useful, they don't capture enough changes or variability in the signals to perform as well as Hjorth features in this model. The time domain features giving as input to 1DCNN model is shown in Figure 6.

The major advantage of the model is its capability of effectively mining spatial features from the EEG data using the convolutional layers. Through recognizing both the local and global patterns, the 1D CNN is able to extract significant information from time-series features. Additionally, the 1D CNN works well with smaller datasets but the model cannot effectively capture long-term temporal patterns, which is difficult for analyzing EEG signals and detecting MCI. This limitation makes it unable to utilize the sequential information contained in the data.

4.5 Time-domain features with 1D CRNN

When the same time-domain features were used with 1D CRNN, similar results were seen. The Hjorth parameter once again had the best accuracy, reaching 88.29%, which was slightly lower than its performance with the 1D CNN. This shows that the CRNN can understand time-based patterns and find useful details in EEG signals. Although Kurtosis and ZCR performed reasonably well, they didn't capture the complexity of EEG signals as effectively as Hjorth, which likely affected

their classification accuracy. 1DCRNN Model with time domain features is shown in Figure 7.

This allows the model to study EEG signals more completely by capturing both spatial and time-based details, which are important for telling MCI apart from HC. However, adding recurrent layers makes the model more complex, which means it takes longer to train and needs more computer power. Also, the large number of parameters can lead to overfitting, especially when using smaller EEG datasets. Even with these challenges, CRNNs are useful for classifying EEG signals because they can analyze time-based patterns so well.

4.6 Time-domain features with 1D PBCNN

Using Population-Based Training (PBT) improved the models a lot by fine-tuning their hyperparameters to find the best settings. When the time-domain features were used with these optimized models, the results were as follows:

With PBT optimization, the Hjorth parameter showed better performance, reaching a validation accuracy of 88.39%. This small improvement compared to the 1D CNN model. This process tune hyperparameters helps the model do a better job of picking out important features and classifying EEG signals.

However, like the 1D CNN, the PBCNN cannot understand long-term patterns in the data since it doesn't model temporal dependencies. Another downside is that PBT requires a lot of computational power because multiple models need to be trained and tested at the same time. Even with these challenges, the PBCNN's ability to adjust its settings dynamically makes it a strong model for finding spatial features in EEG signals. The architecture of 1DPBCNN with time domain features is shown in Figure 8.
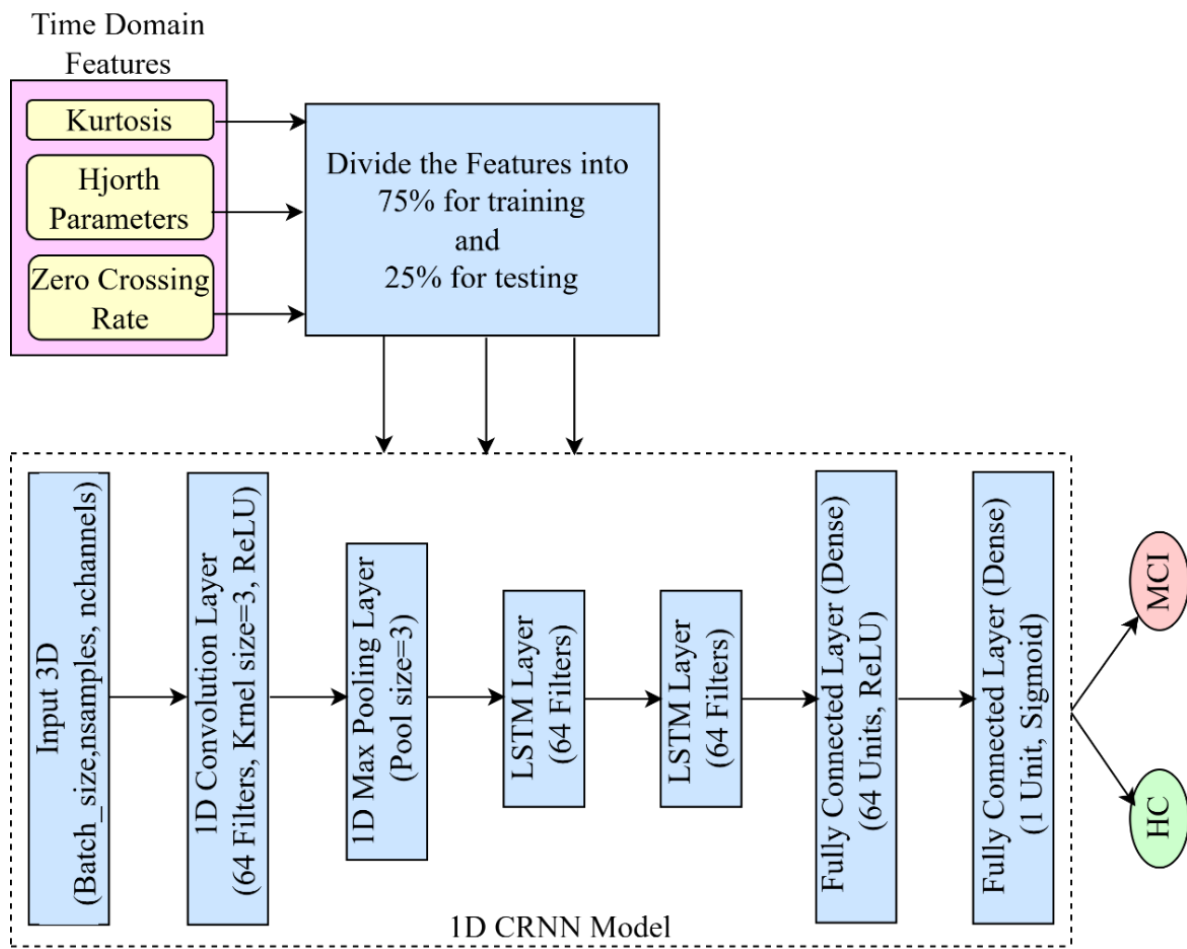

Figure 7. Utilizing time-domain features as input for 1D CRNN to classify HC and MCI classes
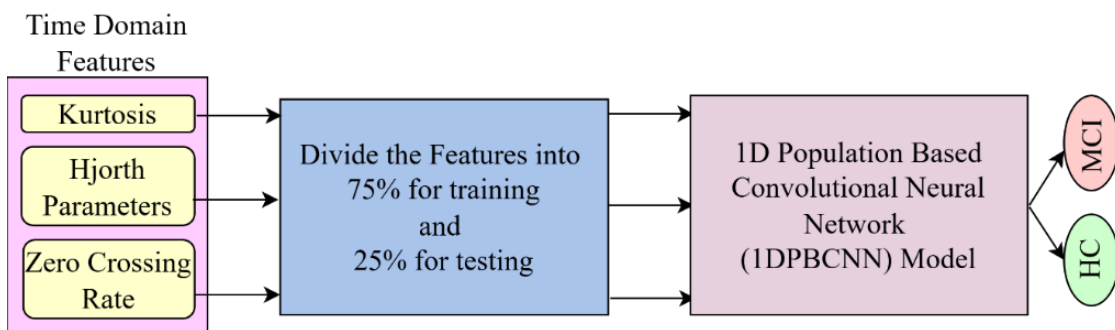

Figure 8. Utilizing time domain features as input for optimized 1D PBCNN to classify HC and MCI classes

4.7 Time-domain features with 1D PBCRNN

The optimized 1D PBCRNN model achieved the highest overall accuracy of 90.01% when using the Hjorth parameter as input. This result shows that the Hjorth parameter is very good at capturing detailed time-based patterns in EEG signals. The convolution layers focus on spatial features while the recurrent layers focus on long term patterns and Population Based Training which makes it effective for analyzing EEG signals. With PBT, the model's hyperparameters are adjusted during training to find the best settings. This process improves the model settings and leads for better performance. Although the PBCRNN performs the best, it requires a lot of computational power, taking more time and memory to train. The high complexity of the model also means it needs to be carefully controlled to avoid overfitting. Despite these challenges, the PBCRNN's ability to capture both spatial and temporal details makes it the most effective model for detecting MCI. The 1DPBCRNN with Time domain features is shown in Figure 9.
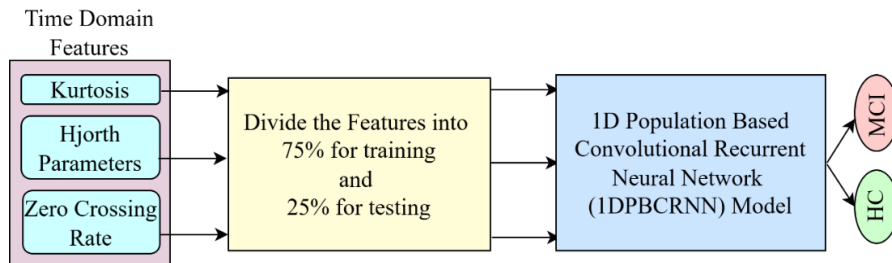

Figure 9. Utilizing time domain features as input for optimized 1D PBCRNN to classify HC and MCI classes
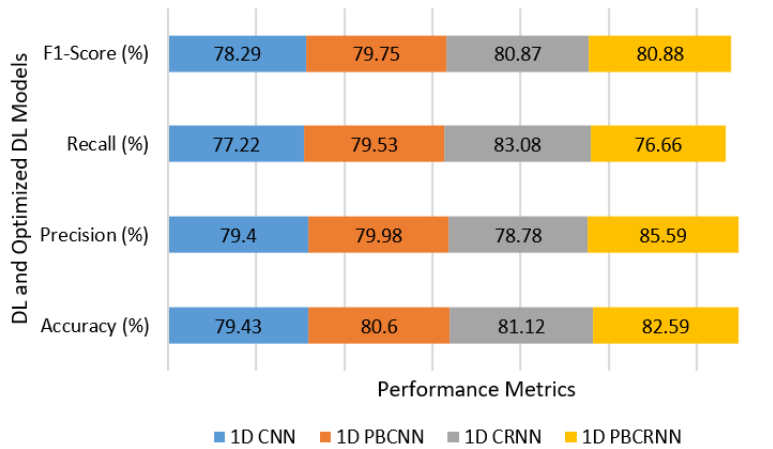

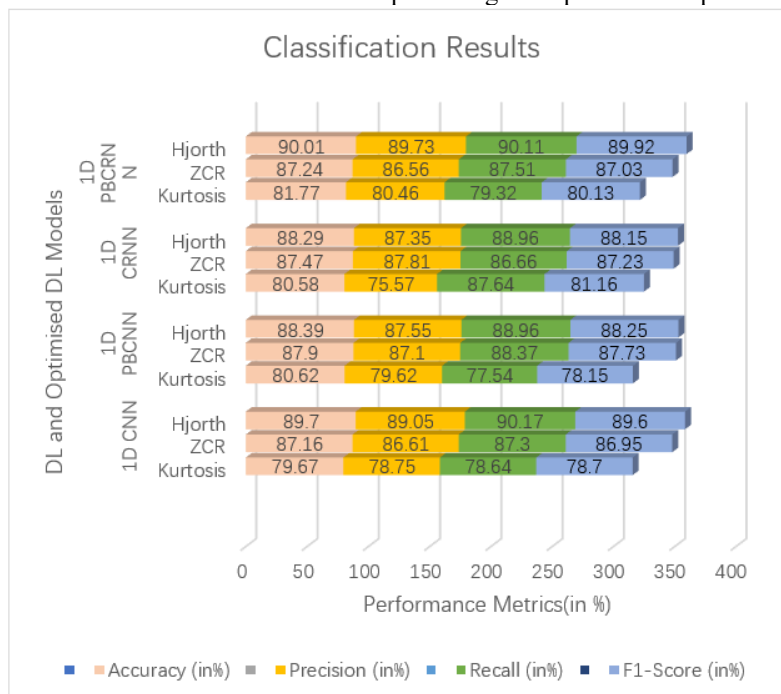**Figure 10.** Classification results for various deep learning and optimized deep learning techniques

Figure 11. Classification accuracy for various feature extraction and deep learning techniques

Table 2. Deep learning models result (IN %)

Deep Learning Models	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
1D CNN	79.43	79.40	77.22	78.29
1D PBCNN	80.60	79.98	79.5 3	79.75
1D CRNN	81.12	78.78	83.0 8	80.87
1D PBCRNN	82.59	85.59	76.6 6	80.88

Table 3. Different feature extraction techniques and deep learning models classification accuracy (IN %)

DL Models	Time Domain Features	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
1D CNN	Kurtosis	79.67	78.75	78.64	78.70
	ZCR	87.16	86.61	87.30	86.95
	Hjorth	89.70	89.05	90.17	89.60
1D PBCNN	Kurtosis	80.62	79.62	77.54	78.15
	ZCR	87.90	87.10	88.37	87.73
	Hjorth	88.39	87.55	88.96	88.25
1D CRNN	Kurtosis	80.58	75.57	87.64	81.16
	ZCR	87.47	87.81	86.66	87.23
	Hjorth	88.29	87.35	88.96	88.15
1D PBCRNN	Kurtosis	81.77	80.46	79.32	80.13
	ZCR	87.24	86.56	87.51	87.03
	Hjorth	90.01	89.73	90.11	89.92

Table 4. The comparison accuracy of existing models with our proposed model

Author	Features	Classifier	Accuracy (in %)
Khatun et al. [28]	Time-frequency Features	SVM	87.90
Morabito et al. [29]	-	Deep Learning on Convolutional Neural Networks (CNN)	80%
Our Proposed Method	Time domain Features	1DPBCRNN	90.01

The accuracy, precision, recall and F1-score results comparison for 1D CNN and 1DCRNN with 1D PBCNN and 1D PBCRNN is shown in Table 2 and the pie chart representation of these results is shown in Figure 10.

The classification results of various time domain features along with deep learning and optimized deep learning models is shown in Table 3. And its pie chart representation is shown in Figure 11.

The comparison of existing models with our proposed model are in Table 4.

5. CONCLUSION

The proposed methods 1DCNN, 1DCRNN, 1DPBCNN, 1D PBCRNN presents unique advantages and disadvantages based on its structure and complexity. The 1D CNN stands out for its simplicity and efficiency in spatial feature extraction, while the 1D CRNN introduces the ability to model temporal dependencies. The optimized models, 1D PBCNN and 1D PBCRNN, leverage PBT to dynamically tune hyperparameters, particularly the 1D PBCRNN model, have significant clinical application value in the early screening and diagnosis of Mild Cognitive Impairment (MCI). The high classification accuracy of 90.01 is achieved using 1D PBCRNN with Hjorth features proves that these methods have the capacity of offering accurate and non-invasive diagnostic assistance. Early detection of MCI is critical in clinical practice, as it allows timely intervention to delay or prevent the progression to Alzheimer's disease. In this study, EEG-based approaches are used and these are especially useful because they are affordable, readily available and can be transported to different locations while other expensive and inaccessible imaging methods like MRI or PET scans are not. The application of these sophisticated models integrated with clinical practices can help the neurologists and other healthcare personnel providers by offering objective, data-driven insights into cognitive function.

There are some limitations to the proposed methods. A significant problem is that they require a lot of data for training which can be overcome by using better data augmentation strategies or transfer learning. Combing EEG data with other datas like MRI or clinical biomarkers can help in accurate analysis. Another area of future research could also involve the enhancement of computational efficiency and also increasing the generality of these models for their effective application in the clinical setting as well.

REFERENCES

[1] Richardson, C., Stephan, B.C., Robinson, L., Brayne, C., Matthews, F.E., Cognitive Function and Ageing Study Collaboration. (2019). Two-decade change in prevalence of cognitive impairment in the UK. European Journal of Epidemiology, 34: 10851092.

[2] Batum, K., Çinar, N., ŞAHİN, Ş., Çakmak, M.A., Karşidağ, S. (2015). The connection between MCI and Alzheimer disease: Neurocognitive clues. Turkish Journal of Medical Sciences, 45(5): 1137-1140. https://doi.org/10.3906/sag-1404-179

[3] Light, G.A., Williams, L.E., Minow, F., Sprock, J., Rissling, A., Sharp, R., Swerdlow, N.R., Braff, D.L. (2010). Electroencephalography (EEG) and event-related potentials (ERPs) with human participants. Current Protocols in Neuroscience, 52(1): 6-25. https://doi.org/10.1002/0471142301.ns0625s52

[4] Teng, C.L., Zhang, Y.Y., Wang, W., Luo, Y.Y., Wang, G., Xu, J. (2021). A novel method based on combination

of independent component analysis and ensemble empirical mode decomposition for removing electrooculogram artifacts from multichannel electroencephalogram signals. Frontiers in Neuroscience, 15, 729403. https://doi.org/10.3389/fnins.2021.729403

[5] Jiang, X., Bian, G.B., Tian, Z. (2019). Removal of artifacts from EEG signals: a review. Sensors, 19(5): 987. https://doi.org/10.3390/s19050987

[6] Echegoyen, I., López-Sanz, D., Martínez, J.H., Maestú, F., Buldú, J.M. (2020). Permutation entropy and statistical complexity in mild cognitive impairment and Alzheimer's disease: An analysis based on frequency bands. Entropy, 22(1): 116. https://doi.org/10.3390/e22010116

[7] Amin, H.U., Malik, A.S., Ahmad, R.F., Badruddin, N., Kamel, N., Hussain, M., Chooi, W.T. (2015). Feature extraction and classification for EEG signals using wavelet transform and machine learning techniques. Australasian Physical & Engineering Sciences in Medicine, 38: 139-149. https://doi.org/10.1007/s13246-015-0333-x

[8] Ramírez-Arias, F.J., García-Guerrero, E.E., Tlelo-Cuautle, E., Colores-Vargas, J.M., García-Canseco, E., López-Bonilla, O.R., Inzunza-González, E. (2022). Evaluation of machine learning algorithms for classification of EEG signals. Technologies, 10(4): 79. https://doi.org/10.3390/technologies10040079

[9] Saeidi, M., Karwowski, W., Farahani, F.V., Fiok, K., Taiar, R., Hancock, P.A., Al-Juaid, A. (2021). Neural decoding of EEG signals with machine learning: A systematic review. Brain Sciences, 11(11): 1525. https://doi.org/10.3390/brainsci11111525

[10] Craik, A., He, Y., Contreras-Vidal, J.L. (2019). Deep learning for electroencephalogram (EEG) classification tasks: A review. Journal of Neural Engineering, 16(3): 031001. https://doi.org/10.1088/1741-2552/ab0ab5

[11] Sur, S., Sinha, V.K. (2009). Event-related potential: An overview. Industrial Psychiatry Journal, 18(1): 70-73. https://doi.org/10.4103/0972-6748.57865

[12] Al-Baddai, S., Al-Subari, K., Tomé, A., Volberg, G., Lang, E.W. (2015). A combined EMD—ICA analysis of simultaneously registered EEG-fMRI data. BMVA, 2015(2): 1-15.

[13] Prochazka, A., Kukal, J., Vysata, O. (2008). Wavelet transform use for feature extraction and EEG signal segments classification. In 2008 3rd International Symposium on Communications, Control and Signal Processing, Saint Julian's, Malta, pp. 719-722. https://doi.org/10.1109/ISCCSP.2008.4537317

[14] Glaser, J.I., Benjamin, A.S., Chowdhury, R.H., Perich, M.G., Miller, L.E., Kording, K.P. (2020). Machine learning for neural decoding. eNeuro, 7(4): ENEURO.0506-19. https://doi.org/10.1523/ENEURO.0506-19.2020

[15] Kotowski, K., Stapor, K., Ochab, J. (2020). Deep learning methods in electroencephalography. Machine Learning Paradigms: Advances in Deep Learning-based Technological Applications, 18: 191-212. https://doi.org/10.1007/978-3-030-49724-8_8

[16] Movahed, R.A., Hamedani, N.E., Sadredini, S.Z., Rezaeian, M.R. (2021). An Automated EEG-based mild cognitive impairment diagnosis framework using spectral and functional connectivity features. In 2021 28th National and 6th International Iranian Conference on Biomedical Engineering (ICBME), Tehran, Iran, pp. 271-275. https://doi.org/10.1109/ICBME54433.2021.9750291

[17] Timothy, L.T., Krishna, B.M., Menon, M.K., Nair, U. (2014). Permutation entropy analysis of EEG of mild cognitive impairment patients during memory activation task. In Fractals, Wavelets, and Their Applications: Contributions from the International Conference and Workshop on Fractals and Wavelets, 92: 395-406. https://doi.org/10.1007/978-3-319-08105-2_25

[18] Huang, Z., Wang, M. (2021). A review of electroencephalogram signal processing methods for brain-controlled robots. Cognitive Robotics, 1: 111-124. https://doi.org/10.1016/j.cogr.2021.07.001

[19] Sekhar, J.C., Domathoti, B., Santibanez Gonzalez, E.D. (2023). Prediction of battery remaining useful life using machine learning algorithms. Sustainability, 15(21): 15283. https://doi.org/10.3390/su152115283

[20] Javidi, S., Mandic, D., Cheong Took, C., Cichocki, A. (2011). Kurtosis based blind source extraction of complex noncircular signals with application in EEG artifact removal in real-time. Frontiers in Neuroscience, 5: 9396. https://doi.org/10.3389/fnins.2011.00105

[21] Wagh, K.P., Vasanth, K. (2022). Performance evaluation of multi-channel electroencephalogram signal (EEG) based time frequency analysis for human emotion recognition. Biomedical Signal Processing and Control, 78: 103966. https://doi.org/10.1016/j.bspc.2022.103966

[22] Staudinger, T., Polikar, R. (2011). Analysis of complexity based EEG features for the diagnosis of Alzheimer's disease. In 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Boston, MA, USA, pp. 2033-2036. https://doi.org/10.1109/IEMBS.2011.6090374

[23] Giudice, M.L., Varone, G., Ieracitano, C., Mammone, N., Bruna, A.R., Tomaselli, V., Morabito, F.C. (2020). 1D Convolutional Neural Network approach to classify voluntary eye blinks in EEG signals for BCI applications. In 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, pp. 1-7. https://doi.org/10.1109/IJCNN48605.2020.9207195

[24] Xu, G., Ren, T., Chen, Y., Che, W. (2020). A one-dimensional CNN-LSTM model for epileptic seizure recognition using EEG signal analysis. Frontiers in neuroscience, 14: 578126. https://doi.org/10.3389/fnins.2020.578126

[25] Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W.M., Donahue, J., Razavi, A., Kavukcuoglu, K. (2017). Population based training of neural networks. arXiv preprint arXiv:1711.09846. https://doi.org/10.48550/arXiv.1711.09846

[26] Kashefpoor, M., Rabbani, H., Barekatain, M. (2016). Automatic diagnosis of mild cognitive impairment using electroencephalogram spectral features. Journal of Medical Signals & Sensors, 6(1): 25-32.

[27] Kashefpoor, M., Rabbani, H., Barekatain, M. (2019). Supervised dictionary learning of EEG signals for mild cognitive impairment diagnosis. Biomedical Signal Processing and Control, 53: 101559. https://doi.org/10.1016/j.bspc.2019.101559

[28] Khatun, S., Morshed, B.I., Bidelman, G.M. (2017). Single channel EEG time-frequency features to detect Mild Cognitive Impairment. In 2017 IEEE International Symposium on Medical Measurements and Applications

(MeMeA), Rochester, MN, USA, pp. 437-442. https://doi.org/10.1109/MeMeA.2017.7985916

[29] Morabito, F.C., Campolo, M., Ieracitano, C., Ebadi, J.M., Bonanno, L., Bramanti, A., Bramanti, P. (2016). Deep convolutional neural networks for classification of mild cognitive impaired and Alzheimer's disease patients from scalp EEG recordings. In 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), Bologna, Italy, pp. 1-6. https://doi.org/10.1109/RTSI.2016.7740576