RESEARCH PAPER

A Parameter Optimization of Ensemble Deep Alzheimer's Disease Detection Network using Red-Crowned Crane Optimization Algorithm

Karpagam[1], Rosiline Jeetha[2]
[1]Department of Computer Science, Dr.N.G.P.Arts and Science College, Coimbatore, India
[2]Computer Science with AI, Nirmala College for Women, Coimbatore, India

How to cite this article: Karpagam, Jeetha R. A Parameter Optimization of Ensemble Deep Alzheimer's Disease Detection Network using Red-Crowned Crane Optimization Algorithm. Int J Drug Deliv Technol. 2026;16(36s): 390-402. DOI: 10.25258/ijddt.16.36s.46

Abstract— **Timely identification of Alzheimer's Disease (AD) facilitates interventions that may delay the disease advancements. In order to detect structural and functional changes associated with AD, MRI provides high-resolution pictures of the brain. But, its interpretation was challenging as it fails to access the high-dimensional data. Several Deep Learning (DL) models have been created to anticipate AD in its initial stages. Amongst, Enhanced Ensemble Brain Connected AD Network (EEBCADnet) was developed. This model combines sMRI (sMRI)features extracted using VGGNet and One Dimensional Convolutional Neural Network (1D-CNN) with functional connectivity (FC) features derived through graph theory from functional MRI (fMRI) for AD classification. But, the hyperparameter of VGGNet and 1DCNN were not optimized effectively properly leading to lower prediction accuracy and increased computational complexity. Hence, this paper, Red-Crowned Crane Optimization (RCCO) alogrithm is suggested to fine-tune the hyperparameters of VGGNet and 1DCNN in EEBCADnet for efficient AD prediction. The RCCO algorithm is inspired by the adaptive behaviors of red-crowned cranes, efficiently explores the hyperparameter search space, balancing exploration and exploitation to converge on the optimal configurations. By applying RCCO, the hyperparameters of VGGNet and 1DCNN are optimized both globally (jointly across both modules) and locally (for each network individually) to better model performance. The proposed RCCO-based hyperparameter tuning significantly enhances the diagnostic accuracy and lowers the computational complexity of EEBCADnet model for early AD detection. The complete work is termed as Optimized EEBCADnet (OEEBCADnet). The test results reveals that the proposed model achieves 99.17% and 99.03% of accuracy on Kaggle AD dataset and AD Neuroimaging Initiative dataset (ADNI) surpassing other existing models.**

Keywords*— *Alzheimer's Disease, Magnetic Resonance Imaging, Deep Learning, Red-Crowned Crane Optimization, Hyper parameter Tuning

I. INTRODUCTION

AD is a neurological illness that worsens with time, causing damage to nerve cells and a decrease in cognitive function as β-amyloid plaques and neurofibrillary tangles increase in the brain [1]. AD primarily affects older adults with its occurrence increasing significantly with their perspective age. The diseases severely impact the key brain regions like hippocampus plays a vital role in forming memories; the entorhinal cortex involved in memory and spatial navigation; and the cerebral cortex that responsible for higher cognitive skills like language and reasoning [2]. Damage in these areas results in symptoms like memory impairment, diminished thinking capacity and changes in behavior and mood [3].

In all, over 55 million individuals throughout the world are living with dementia, which includes AD. By 2050, experts predict that number might rise to 152.8 million [4]. AD prevalence is higher in Europe and North America compared to South America, Asia and Africa while countries like Japan and South Korea experiences a rapid increase due to population aging which poses a considerable difficulty for the health care system [5]. Globally, dementia is a major public health concern, and the World Health Organization (WHO) has urged governments to address this issue [6]. As a result, early diagnosis of changes in these areas is critical for successful intervention and treatment.

Several medicinal therapies may help to reduce symptoms in the early AD stages. Making a diagnosis involves imaging tests, mental state evaluations, neurological and physical testing and a medical history [7]. Mapping and computational technologies such as MRI, electroencephalography (EEG), and genotyping will allow for trustworthy detection. Brain Computed Tomography (CT) [8]. Neuropsychologists are able to identify the early stages of AD and begin treatment at an earlier stage because to these technologies that have revealed signs of brain shrinkage [9].

Among these imaging models, MRI provides best results in the prediction of AD [10]. MRI has the ability to detect the brain atrophy particularly in the regions of hippocampus and entorhinal cortex which are affected by the diseases [11]. It also assists in monitoring the AD diseases progression by providing high image resolutions that highlight each structural modifications over time [12]. But, MRI fails to access the high-dimensional data it generated and challenging to analyze manually due to its complexity and size. The extensive dimension of neuroimaging data is another challenge for conventional approaches, which frequently necessitate substantial human feature engineering. Also, some models might struggle to generalize across different datasets and may not always capture subtle or complex patterns in the data.

DL models are applied to address these limitations by automating feature extraction and identifying the subtle patterns indicative of AD [13]. DL models excels at analyzing large dataset and learning features from MRI images significantly advancing early AD prediction which improvises diagnostic accuracy and supports personalized treatment strategies [14].

Some examples of DL models like Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTMs), Deep Belief Networks (DBNs), Recurrent Neural Networks (RNNs) and variations on these models [15]. There are a number of DL models that have been created to forecast AD.

In order to speed up AD detection, for example, DenseNet-121 was integrated in tandem with LeNet and AlexNet [16]. But the gradient losses let this model down. Using MRI scans of the brain, the ResNet-50 model was run to identify AD [17]. However, gradient disappearance is a real possibility, which can make gradient descent less than ideal. A Ensemble (ECNN) [18] was trained to classify AD based on brain anatomy using Line Segment Feature Analysis (LFA), VGGnet, and a 1DCNN model. VGGnet model developed broad antigens from raw MRI data, while LFA retrieved the key morphological features. A 1DCNN-based approach to extracting local biomarkers from visual data was an improvement over previous methods for AD detection using vector signals. However, the ECNN model mainly relies on sMRI and LFA-based shape features, limiting its ability to capture comprehensive brain features. Since MRI reflects only anatomical structure, functional brain activity related to AD may be overlooked. When supplementary data is missing, the model may struggle to capture minute brain interactions, which in turn lowers its classification accuracy.

To tackle this issues, an EEBCADnet was developed which integrates sMRI and fMRI data was created to overcome the above concerns. Primarily, the suggested technique generates 360 ROIs from pre-processed fMRI data which has been segmented using the Glasser atlas. To record both local and global network measurements, FC matrices were generated by estimating the correlations among ROI time series. Graph-theoretical feature extraction was then used to round out the process. In parallel, sMRI data are processed using VGGNet to extract deep structural representations, while LFA captures feature points and line segments and these features are further refined using a 1D-CNN to learn discriminative local patterns. We apply a concatenation method to create a fused feature vector that combines the features of fMRI and sMRI. The final step in AD prediction is a fully linked layer, which is tracked by a softmax layer. But, the hyperparameters of 1DCNN and VGGNet of EEBCADnet model were not fine-tuned properly which leads to suboptimal performance and higher computational complexity. This might hinder the model's ability to fully leverage the extracted features for accurate AD prediction. The accuracy and efficiency of the model could be even improved with proper adjustment of the hyperparameters.

In this paper, an OEEBCADnet model is developed to enhance the diagnostic accuracy and minimize the computational complexity in AD prediction. This model employs a RCCO-based hyperparameter tuning strategy that jointly and separately optimizes critical parameters across the 1D-CNN and VGGNet components. The jointly optimized (global) hyperparameters which affect both modules simultaneously include learning rate, epochs, batch size, optimizer type, activation function, momentum, weight decay, dropout rate, learning rate scheduler, early stopping criteria and feature fusion weight. Module-specific (separate) hyperparameters will fine-tune individually on each network include freeze layers for VGGNet and kernel size or pooling type for 1D-CNN Inspired by the cooperative and adaptive actions of red-crowned hoists like searching dispersion, danger evasion, roosting aggregation and courtship displays, the model effectively explores the hyperparameter search space, converges toward the most promising configurations, balances exploration and exploitation and avoids local stagnation leading to optimal joint and module-level tuning of the 1D-CNN and VGGNet parameters [19]. This combination of joint and separate optimization ensures synergistic feature learning, faster convergence and improved classification performance. Consequently, OEEBCADnet achieves superior AD prediction accuracy and lower computational overhead.

This paper's parts are constructed as follows: Previous research is included in Part 2. The suggested model is explained in Part 3, and validation outcomes are shown in Part 4. This study is clarified and possible enhancements are suggested in Part 5.

II. LITERATURE SURVEY

A 3D-CNN structure was developed for AD forecasting using MRI data in neuroimaging genetics [20]. This approach, T1-weighted MRI images were obtained with volumetric 3D longitudinal Magnetization Prepared Rapid Gradient Echo Imaging (MPRAGE) with different resolutions. Then, the Gradwrap, B1 variation modification, and N3 adjustment models were used for image pre-processing. Brain Extraction Tools (BET) was employed to retrieve useable cerebral images. FMRIB Automated Segmentation Tool (FAST) tool was employed for image segmentation. 3DCNN was applied to extract the features and classify the AD stages. However, large prevalence of aberrations could have reduced the trustworthiness of specific form attributes.

A hybrid deep learning model was suggested to detect AD using Deep Convolutional Generative Adversarial Network (DCGAN) and Deep CNN (DCNN) models [21]. The collected MRI images was pre-processed and the data was generated using DCGAN model. The image resolution was improvised using Super Resolution Generative Adversarial Networks (SRGAN). CNN was applied for feature extraction and FC layer in CNN was applied for AD prediction. But, this model results with overfitting and biased learning diminishes the hypothesis's steady training effectiveness.

An effective method for AD diagnosis was devised using an improved Xception simulation integrated with a snapshot ensemble approach [22]. This solution employed an augmented Xception architecture to generate varied snapshots, offering distinct insights into MRI characteristics.A option-level integration technique was implemented by integrating selection results using a random forest meta-learner through a mixing mechanism. Finally, ensemble methodology was applied for AD prediction. But, this model has high training time and intensive computational resources. A standardized volumetric CNN was constructed for AD identification using T1-weighted MRI images [23].

In this method, pre-processed techniques such as N4 modification, skull removal, versus rigorous enrollment were employed on MRI images to obtain specific for AD and normal

images. Then, procedures for augmentation such as rotations and rescaling were employed to improve the models accuracy. Finally, ConvNet model was employed to extract and detect the features for the AD classification. But, the models hyper-parameter needs to be optimized that severely affects the overall model performance's.

A Two-stage Multimodal Data Fusion Integrated Incremental Learner Ensemble Classifier (TMDFILE) was developed for AD prediction [24]. This model combines temporal, spatial, spectral, audio and text data modalities utilising a gating mechanism to optimise the contribution of each modality. Incremental learning was employed to adjust evolving data patterns and enhance long-term performance in AD prediction. But, this model relies on multiple data which struggles for appropriate data interpretation.

An AD forecasting technique was created using Fisher Mantis Optimization and hybrid deep learning frameworks [25]. Initially, MRI images underwent preprocessing involving standardization and noise removal. Feature extraction utilized texture characteristics derived via the Gray-Level Co-occurrence Matrix (GLCM) and spatial data obtained through a pretrained VGG-16 network. Fisher Mantis Optimization (FMO) was utilized for optimum choice of attributes. The attributes chosen were categorized utilizing a CNN-LSTM model, which captured both spatial and temporal trends. However, the model trained on the limited dataset leads to potential overfitting and lower accuracy.

An Alzheimer Recognition Ensemble Network (ALZENET) was proposed for MRI-based classification [26]. To rectify class disparity, they utilized Synthetic Minority Over-Sampling Technique (SMOTE). The model utilized a multi-layer CNN built from scratch, combined with an ensemble of VGG-16, Inception v3, and ResNet-50 architectures to accurately classify dementia stages. But, complexity of the ensemble model might result in high computational costs and longer training times, limiting its applicability in real-time clinical environments.

An Exponential Gaussian Error Linear Unit–Squeeze Net (EGELU-SZN) methodology was constructed for the premature identification and prediction of AD [27]. The skull stripping was performed by Adaptive Median Otsu's Thresholding (AMOT), segmentation by Rectilinear Mayfly Optimization-centric Automatic Seeded Region Growing (RMF-ASRG) algorithm, attribute extraction and decrease in feature via Reflective Correlation Principal Component Analysis (RCPCA). Ultimately, EGELU-SZN was applied for AD classification. But, the reliance on multiple complex algorithms might increase computational complexity and reduce the model's efficiency in real-time applications.

An AD prediction method was presented using Harris Hawks Optimization (HHO) and a MultiLayer Perceptron (MLP)–LSTM hybrid network [28]. In this model, CNN and Gray Level Co-occurrence Matrix (GLCM) was applied to obtain traits from MRI images. HHO was utilized to identify particularly signification characteristics. The selected features have been trained using MLP and LSTM to classify tumors as malignant or benign. However, this model was easy prone noisy images and has high generazability error.

A CNN–Swin Transformer (CNN-ST) model was developed for AD prediction [29]. In this model, Deep Convolution Generative Adversarial Network (DCGAN) was used to generate synthetic MRI slices to enhance class representation. The EffSwin-XNet model combines EfficientNet-B0 and the ST for both local and global feature extraction from MRI brain images. Feature fusion attention mechanism was used to highlight discriminative features. Grad-CAM was employed for explain ability, visualizing the brain areas that influence classification decisions. But, this model was evaluated on a limited dataset, resulting in reduced generalizability due to endless configurations.

III. PROPOSED METHODOLOGY

In this section, the proposed OEEBCADnet model is proposed for efficient AD prediction. Fig. 1 illustrates the structure of the proposed OEEBCADnet framework.

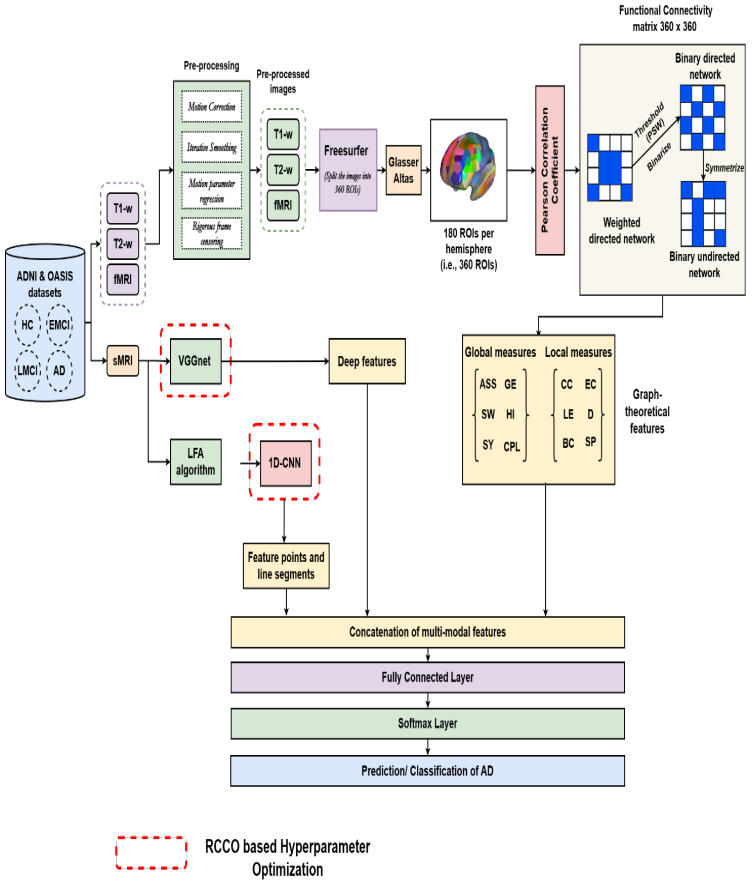

Fig. 1. Pipeline of the Proposed Model

A. Hyperparameter Optimization of EEBCADnet using RCCO

In this section, the complete illustration of the RCCO model optimizing the EEBCADnet for efficient AD prediction is provided.

*1) Behavioural Anatomy:*Red-crowned cranes is sizable wading birds belonging to the Gruidae group, part of the Gruiformes

order, distinguished by the red crowns atop their heads. They typically stand 1.5 to 1.6 meters tall, with a body ranging from of 1 to 1.5 meters, a wingspan of 2.2 to 2.5 meters, that weight ranging from 5 to 10.5 kg. Male and female cranes have nearly identical white coats. Their heads are unadorned and crimson, while their chins, throat, and neck predominantly exhibit a dark brown hue. The main plane feathers and body plumes are predominantly white, while supplementary and intermediate a plane wings are black. The long and curved primary wings cover the tail, which appear black when the crane is standing.

Red-crowned cranes epitomize joy, durability, and fidelity, but they are now among the rarest crane species. They are enumerated in the International Union for Conservation of Nature (IUCN) Red List of Threatened Species and are protected in China. These cranes are primarily found in China, Japan, Korea, Mongolia, and Russia. In Japan, they are non-migratory, while those in other countries migrate seasonally. Typically, they depart their wintering habitats in February or March to breed in places like the Russian Far East and Heilongjiang, China and return in September or October to winter in Korea and east-central China. During migration, they often form large groups of 40–50 peoples and sometimes over 100. Subsequent to immigration, they reside in smaller family categories.

Red-crowned cranes forage in pairs, alone, or occasionally in small groups. Their diet includes fruits, tadpoles, and the roots, foliage, and clams of marine life. At night, they congregate in familial groups within reed beds or shoals adjacent to water. While foraging or resting, they remain highly alert, watching for danger and taking flight with loud calls when threatened. Courtship begins in late March or early April, with the male crane raising its head, singing, and spreading its wings. The female responds, and the two engage in synchronized singing, jumping, and dancing. Their dance involves graceful movements, such as neck stretches, head raises, knee bending and leaping.

This study derives inspiration from the feeding, fleeing, roosting, and dancing behaviors of red-crowned cranes to develop the RCCO methodology. The subsequent part provides a detailed analysis and modelling of these behaviors.

*2) Population Initialization:*The RCCO algorithm, developed for this study, is an unique swarm-based metaheuristic optimization technique for optimizing the hyperparameters of the EEBCADnet model's VGGNet and 1DCNN layers. This algorithm simulates the cooperative and adaptive living behaviors of red-crowned cranes, where each crane's position corresponds to a candidate set of hyperparameter values, and by applying RCCO, the hyperparameters of VGGNet and 1D-CNN are optimized both globally (jointly across both modules) and locally (for each network individually) to enhance overall model performance.

Similar to other swarm intelligence algorithms, RCCO begins by randomly initializing a population of cranes (X < sub > cranes </sub >) within defined upper and lower bounds of each hyperparameter, as expressed using (1).

$$X_{cranes} = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,d} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,d} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{n,1} & \cdots & x_{n,j} & \cdots & x_{n,d} \end{bmatrix} \quad (1)$$

$$x_{i,j} = lb_b + r_0.\left(ub_j - lb_j\right), i = 1,2,..,n; j = 1,2,..,d \quad (2)$$

Here, X_i stands for the hyperparameter vector of the i^{th}crane, $x_{i,j}$for the j^{th}dimension of the i^{th}red-crowned crane, n for the number of red-crowned cranes, d for the number of decision variables, and $r_0$for the random value between 0 and 1. The upper and lower bounds of the j^{th}hyperparameter are represented as ub_j and lb_j, respectively.The quality of each candidate hyperparameter configuration is evaluated using the objective fitness function $F(X)$ which measures the classification accuracy of EEBCADnet trained under specific set of hyperparameter values generated by the RCCO alogrithm. The fitness function gives in (3):

$$F_{cranes} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_n) \end{bmatrix} = \begin{bmatrix} F\left(x_{1,1},..,x_{1,j},\dots,x_{1,d}\right) \\ \vdots \\ F\left(x_{i,1},..,x_{i,j},\dots,x_{i,d}\right) \\ \vdots \\ F\left(x_{n,1},..,x_{n,j},\dots,x_{n,d}\right) \end{bmatrix} \quad (3)$$

At this point, $F(X_i)$stands for the fitness score (for instance, validation accuracy) that corresponds to the i^{th}set of hyperparameters. During iterations, the cranes update their positions based on exploration–exploitation dynamics inspired by foraging and cooperative movement patterns, allowing the RCCO algorithm to discover the optimal joint and module-specific hyperparameters for the VGGNet and 1D-CNN components of EEBCADnet.

3) Mathematical model of RCCO for optimizing the hyperparameters: The RCCO algorithm incorporates position update equations that mimic clever methods exhibited by red-crowned cranes in their group behaviors, such as foraging, avoiding danger, acquiring for roosting, and crane dancing. These behaviors are mathematically modelled to guide the search process for optimal hyperparameter configurations of the EEBCADnet model. The main behavioral rules are described as follows:

Dispersing for Foraging: At the start of each iteration, the best hyperparameter set discovered so far is considered the ideal configuration (best habitat). As the optimization process proceeds, the cranes (search agents) disperse to explore new candidate hyperparameter combinations in search of better performance (higher classification accuracy). Two types of cranes are involved: Random foragers which explore locally around the current best solution to perform fine-tuning of hyperparameters such as learning rate and dropout rate. Long-

distance foragers perform global exploration, testing significantly different combinations of hyperparameters (e.g., optimizer type, kernel size) to avoid premature convergence.

Avoiding Danger: Cranes performing global exploration (long-distance foragers) are more prone to danger, symbolizing ineffective or unstable configurations that cause poor convergence or overfitting. When a crane encounters such a condition (low fitness value), it quickly alters its search direction, mimicking the escape response of red-crowned cranes. This technique avoids hyperparameter tuning blocks and increases population variety in the RCCO algorithm.

Gathering for Roosting: When a crane identifies a new and superior hyperparameter configuration with higher fitness (accuracy), it becomes the new habitat for the population. Other cranes (solutions) gradually move toward this configuration, reinforcing exploitation around the most promising search region. This collective movement allows RCCO to converge efficiently toward optimal hyperparameter values that balance accuracy, generalization, and computational cost.

Crane Dance: With a certain probability, two cranes with the highest fitness values (best two hyperparameter sets) are selected to perform a crane dance, symbolizing a pairwise knowledge exchange. During this process, their parameter combinations are partially recombined to produce new candidate solutions. This strategy accelerates convergence by combining beneficial characteristics of the top-performing configurations, so making EEBCADnet run better overall.

The RCCO algorithm achieves a good balance between exploration and utilization by incorporating these physically inspired procedures. This dynamic adaptation enables efficient optimization of both global (joint) and module-specific (local) hyperparameters of VGGNet and 1D-CNN, leading to improved diagnostic accuracy and reduced computational complexity for AD prediction.

*4) Principles Based on Foraging and Roosting Behaviours:*A specific proportion of red-crowned cranes are classified as long-distance foragers or random foragers based on the fitness values derived from the objective function $F(X)$ .This proportion indicates the EEBCADnet model's classification accuracy. For this task, we sort all possible sets of hyperparameters, or cranes, by fitness. The best-fitting configurations, or random foragers, are at the top of the list, while the worst-fitting ones, or long-distance foragers, are at the bottom. In this context, cranes with lower fitness values correspond to unstable or inefficient hyperparameter combinations that are far from optimal, and therefore, they are given the ability to explore wider regions of the search space to identify better solutions.To illustrate the locations of the random foragers $\left(X_{rf}\right)$ and long-distance foragers $\left(X_{lf}\right)$, consider the following in (4).

$$X_{rf} = \begin{bmatrix} X_{F1} \\ \vdots \\ X_{FK} \end{bmatrix} = \begin{bmatrix} x_{F1,1} & \cdots & x_{F1,j} & \cdots & x_{F1,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{Fk,1} & \cdots & x_{Fk,j} & \cdots & x_{Fk,d} \end{bmatrix} \quad (4)$$

$$X_{lf} = \begin{bmatrix} X_{F(k+1)} \\ \vdots \\ X_{Fn} \end{bmatrix} = \begin{bmatrix} x_{F(k+1),1} & \cdots & x_{F(k+1),j} & \cdots & x_{F(k+1),d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{Fn,1} & \cdots & x_{Fn,j} & \cdots & x_{Fn,d} \end{bmatrix} \quad (5)$$

The first k crane locations after sorting represent strong candidate hyperparameter sets, denoted as $X_{F1}, \ldots, X_{Fk}$, whereas the remaining weaker candidates are represented by $x_{F(k+1)}, \ldots, x_{Fn}$.When hyperparameter values are updated, the new positions reached by long-distance foragers and random foragers are given in (7) and (6), respectively.

$$X'_i\,(t+1) = X_i\,(t) + c_1\,.R.\left(X_{home}(t) - X_i(t)\right), i = F1, \ldots, Fk \quad (6)$$

$$X'_i\,(t+1) = X_i\,(t) + c_2\,.\left(X_{home}(t) - X_i(t)\right), i = F(k+1), \ldots, Fn, \quad (7)$$

The current hyperparameter position of the i^{th} crane is represented by $\mathrm{X_i(t)}$, whereas the position of the ideal habitat acquired after i iterations is represented by $\mathrm{X_{home}(t)}$. This indicates the best position that the entire population of red-crowned cranes has reached after t iterations. The update strength is controlled by the step adjustments coefficients $\mathrm{c_1}$and $\mathrm{c_2}$, and R is a $1 \times \mathrm{d}$ random matrix with entries that are random numbers in the range [0,1]. Here, we use $\mathrm{c_1} = 2$ to represent random foragers' local exploration, and we use $\mathrm{c_2} = 5 - 4\mathrm{t}/\mathrm{t_{max}}$, where $\mathrm{t_{max}}$is the maximum number of iterations, to show that $\mathrm{c_2}$ drops linearly from 5 to 1.This ensures the long-distance foragers perform boarder exploration during early iterations and gradually focus on exploitation in later iterations allowing the algorithm to efficiently identify the high-performing hyperparameter configurations for both VGGNet and 1DCNN modules.

In the early stages, long-distance foragers explore the wide regions of the hyperparameter search space, maintaining the strong population diversity while random foragers fine-tune parameters around promising solutions. As iteration progress, the population gradually converges towards the most effective configurations, while long-distance foragers approach the current best positions. With each iteration, a danger avoidance mechanism is implemented to prevent the system from being stuck, the probability of encountering danger i.e., convergence to a local optimum increases prompting the long distance foragers to relocate to new positions.

A random risk coefficient $c_r \in [0,1]$ is assigned to each long-distance forager when $c_r < (t/t_{max})^{1/2}$, the crane updates its position as follows as in (8).

$$X'_i\,(t+1) = X'_i\,(t+1) + r_1.\left(X_{rand} - X'_i\,(t+1)\right) + r_2.\quad \left(X_{ipbest} - X'_i\,(t+1)\right), i = F(k=1), \ldots, Fn \quad (8)$$

In this case, X_{rand} represents the randomly produced hypermeter setup, X_{ipbest} represents the best configuration previously identified by the same crane, $r_1, r_2 \in [1,2]$ are

random coefficients. This mechanism effectively enhances the exploration in later iterations preventing premature convergence and enabling RCCO to continue searching for better hyperparameters sets. When the search reaches nightime (end of an iteration), the cranes perform gathering for roosting, where the best-performing configuration discovered so far becomes the new habitat. All cranes, including both random and long-stance foragers move toward the updated optimal configuration according in (9).

$$X_i\,(t+1) = X'_i\,(t+1) + c_3.r_3.(X'_{home}(t+1) - X'_i\,(t+1),) \quad (9)$$

Where $X'_{home}(t+1)$ stands for the optimal habitat at the present moment (whether it has been updated or not), $c_3 = 2 - t/t_{max}$ is the time-varying step coefficient, and $r_3 \in [0,1]$ indicates a random scalar.The gradual reduction of c_3 ensures that the cranes (hyperparameter sets) move closer to the optimal region as iterations proceed balancing exploration and exploitation. Through, this adaptive search mechanism, RCCO efficiently refines both global and module specific hyperparameters accelerating convergence and improving the diagnostic performance of the EEBCADnet model.

5) Tactics Based on Crane Dance: Based on the red-crowned cranes' dancing behaviour, this section offers an updating technique. Here, a male and female crane representing the two hyper parameter configurations with the highest fitness values successfully pair up and communicate with each other. This causes other red-crowned cranes to congregate around them so they can see their dance. Imagine a couple of red-crowned cranes as searching agents with the highest and lowest fitness values, respectively. Next, the following model represents the positional update formula that is applied to update the sets of candidate hyper parameters:

$$X'_i(t+1) = X_i(t) + u.\left(r_4.X_{first}(t) - X_i\,(t)\right) \quad (10)$$

$$X''_i(t+1) = X_i(t) + u.\left(r_4.X_{second}(t) - X_i\,(t)\right) \quad (11)$$

$$X_i(t+1) = \frac{X'_i(t+1) + X''_i(t+1)}{2} \quad (12)$$

The variance in hyper parameter updates is controlled by the deviation coefficient, r_4, which is a random value between 0 and 0.1. Following t iterations, X_{first} represents the optimal global hyperparameter configuration and X_{second} the second-best.we see that u is a random number drawn from a gaussian distribution in (13).

$$u \sim N(1, \sigma_u^2), \sigma_u = 1 - \frac{t}{t_{max}} \quad (13)$$

This method involves incorporating the best-performing parameter value into subsequent iterations of the red-crowned cranes by updating their positions according to the top two configurations. The purpose of the deviation coefficient r_4 is to offer controlled variability to the hyper parameter updates and hence prevent the method from being over-exploited throughout iterations. According to Equation (13), under the following circumstances, the degree of dispersion of the Gaussian distribution is adaptively adjusted:

- In early iterations, larger σ_u values allow more diverse hyperparameter exploration (global search).
- In later iterations, σ_u gradually decreases, and u approaches 1, promoting intensive local search (exploitation) around the best hyperparameter settings found so far.

Hyperparameter settings both globally and locally are dynamically updated through this crane dance mechanism which combines the strengths of the two best solutions to discover superior hyperparameter configurations of VGGNet and 1DCNN efficiently.

6) Adaptive Position Update Strategy in RCCO: The RCCO algorithm iteratively selects a position update approach from a set of mechanisms modelled after crane dance and foraging-roosting behaviours. A constant value between 0.1 and 0.9, known as the probability coefficient (P_c),governs the selection. In every iteration, a random integer ($r_5 \in [0,1]$) is produced. If ($r_5 < P_c$), the position update strategy that is based on foraging and roosting behaviours is used. On the other hand, the position update strategy that corresponds to crane dance is used if ($r_5 \geq P_c$).By using probabilistic selection, RCCO is able to guarantee both border search and refinement around solid configurations during the hyper parameter optimization process, dynamically balancing the discovery and utilization phases.

The value of P_c significantly influences the search dynamics of the algorithm. A smaller P_c emphasizes exploitation, allowing the algorithm to perform intensive local searches around promising hyperparameter configurations of the EEBCADnet model, while a larger P_c promotes global exploration, encouraging the discovery of new and diverse hyperparameter regions across the search space. The analysis conducted with various P_c values (0.1, 0.3, 0.5, 0.7, and 0.9) demonstrates that smaller values of P_c accelerate convergence and improve accuracy for simpler (unimodal) optimization landscapes, whereas larger values enhance exploration and stability for complex (multimodal) search spaces. Thus, appropriate tuning of P_c is crucial for achieving an optimal balance between convergence speed and search diversity.

The RCCO algorithm's adaptive nature is determined by its internal parameters, that consist of the probability coefficient P_c and the ratio of random foragers to long-distance foragers. The red-crowned crane population size n, the maximum number of function evaluations FE_{max}, and the maximum number of iterations (t_{max}) are the external parameters. The optimization process continues until the specified iteration or evaluation limit is reached, ensuring computational efficiency and convergence toward the most

effective hyper parameter configuration for EEBCADnet. The parameter setting for RCCO is provided in Table 1.

The set of optimal hyper parameters for the EEBCADnet model, optimized using the RCCO algorithm for AD prediction is presented in Table 2. This table includes both the search ranges explored during the optimization process and the corresponding optimal values that yielded the best model performance.

In this study, we tune a combination of model-specific and shared hyper parameters: VGGNet-specific parameters 1D-CNN parameters and several joint hyperparameters affecting both modules. These fine-tuned parameters maximize the EEBCADnet model's prediction capabilities and offer efficient training dynamics.

TABLE 1. PARAMETER SETTINGS FOR RCCO

Parameters	*Range / Value*
Probability coefficient (pc)	0.1 – 0.9 (commonly 0.7)
Ratio (Random : Long-distance) (k : n−k)	Typically 1 : 1
Population size (n)	50
Maximum iterations (tmax)	1000
Maximum function evaluations (FEmax)	50,000 (CEC-2005) / 100,000 (CEC-2022)
Step size coefficient c1	Constant = 2
Step size coefficient c2	Decreases linearly: 5 → 1 (5 − 4·t/tmax)
Step size coefficient c3	Decreases linearly: 2 → 1 (2 − t/tmax)
Deviation coefficient	Random in [0, 0.1]
Risk coefficient	Random in [0, 1]

TABLE 2. LIST OF OPTIMAL HYPERPARAMETERS FOR EEBCADNET MODEL

Parameter	Modules	Search Range	Optimal Value
Freeze layers	VGGNet only	[4, 6, 8,10]	8
No. of filters per convolutional layers	VGGNet only	[32, 64, 96, 128]	128
Kernel size	1D-CNN only	[3, 5, 7]	5
Pooling type	1D-CNN only	[Max, Average]	Max
Learning rate	Both	0.00001 – 0.1	0.0005
Epochs	Both	100 – 150	120
Batch size	Both	8 – 64	32
Optimizer	Both	[SGD, Adam, RMSProp, Adagrad]	Adam
Activation function	Both	[ReLU, GELU, LeakyReLU]	ReLU
Momentum	Both	0.5 – 0.99	0.9
Weight decay	Both	0 – 0.01	0.0007
Dropout rate	Both	0.3 – 0.7	0.4
Learning rate scheduler	Both	[Step, Cosine, Exponential]	Cosine
Early stopping patience	Both	5 – 20	10
Feature fusion weight	Both	0 – 1	0.6

Fig 2 and the following algorithm show the suggested model's flow.

Algorithm: RCCO based Hyperparameter Selection in EEBCADnet model

Input: Time maximum for iterations t_{max}, and maximum number of evaluations of functions FE_{max}, n (population size), P_c (probability coefficient), and the ratio of random foragers to long-distance foragers denoted as k: $(n-k)$.

Output: Maximal hyper parameter settings X_{best} and their associated fitness values $F(X_{best})$ (E.g. Validation accuracy)

1. Create initial sets of hyper parameters X_{cranes}, which represents the red-crowned crane population, using (1) and (2).
2. Make $FEs = 0$ and $t = 0$.
3. Assuming that $(t < t_{max} \backslash FEs < FE_{max})$
4. Evaluate the fitness of each hyperparameter set using Eq. (3) (train EEBCADnet with the given hyper parameters and compute validation accuracy).

5. Record the best (first) and second-best hyperparameter configurations so far.
6. Make a choice between foraging and roosting if $r_5 < p_c$.
7. Select X_{home} as the location that corresponds to the original fitness value, which represents the best set of hyper parameters at the moment.
8. Classify each hyper parameter set containing red-crowned cranes by their fitness scores.
9. Local hyper parameter adjustment when $i = F1{:}FK$ /Foraging of random foragers.
10. Update the hyper parameter sets using Eq. (6) (Recalculate fitness for these updated hyperparameter sets)
11. Finish with
12. Global hyper parameter search for the function $i = F(k+1){:}Fn$ /Foraging of long-distance foragers.
13. Apply Eq. (7) to the global hyper parameter sets, which contain the positions of long-distance foragers.
14. (Escape behaviour to avoid regions with low hyper parameters) if $c_r < \left({}^{t}/_{t_{max}}\right)^{1/2}$.
15. Generate a random hyperparameter set X_{rand} and record previous best for this agent (X_{ipbest} of distant foragers).
16. To move to a new potentially valuable location, update the long-distance foragers' positions (sets of improved hyper parameters) using Eq. (8).
17. end if
18. Evaluate once again how fit long-distance foragers really.
19. end for
20. Make sure to update X_{home} if any better hyperparameter configuration is identified during foraging by comparing the fitness values of all red-crowned cranes with the fitness values of X_{home}.
21. When $i = 1{:}n$, Roosting behaviour of red-crowned cranes
22. Using Eq. (9), shift all red-crowned crane positions, or hyper parameter sets, to the optimal configuration X_{home}.
23. end for
24. Change (FEs = FEs + 2n)
25. In any case
26. With respect to $i = 1$, the Crane dance strategy is n.
27. \\Pairwise recombination of top hyperparameter sets
28. Applying Eq. (12), which combines the top two configurations, i.e., the best learning rate with the second-best optimizer, update the positions of red-crowned crams for each pair of hyper parameters.
29. Finish
30. modify FEs = FEs + n.
31. Finish if
32. $t = t + 1$
33. Finish while
34. This function should return X_{best} and $F(X_{best})$,which stand for the best validation performance and the hyper parameter configuration.

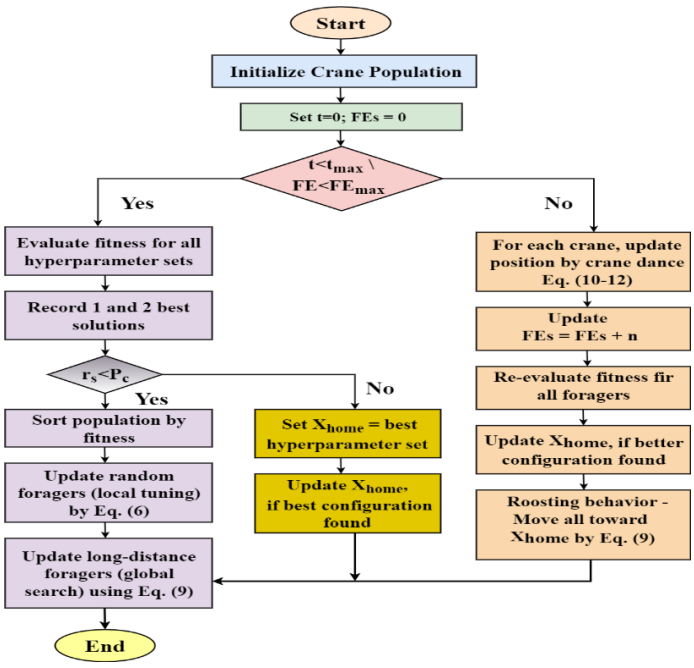

Fig. 2. RCCO based Hyper parameter Selection in EEBCADnet model

IV. RESULT AND DISCUSSION

To understand the efficacy of the proposed EEBCADnet model, this section compares its performance to that of existing traditionally trained models, such as ALZENET [26], EGELU-SZN [27], HHO-MLP-LSTM [28], EffSwin-XNet [29], and EEBCADnet [18].

A. Dataset Description

For the performance evaluation, two datasets are used, i.e., Kaggle AD dataset with 4 classes and ADNI Dataset with 3 classes.

Kaggle AD Dataset [32]: This dataset consisted of 6400 MRI images with a resolution of 256 × 256 pixels that have been classified into four classes like Moderate Demented (MOD) - 64, Mild Demented (MID) - 896, Very MID (VMD) - 2240 images and Non-Demented (ND) - 3200 images. The images are depicted in fig 3.

ADNI dataset [33]: This dataset consists of totally 5154 images with 3 classes like 1440 - Common Normal (CN) subjects, 2590

- Mild Cognitive Impairment (MCI) and 1124 - AD subjects, the images are presented in fig4.

B. Experimental Setup and Parameter Settings

The experiment was performed on a computer configured with an Intel® Core TM i7-8700 CPU, 16GB memory, NVIDIA GTX-1060 GPU, and Windows 10 OS. The programming platform is Python 3.12 and the DL framework is Tensorflow 1.6. Both the datasets are splitted into 70% for training and 30% for testing.

C. Performance Evaluation Metrics

Standard evaluation tools for AD diagnosis include recall, F1-score, precision, and accuracy. The dataset under consideration consists of different classes for evaluation.

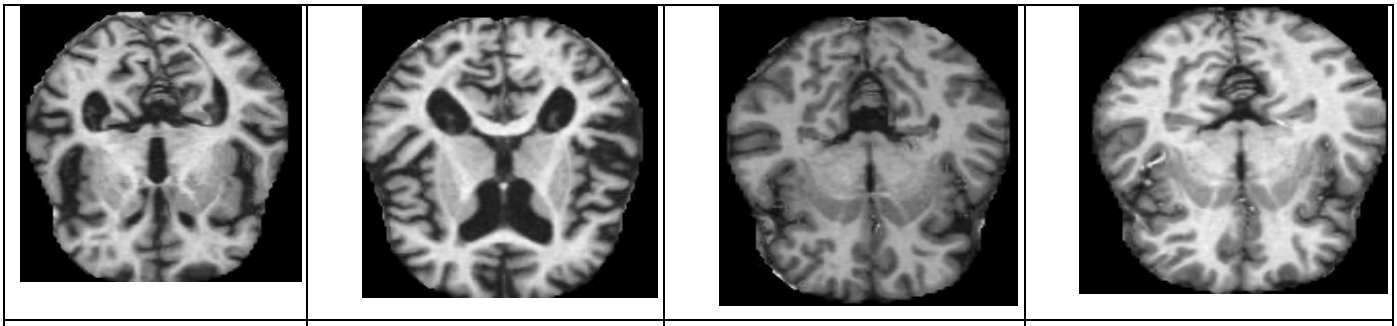

Fig. 3. Sample Images of Kaggle AD Dataset with their Respective Classes

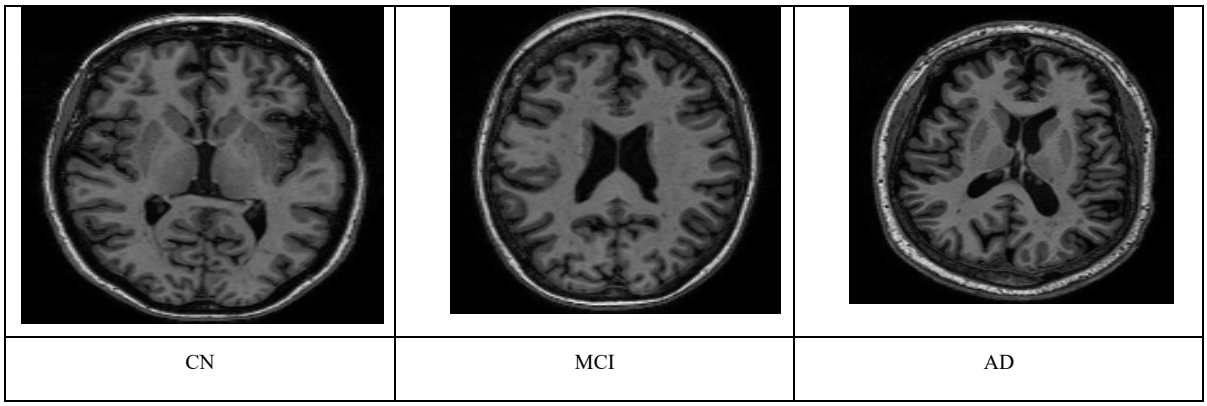

Fig. 4. Sample Images of ADNI Dataset with their Respective Classes

During the categorization process four possibilities may occur: True Positives (TP): When the model accurately detects an individual with different classes of AD. For E.g. MID is exactly predicted as MID\CN as CN.

False Positive (FP): If the model gets NC or MOD wrong, it results in an inaccurate prediction.

False Negative (FN): When there is no detection by the model AD, MCI or dementia despite the subject actually having the condition.

True Negatives (TN): When the model correctly classifies the subject as ND or the subject as NC.

Based on these classification results, the evaluation metrics are defined and calculated.

Accuracy: It is the percentage of correctly classified different AD stages like AD, MCI, ND, NC, MID, MOD, VMD by the test model, reflecting the classification ability of the model.

$$Accuracy = \frac{True\ Positive\ (TP)+True\ Negative\ (TN)}{TP+TN+False\ Positive\ (FP)+False\ Negative\ (FN)} \quad (14)$$

Precision: It is computed as:

$$Precision = \frac{TP}{TP+FP} \quad (15)$$

Recall: It is measured by

$$Recall = \frac{TP}{TP+FN} \quad (16)$$

F1-score: It is determined by

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (17)$$

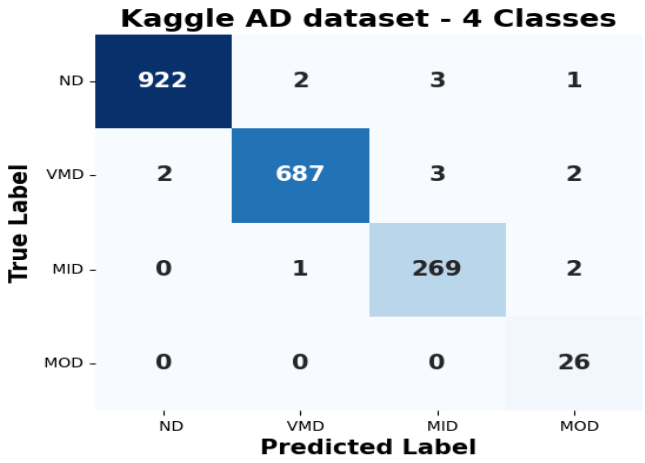

Fig. 5. Confusion Matrix for the Proposed Model for Kaggle Dataset (Testing set)

D. Performance Analysis of Proposed model with Existing models on Kaggle Dataset with 4 Classes for AD Classification

In this section, the proposed OEEBCADnet model is compared with other models like ALZENET, EGELU-SZN, HHO-MLP-LSTM, EffSwin-XNet and EEBCADnet on ADNI dataset with 4 classes like MID, MOD, ND and VMD images for AD classification which is briefly provided in section 4.1 by using the performance indicators as in section 4.3. Confusion Matrix of testing set for the developed model is shown in fig 5.

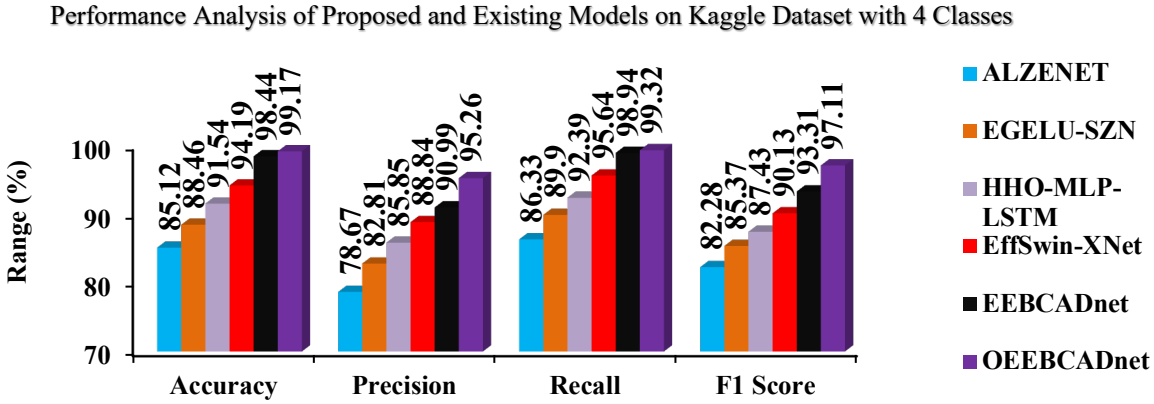

Fig. 6. Scrutiny of Various AD stages Detection Models on Kaggle AD Dataset with 4 Classe

In Fig. 6, the analysis of proposed OEEBCADnet model and standard models are evaluated using different metrics on Kaggle AD Dataset for 4 classes in AD classification. In this analysis, the accuracy of OEEBCADnet is 16.51 %, 12.11 %, 8.34 %, 5.29 % and 0.74 % higher than other models like ALZENET, EGELU-SZN, HHO-MLP-LSTM, EffSwin-XNet and EEBCADnet models correspondingly. The precision of OEEBCADnet is 21.09 %, 15.03 %, 10.96 %, 7.23 % and 4.69 % higher than other models respectively.

The recall of O EEBCADnet is 15.05 %, 10.48 %, 7.50 %, 3.85 % and 0.38 % higher than other models and the F1-score of O EEBCADnet is 18.03 %, 13.75 %, 11.07 %, 7.74 % and 4.07 % higher than other models respectively. This is because the suggested model utilizes the novel metaheuristic alogrithm (RCCO) to optimize the parameters which enhances model's performances in AD classification.

E. Performance Analysis of Proposed model with Existing models on ADNI Dataset with 3 Classes for AD prediction

In this section, the proposed OEEBCADnet model is compared with other models like ALZENET, EGELU-SZN, HHO-MLP-LSTM, EffSwin-XNet and EEBCADnet on ADNI dataset with 3 classes like CN, MIC and AD images for AD prediction which is briefly provided in section 4.1 by using the performance indicators as in section 4.3. Confusion Matrix of testing set for the proposed model is shown in figure 7.

In Fig. 8, the analysis of proposed OEEBCADnet model and standard models are evaluated using different metrics on ADNI Dataset for 3 classes in AD classification. In this analysis, the accuracy of OEEBCADnet is 20.27 %, 13.24 %, 8.98 % 6.34 % and 0.66 % higher than other models like ALZENET, EGELU-SZN, HHO-MLP-LSTM, EffSwin-XNet and EEBCADnet models correspondingly.

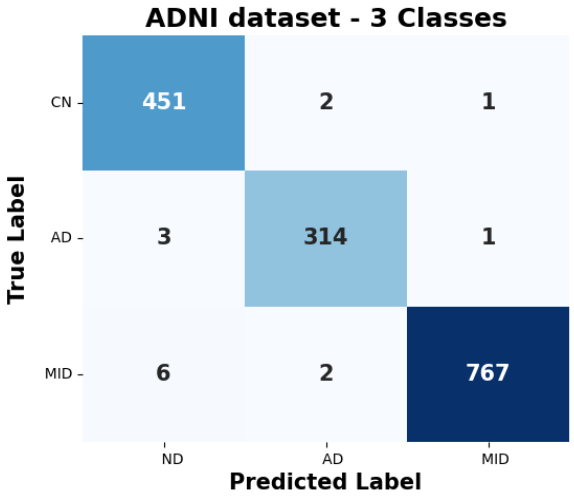

Fig. 7. Confusion Matrix for the Proposed Model for ADNI Dataset (Testing set)

The precision of OEEBCADnet is 21.69 %, 15.38 %, 8.47 %, 5.32 % and 0.83% higher than other models respectively. The recall of OEEBCADnet is 16.93 %, 13.83 %, 7.61 %, 5.08 % and 0.62 % higher than other models and the F1-score of OEEBCADnet is 19 %, 14.72 %, 9.10 %, 5.15 % and 0.76 % higher than other models respectively. This is because the proposed model lowers the computational complexity and boost the models performance by using the RCCO model for better AD prediction.

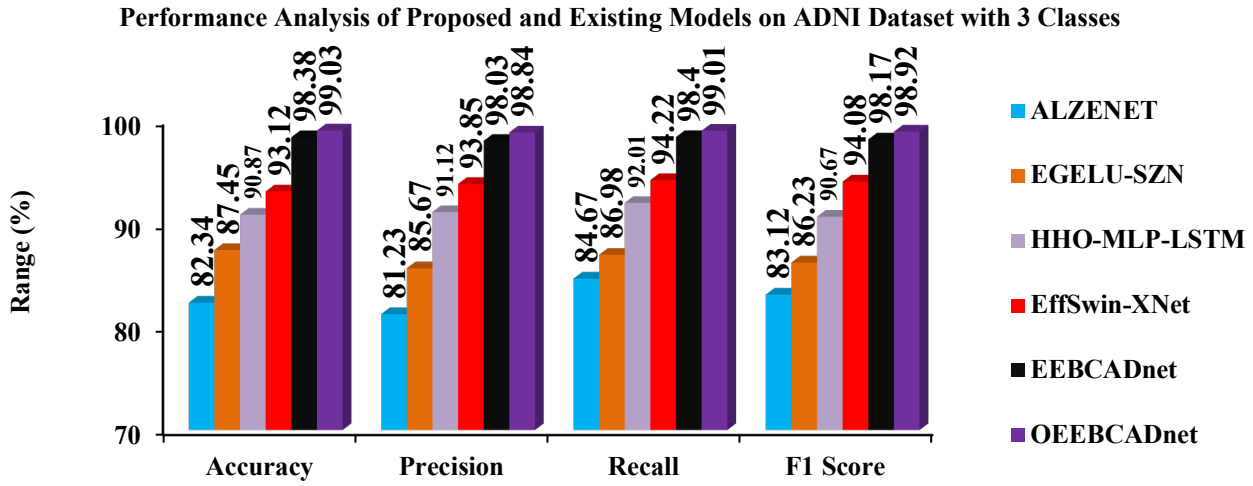

Fig. 8. Scrutiny of Various AD stages Detection Models on ADNI Dataset with 3 Classes

Table 3 depicts the computational efficiency comparison of EEBCADNET and OEEBCADnet on Kaggle and ADNI Dataset. The suggested technique consistently reduces training time (s), inference time (s), and memory usage (MB) on both CPU and GPU architectures. Both models significantly reduce training times on CPU and GPU, with a reduction of approximately 5-10% compared to the baseline values. In terms of inference time, OEEBCADnet delivers faster results with a decrease of around 15-20% on both hardware platforms, suggesting its suitability for real-time applications. While the OEEBCADnet model utilizes slightly more CPU memory than EEBCADnet, its GPU memory usage is optimized, showing a reduction of 5-10% compared to other models. These findings highlight that OEEBCADnet is not only computationally efficient but also well-suited for deployment in resource-constrained environments for effective AD prediction.

Table 4 compares the performance of proposed OEEBCADnet model for without hyperparameter optimization, Random Search, Grid Search and RCCO (metaheuristic model applied in this study on the Kaggle and ADNI dataset. Without hyperparameter optimization (HO), the model's performance drops by 3%-4% across all metrics, emphasizing the importance of fine-tuning parameters for optimal convergence. While both Grid Search and Random Search show improvements over the unoptimized model, they still fall short compared to RCCO. Grid Search provides modest gains but is less effective than RCCO and Random Search offers some improvement, though it remains less efficient. Overall, RCCO significantly outperforms both methods, delivering superior precision, accuracy, F1-score and recall.

TABLE 3 COMPUTATIONAL EFFICIENCY COMPARISON OF OEEBCADNET WITH OTHER MODEL

Hardware	Kaggle Dataset		ADNI Dataset	
	EEBCADnet	OEEBCADnet	EEBCADnet	OEEBCADnet
Training time (s)				
CPU	1250	960	940	880
GPU	1870	1260	1610	1030
Inference Time (s)				
CPU	8.6	5.9	7.2	3.9
GPU	5.4	2.1	4.6	2.8
Memory Usage (MB)				
CPU	2260	3870	2140	3120
GPU	1860	1040	1460	2690

TABLE 4 IMPACT OF HYPERPARAMETER OPTIMIZATION OF PROPOSED OEEBCADNET MODEL

Dataset	Metric	Without Hyperparameter optimization	Random Search	Grid Search	RCCO
Kaggle Dataset	Accuracy (%)	85.12	88.76	92.13	99.17
	Precision (%)	84.37	89.14	91.86	95.26
	Recall (%)	85.68	88.97	92.45	99.32
	F1-Score (%)	84.95	89.33	91.92	97.11
ADNI Dataset	Accuracy (%)	84.89	89.27	92.34	99.03
	Precision (%)	85.41	88.66	91.57	98.84
	Recall (%)	84.76	89.02	92.11	99.01
	F1-Score (%)	85.03	88.95	91.84	98.92

CONCLUSION

This article explains how to tune the EEBCADnet model's parameters for accurate AD prediction using RCCO algorithm. The CCO algorithm is inspired by the adaptive behaviors of red-crowned cranes, efficiently explores the hyperparameter search space, balancing exploration and exploitation to converge on the optimal configurations. By applying RCCO, the hyper parameters of VGGNet and 1DCNN are optimized both globally and locally for better model performance. The proposed RCCO-based hyperparameter tuning significantly enhances the diagnostic accuracy and lowers the computational complexity of EEBCADnet model for early AD detection. The test results reveals that the proposed model achieves 99.17% and 99.03% of accuracy on Kaggle AD dataset and ADNI datasets surpassing other existing models.

REFERENCES

[1]. W. Smyth, E. Fielding, E. Beattie, A. Gardner, W. Moyle, S. Franklin, ... and M. MacAndrew, "A survey-based study of knowledge of Alzheimer's disease among health care staff," BMC Geriatr., vol. 13, pp. 2, 2013,*doi: 10.1186/1471-2318-13-2.*

[2]. O. Zanetti, S. B. Solerte, and F. Cantoni, "Life expectancy in Alzheimer's disease (AD)," Arch. Gerontol. Geriatr., vol. 49, pp. 237–243, 2009, *doi:10.1016/j.archger.2009.09.035.*

[3]. J. Rasmussen and H. Langerman, "Alzheimer's disease–why we need early diagnosis," Degener. Neurol. Neuromuscul. Dis., pp. 123–130, 2019, *doi:10.2147/DNND.S228939.*

[4]. M. W. Bondi, E. C. Edmonds, and D. P. Salmon, "Alzheimer's disease: past, present, and future," J. Int. Neuropsychol. Soc., vol. 23, pp. 818–831, 2017, *doi:10.1017/S135561771700100X.*

[5]. Breijyeh and R. Karaman, "Comprehensive review on Alzheimer's disease: causes and treatment," Molecules, vol. 25, pp. 5789, 2020,*doi:10.3390/molecules25245789*.

[6]. S. Sindi, F. Mangialasche, and M. Kivipelto, "Advances in the prevention of Alzheimer's disease," F1000Prime Rep., vol. 7, pp. 50, 2015,*doi: 10.12703/P7-50.*

[7]. Y. Chen, X. Qian, Y. Zhang, W. Su, Y. Huang, X. Wang, ... and Y. Ma, "Prediction models for conversion from mild cognitive impairment to Alzheimer's disease: a systematic review and meta-analysis," Front. Aging Neurosci., vol. 14, pp. 840386, 2022, *doi:10.3389/fnagi.2022.840386*.

[8]. E. M. Reiman and W. J. Jagust, "Brain imaging in the study of Alzheimer's disease," Neuroimage, vol. 61, pp. 505–516, 2012,*doi: 10.1016/j.neuroimage.2011.11.075.*

[9]. W. M. van Oostveen and E. C. de Lange, "Imaging techniques in Alzheimer's disease: a review of applications in early diagnosis and longitudinal monitoring," Int. J. Mol. Sci., vol. 22, pp. 2110, 2021,*doi: https://doi.org/10.3390/ijms22042110.*

[10]. A. Chandra, G. Dervenoulas, M. Politis, and Alzheimer's Disease Neuroimaging Initiative, "Magnetic resonance imaging in Alzheimer's disease and mild cognitive impairment," J. Neurol., vol. 266, pp. 1293–1302, 2019,*doi:10.1155/2018/5360375.*

[11]. P. Vemuri and C. R. Jack Jr., "Role of structural MRI in Alzheimer's disease," Alzheimers Res. Ther., vol. 2, pp. 23, 2010,*doi: 10.1186/alzrt47*.

[12]. R. Wolz, V. Julkunen, J. Koikkalainen, E. Niskanen, D. P. Zhang, D. Rueckert, ... and Alzheimer's Disease Neuroimaging Initiative, "Multi-method analysis of MRI images in early diagnostics of Alzheimer's disease," PLoS One, vol. 6, pp. e25446, 2011,*doi: 10.1371/journal.pone.0025446.*

[13]. E. G. Marwa, H. E. D. Moustafa, F. Khalifa, H. Khater, and E. AbdElhalim, "An MRI-based deep learning approach for accurate detection of Alzheimer's disease," Alexandria Eng. J., vol. 63, pp. 211–221, 2023,*doi:10.1016/j.aej.2022.07.062.*

[14]. E. Altinkaya, K. Polat, and B. Barakli, "Detection of Alzheimer's disease and dementia states based on deep learning from MRI images: a comprehensive review," J. Inst. Electron. Comput., vol. 1, pp. 39–53, 2020, *doi: 10.33969/JIEC.2019.11005.*

[15]. T. J. Saleem, S. R. Zahra, F. Wu, A. Alwakeel, M. Alwakeel, F. Jeribi, and M. Hijji, "Deep learning-based diagnosis of Alzheimer's disease," J. Pers. Med., vol. 12, pp. 815, 2022,*doi: 10.3390/jpm12050815.*

[16]. R. A. Hazarika, A. K. Maji, D. Kandar, E. Jasinska, P. Krejci, Z. Leonowicz, and M. Jasinski, "An approach for classification of Alzheimer's disease using deep neural network and brain magnetic resonance imaging (MRI)," Electronics, vol. 12, pp. 676, 2023,*doi: 10.3390/electronics12030676.*

[17]. V. P. Nithya, N. Mohanasundaram, and R. Santhosh, "An early detection and classification of Alzheimer's disease framework based on ResNet-50," Curr. Med. Imaging, vol. 20, pp. e250823220361, 2024, *doi: 10.2174/1573405620666230825113344.*

[18]. C. M. Kim and W. Lee, "Classification of Alzheimer's disease using Ensemble Convolutional Neural Network with LFA Algorithm," IEEE Access, 2023,*doi: 10.1109/ACCESS.2023.3342917.*

[19]. S. Karpagam and B. R. Jeetha, "An Enhanced Ensemble Alzheimer Disease Detection Network with a Refined Line Segment Feature Analysis," 2026.

[20]. D. Chakraborty, Z. Zhuang, H. Xue, M. B. Fiecas, X. Shen, and W. Pan, "Deep learning-based feature extraction with MRI data in neuroimaging genetics for Alzheimer's disease," Genes, vol. 14, pp. 626, 2023,*doi: 10.3390/genes14030626.*

[21]. R. SinhaRoy and A. Sen, "A hybrid deep learning framework to predict Alzheimer's disease progression using generative adversarial networks and deep convolutional neural networks," Arab. J. Sci. Eng., vol. 49, pp. 3267–3284, 2024.

[22]. C. Mahanty, T. Rajesh, N. Govil, N. Venkateswarulu, S. Kumar, A. Lasisi, ... and W. A. Khan, "Effective Alzheimer's disease detection using enhanced Xception blending with snapshot ensemble," Sci. Rep., vol. 14, pp. 29263, 2024,doi: *10.1038/s41598-024-80548-2.*

[23]. N. Goenka, A. K. Sharma, S. Tiwari, N. Singh, V. Yadav, S. Prabhu, and K. Chadaga, "A regularized volumetric ConvNet based Alzheimer detection using T1-weighted MRI images," Cogent Eng., vol. 11, pp. 2314872, 2024,*doi: 10.1080/23311916.2024.2314872.*

[24]. M. Leela and K. Helenprabha, "Incremental learning based two-level multimodal data fusion model for Alzheimer disease prediction on different data modalities," Connection Sci., vol. 37, pp. 2458501, 2025,*doi: 10.1080/09540091.2025.2458501.*

[25]. S. Abbas, M. Yeniad, and J. Rahebi, "Alzheimer's Disease prediction using Fisher mantis optimization and hybrid deep learning models," Diagnostics, vol. 15, pp. 1449, 2025,*doi:10.3390/diagnostics15121449.*

[26]. M. Asaduzzaman, M. K. Alom, and M. E. Karim, "ALZENET: Deep learning-based early prediction of Alzheimer's disease through magnetic resonance imaging analysis," Telemat. Inform. Rep., vol. 17, pp. 100189, 2025, *doi: 10.1016/j.teler.2025.100189.*

[27]. B. Sathyabhama and M. Kannan, "An effective deep learning-based automatic prediction and classification of Alzheimer's disease using EGELU-SZN technique," Neural Comput. Appl., vol. 37, pp. 6915–6932, 2025.

[28]. R. Ghadami and J. Rahebi, "Alzheimer's prediction methods with Harris Hawks Optimization (HHO) and deep learning-based approach using an MLP-LSTM hybrid network," Diagnostics, vol. 15, pp. 377, 2025,*doi: 10.3390/diagnostics15030377.*

[29]. K. Velu and N. Jaisankar, "Design of a CNN–Swin transformer model for Alzheimer's disease prediction using MRI images," IEEE Access, vol. 99, pp. 1–1, 2025,*doi: 10.1109/ACCESS.2025.3602316.*

[30]. Tourist55, "Alzheimer's Dataset (4 Class of Images)," Kaggle, 2025. [Online]. Available: https://www.kaggle.com/tourist55/alzheimers-dataset-4-class-of-images. [Accessed: Mar. 9, 2026].

[31]. Katalniraj, "ADNI Extracted Axial Images," Kaggle, 2025. [Online].Available: https://www.kaggle.com/datasets/katalniraj/adni-extracted-axial. [Accessed: Mar. 9, 2026].