International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Deep learning Based Alzheimer's Disease Detection Model using Multimodal Data

Anjani Yalamanchili[1*], Dr. D. Venkatasekhar[2], Dr. G. Vijay Kumar[3]

Submitted: 25/01/2024 **Revised**: 03/03/2024 **Accepted**: 11/03/2024

Abstract: Alzheimer's disease (AD) is considered an irreversible and progressive neurodegenerative disease that causes mortality in older people. Thus, early AD detection offers a significant role in controlling and preventing its progression. This paper proposes an DL-based AD detection model (DL-ADDM) based on deep learning (DL) together with multimodal feature processing that includes IoT data and image data for early-stage AD detection. The inputs of different data formats are fed into a multimodality-based integrated classifier model (MICM). For the generation of IoT data modal, pre-processing is carried through missing value imputation, data normalization and validation. Then, the features of the IoT data are extracted through a weighted auto encoder (WAE) along with an Attention-based Bi-directional long short-term memory (Attn_BiLSTM). In the image model, the pre-processing of images is performed using a Min-Max Gaussian filtering (M-squared GF) approach. Then, the informative features are extracted using the convolutional capsule network_chimp optimization algorithm (Conv_Capsnet_COA). Afterwards, the feature vectors obtained from the multimodal data processing are fused using the deep fusion strategy (DFS). Finally, the multimodal data output is classified through the softmax unit. The proposed DL-ADDM is simulated on the Python platform and the performances are evaluated using the main cognitive testing dataset (IoT data) and the Kaggle dataset (image data). The simulated outcomes showed that the proposed model had reached a maximum accuracy of 99.5% for CN/AD, 99.3% for CN/MCI and 99.6% for MCI/AD class.

Keywords: *Deep learning, pre-processing, multimodal feature processing, optimization algorithm, weighted autoencoder, deep fusion strategy, and multi-class classification.*

1. Introduction

Due to the advancement of technology, the IoT (Internet of things) have been utilized in various fields, and an IoT-based monitoring and detection model assists physicians. IoT connects millions of smart devices and is used to communicate with the less human intervention [1]. Alzheimer's disease (AD) is an incurable, chronic and irreversible neurodegenerative syndrome. Older people are suffered due to dementia because of AD. This disease slightly emerges from mild, moderate and severe stages. According to the World Health Organization (WHO) report, AD affects 51 million people and is expected to triple by 2050 [2-4]. AD is caused due to abnormal growth of proteins such as tangles and amyloid in the brain. The next stage of AD is MCI (mild cognitive impairment), which causes changes in cognitive function. The final phase is dementia which causes memory loss and an inability to carry out normal activities. Still, no treatment is available to cure the disease. Hence, there is a need for a model which identify the disease in its initial stages [5-8]. People who are advanced from MCI to AD are considered pMCI, and those who aren't advanced from MCI to AD are considered sMCI. Various research works are analyzed to understand the pathological factors in the brain. Imaging modalities such as cerebrospinal fluids (CSF), positron emission tomography (PET), functional and structural magnetic resonance imaging (fMRI and sMRI) are utilized as biomarkers for classifying the stages of diseases [9]. These biomarkers show the functional and structural details of the brain. Further, the sMRI is classified into cognitively normal (CN), MCI and AD. MMSE (mini mental state examination) scores are recently deliberated for AD diagnosis. It is a thirty-point questionnaire and is utilized for screening dementia. Then, the LM (logical memory) test is utilized to assess verbal memory [10-13].

Magnetic resonance imaging (MRI) modalities ensure a noninvasive and efficient model for evaluating and understanding brain function. They played a major role in medical practice and were declared essential biomarkers for AD progression. During the last two decades, efforts are taken by experts to early diagnosis of AD using machine learning (ML) and deep learning models (DL) [14]. Various computer-aided diagnosis (CAD) systems are introduced using ML models to decode disease states from MRI images. The conventional ML-based model extracted the features manually.

[1] Research Scholar, Department of Information Technology,
Annamalai University, Annamalai Nagar, Tamil Nadu, 608002, India.
[*] Email: anjaniyalamanchili@gmail.com
[2] Professor, Department of Information Technology,
Annamalai University, Annamalai Nagar, Tamil Nadu, 608002, India.
Email: ramavenkatasekar@yahoo.co.in
[3] Professor, Department of Computer Science & Engineering,
SSCET, Challapalli, Andhra Pradesh, 521131, India.
Email: vijayg.teja@gmail.com

Quantitative analyses are developed to analyze the brain's texture, shape, surface, volume and thickness [15]. In addition, the DL models have shown remarkable achievements based on prediction in various clinical fields. The DL model CNN (convolutional neural network) has been majorly employed for extracting and classifying the AD stages. Due to the network's complexity and increasing depth, the performance of the classification is improved. The DL model's recurrent neural network (RNN) is used to extract the deep longitudinal features. The variants of RNN are LSTM (long short term memory) as well as Bi-LSTM (Bidirectional LSTM) are used for predicting the stages of AD [16-17]. Inspired by the emergence of DL models and attention modules in the clinical fields, this works introduces a DL model for extracting and classifying multimodal data.

Motivation: The process of classifying multimodal data tends to be challenging in AD detection. Fewer efforts were promoted to generate an efficient approach for multimodal data classification. Most of the existing approaches are appropriate for a single modality, and huge data support cannot be attained. Recently, DL-based approaches have gained greater attention towards multimodal data classification. The existing approaches have failed to generate improved accuracy in AD detection due to the huge accumulation of data, degraded transfer learning ability and found hard to extract valuable features. Also, the execution time is more to train huge data, so the detection speed leads to a minimum. There are high chances of classification error which deteriorates the system's overall performance. To conquer these issues, the proposed work adopts a DL-based multimodal data classification with novel approaches.

Contribution: The foremost objective of this paper is to provide a new DL-based multimodal data classification model for detecting AD at an early stage. The contributions of the proposed DL-ADDM are enumerated as follows:

- To develop artificial intelligence and the Internet of things that enable multi-model feature processing for early AD detection.
- To exact text and image features using attention-based bi-directional long short term memory (Attn_BiLSTM) and convolutional capsule network_chimp optimization algorithm (Conv_Capsnet_COA).
- To fuse the multimodal features using deep fusion strategy (DFS) and predicting the performance.
- To compare the performance of the proposed DL-ADDM with other techniques based on accuracy, precision, f-score, recall and specificity in terms of MCI/AD, CN/MCI and CN/AD.

The paper is summarized as follows: Section 2 discusses recent work by various authors on the proposed strategy. Section 3 designates the methodology of the proposed DL-ADDM. Section 4 contains the experimental results of the proposed DL-ADDM. Finally, the conclusion and future work are presented in Section 5.

2. Related Works

Some of the recent works based on AD classification using text and image data are listed in this section.

Qiu, Shangran et al. [18] presented a fusion of DL models of MRI, LM and MMSE for diagnosing MCI. The MCI and CN data of the individuals have been collected from the NACC (National Alzheimer Coordinating Center) dataset. The DL models trained on MRI images were integrated to generate the fused MRI model for predicting CN and MCI. Two MLP (multi-layer perceptron) were created with LM and MMSE outcomes. Finally, the MLP and fused MRI were integrated by majority voting. The fused approach was superior to an individual approach and achieved a better accuracy of 90.9%.

On considering the classification accuracy, El-Sappagh et al. [19] presented a two-step DL approach to detect and predict AD from MCI. The initial phase of the model consisted of multi-class classification processes that predicted the stages like MCI, AD or CN. In the next phase, the regression model was used to identify the correct conversion period of MCI. This model was based on data from 1371 samples obtained with ADNI. Extensive experiments were performed with several ML and DL models. The classification performance achieved by LSTM was 93.8%, and this model outperformed other ML models.

On account of considering better data utilization, Divya and Shantha Selva Kumari [20] presented GA (genetic algorithm) with various ML models for AD classification. This work classified the chronic disorder as CN, AD and MCI based on MRI images. Initially, the ADNI dataset was pre-processed, and recursive feature elimination (RFE) with GA was used for selecting the features. Finally, GA-based LR (logistic regression) achieved a better MMSE value of a 2.6% increment in accuracy and an 8.7% increment in sensitivity. This model was created with fewer features compared to other works.

El-Sappagh et al. [21] presented an AD progression model based on multimodal data to minimize the computational complexity. The phases include collecting data, pre-processing, splitting data, balancing data, optimizing hyperparameters, training and predicting AD. This model fused the feature of comorbidity and medication. Here, the dataset is obtained in the format of 5-time series modalities. Five ML models were merged and performed classification based on four class and three class tasks. Among all models, the random classifier forest (RF) achieved better performance and was stable.

On considering the training time complexity, Helaly et al. [22] presented a DL model for early AD detection, and the four stages were multi-classified. In this work, two approaches were utilized for classifying and detecting AD. Initially, the CNN was used for dealing with 2D and 3D images and the second model, VGG19, was used to fine-tune for detecting the AD. Initially, the CNN achieved better accuracy of 93.6%, and fine-tuning of VGG achieved a better accuracy of 97%. This model was simple, and it reduced the overfitting issues and complexity.

To sort out the convergence issues, Zhang et al. [23] demonstrated a 3D dense-based CNN with a connection-wise attention module for AD classification. A densely connected network was used to extract multiple features, and a connection-wise attention module was utilized to combine the connections between the features. Then, the 3D convolution operation is used to acquire the spatial information of the MRIs. This spatial information was combined with features from all layers and utilized for classification. This model achieved better accuracy of 97.3% to distinguish AD people from the healthy control.

To effectively validate the features and enhance the training capability, Sun et al. [24] presented an enhanced ResNet model intending to diagnose AD at the early stage. Initially, the dataset was collected from ADNI, and ResNet-50 was used to extract the information from the layers. A non-local attention module was utilized in ResNet-50. The spatial transformer network was used for extracting spatial information in the MRI images. This model achieved better F1 scores and recall values of 95.4% and 95.3%, which was more effective than other models. Table 1 presents the advantages and limitations of the existing techniques.

Table 1: Advantages and limitations of existing techniques

Author (s)	Technique (s)	Advantages	Limitations
Qiu, Shangran et al. [18]	Majority voting	The fused approach was superior to an individual approach and achieved better accuracy of 90.9%	This study doesn’t describe how to choose the sub-set of biomarkers for the particular disease
El-Sappagh et al. [19]	Several ML and DL models	This model was clinically acceptable and stable	The decision of the model was not interpretable
Divya and Shantha Selva Kumari [20]	Several ML models	This model was created with fewer features when compared to other works	Achieved imbalanced outcomes because of improper large dataset utilization.
El-Sappagh et al. [21]	Several ML models	Integration of comorbidity and medication features provided better results and a stable model	High computational complexity is resulted because of improper feature learning capability.
Helaly et al. [22]	CNN-VGG19	This model was simple, reducing the complexity and overfitting issues	Training time was high because of high complexity in training the suitable features.
Zhang et al. [23]	3D dense-based CNN	Top-ranked classification and enhanced the discrimination of MCI	It has convergence issues and is computationally expensive in case of training parameters
Sun et al. [24]	Enhanced ResNet model	Efficiently extracted features and provided better recognition	Less training ability because of highly complex data.

Problem statement: In the past decades, various models have been presented individually for the early detection of AD, including classical ML and DL networks. However, most approaches lack multi-class medical image classification and employ AD inspection to check the AD phase and advise patients remotely. Some existing approaches suffer problems such as errors during training, less processing ability for a large amount of data, less accuracy, etc. Therefore, a novel approach based on IoT and DL is compulsory for the early detection of AD and classification.

3. Proposed Methodology

This section proposes an efficient DL-based multimodal model for predicting AD with Deep Fusion Strategy (DFS) and multimodal feature processing. Initially, the medical-related IoT data collected from the main cognitive tests dataset and the image data captured from the Alzheimer MRI pre-processed dataset (Kaggle dataset) are provided as input that is comprised of sensed information, followed by pre-processing and feature extraction. For the generation of IoT data modal, pre-processing is carried out through missing value imputation, data normalization and validation. The next process is feature fusion then the fused features are fed as input to the softmax classifier, which distinguishes the disease detection into different classes. During the data acquisition stage, the input data of a larger size are stored on the Hadoop platform for storage purposes. The block diagram of the proposed strategy is presented in Figure 1.

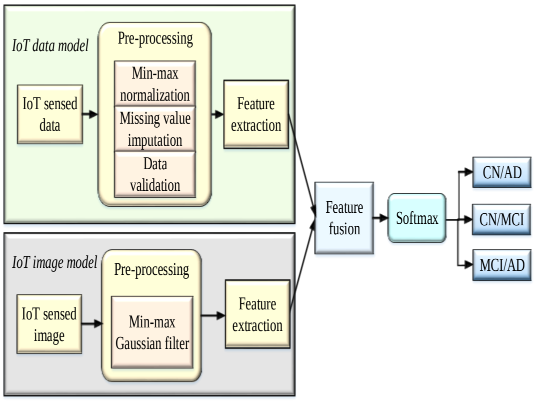


Fig 1: Architecture of proposed strategy

In figure 1, the proposed strategy consists of four layers, including the sensing layer, the analysis layer, the network layer and the application layer. The sensing layer is the lower layer, mainly used to acquire patients' data. The original data captured from the sensing layer is served as an input to the upper layer for the investigation. After processing the data, the original data is processed in the analysis layer to realize the situation of the patients. The doctors can examine the acquired results. This supports the doctors in distinguishing the patient's situation more concretely for endorsing a better treatment plan. In addition, numerous reception and transmission networks, including private networks, heterogeneous networks, mobile communication networks, the Internet, etc., are involved in considering application requests in the network layer. In addition, the data was transferred from the transmission layer to the application layer. Consequently, the application layer focuses on providing services to remote physicians. The results are acquired and inspected through portable, fixed, and mobile devices. Lastly, the doctors analyze the detection outcomes, and a treatment plan is recommended.

3.1 Dataset description

Two datasets like main cognitive testing dataset (IoT data) and the Kaggle dataset (image data) are employed in the proposed research work for multimodal AD classification. The description of two datasets are provided as follows.

IoT data: In the proposed AD detection model, the main cognitive tests are neuropsychological tests administered by a clinical expert, including the clinical dementia rating scale Sum of Boxes (CDR-SB) score, Montreal cognitive assessment (MoCA) and mini-mental state examination (MMSE) are considered for IoT data [25]. The CDR is computed based on scale values 0-18. The people with no dementia have a CDR-SB scale of 0, a scale value of 4.5 for mild dementia, 16-18 for severe dementia, 9.5-15.5 for moderate dementia, 2.5-4 for very mild dementia and 0.5-4 for questionable cognitive impairment. The highest score for MMSE is considered 30. A score value below 24 is normally resembled as abnormal, indicating possible cognitive impairment, and if the score is 25 or above, 25 is considered normal. MoCA is discovered as a brief screening tool for identifying cognitive impairment. Similar to MMSE, the MoCA scores can range between 0 and 30. Here, the people with AD scored an average of 16.2, a score of 26 or above is termed normal, and an average score of 22.1 is considered MCI. The IoT data present in the dataset are in the form of different intervals.

Image data: To obtain the image data, the proposed DL-ADDM has collected Alzheimer's MRI pre-processed dataset [26]. This dataset comprises pre-processed MRI images, and the data is acquired from different hospitals/websites/public repositories. Each image in the dataset has been resized to 128×128 pixels. The Alzheimer MRI pre-processed dataset encompasses 6400 MRI images with four different classes.

Here, 80% of the data are used for training and 20% of the data are used for testing. The major advantages of using IoT data and image data together is to obtain the interacted features from multimodal data whereas more significant information can be gained. The evaluation of complex details can be performed by fusing the features. Also, the capability of the DL model can be expanded which in turn promotes the classification accuracy.

3.2 Generation of IoT data modal

3.2.1 Pre-processing

The pre-processing states that the process of enlightening the substantial information and directing to upsurge overall classification performance. In the proposed DL-ADDM, the pre-processing stage comprises missing value imputation, data normalization and validation.

- ***Missing value imputation:*** Usually, the dataset includes a few missing values, which are replaced by considering the mean of non-missing values [27].
- ***Min-max normalization:*** Typically, the normalization process scales the data to fit within a specific range. Several normalization techniques have been accessible recently; however, min-max normalization [28] is considered in the proposed strategy because it has achieved regularity for a set of data in an appropriate dynamic range.
- ***Data validation:*** As a part of data pre-processing, data validation is performed to validate the collected data for enhancing the quality. Due to the huge data generation, the requirement to improve the data quality is high. Low-quality data generates false predictions or

inaccurate results and may affect the overall detection accuracy. The data validation process is applied to check the data validity in the dataset, including negative time values, empty values or negative amounts.

3.2.2 Feature extraction based on weighted autoencoder

The proposed strategy has employed a weighted autoencoder (WAE) model to extract the significant features for AD classification. WAE is considered an unsupervised neural network comprising a sequentially linked 3-layer structure comprising an input layer, a hidden layer and a reconstruction or output layer. Here, WAE gradually converts certain feature vectors into abstract vectors, which was realized by the non-linear transformation from higher to lower dimensional data space. The operation of WAE involves two stages such as encoding and decoding. In the coding phase, input data can be mapped into the hidden layer by learning the input's latent variables or compressed representations. While in the decoding phase, the data can be reconstructed from the hidden layer representation.

The encoding process from the input layer to the hidden layer was specified as follows,

$$I = h\varphi_1(Y) = \rho(X_{jk}Y + \phi_1 \qquad (1)$$

In the same way, the process of decoding from the hidden layer to the output layer/reconstruction layer is quantified as follows,

$$Z = h\varphi_2(I) = \rho(X_{km}I + \phi_2) \qquad (2)$$

Where $Y = (y_1, y_2, y_3, ..., y_p)$ implies the input data vector y, and $Z = (z_1, z_2, z_3, ..., z_p)$ signifies the reconstruction (or decoder) vector of the output layer. Whereas $I = (i_1, i_2, i_3, ..., i_n)$ suggests the low dimensional from hidden layer to output layer, $Y \in \Re^p$, $Z \in \Re^p$, $I \in \Re^n$ (p characterizes the input vector dimension and n resembles the number of hidden units). $X_{jk} \in \Re^{n \times p}$ represents the weight connection matrix between the input and the hidden layer and $X_{km} \in \Re^{p \times n}$ signifies the weight connection matrix between the hidden and output layer. The structure of WAE is presented in Figure 2.

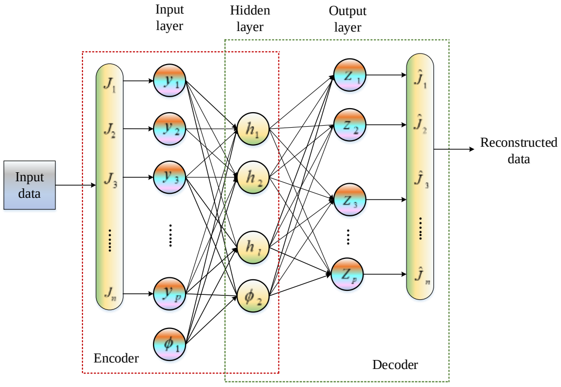


Fig 2: Structure of Weighted Autoencoder

During model training, $X_{km} = X_{jk}^{U}$ generally exist to reconstruct the input data as correctly as possible while minimizing resource consumption. The bias vector of the input and hidden layer is resembled as $\phi_1 \in \Re^{p \times 1}$ and $\phi_2 \in \Re^{n \times 1}$. The activation function of hidden layer neurons and the output layer neurons can be specified as $h\varphi_1(.)$ and $h\varphi_2(.)$, which maps the summation result of the network $[0,1]$. Here, the sigmoid function has employed as the activation function,

$$h\phi_1(.) = h\phi_2(.) = \frac{1}{1 + e^{-y}} \qquad (3)$$

The error between the original data and the output reconstructed data has been minimized by altering the encoder and decoder parameters. Due to hidden layer units, the data output is currently an optimal low-dimensional representation of the original data and includes all data contained in the original data. Furthermore, the mean squared-error function has been used by the reconstruction error function $K_F(X, \phi)$ between I and Z. It is given as follows,

$$K_F(X, \phi) = \frac{1}{2P} \sum_{s=1}^{P} \left\| Z^{(s)} - Y^{(s)} \right\|^2 \qquad (4)$$

Where P indicates the number of input samples.

3.2.3. Attention-based Bi-directional long short term memory

Recently, Recurrent Neural Networks (RNNs) have been broadly employed in several Natural Language Processing (NLP) tasks. LSTM is considered the variant of RNN that efficiently eliminates the gradient loss, and gradient explosion occurs through RNN-based networks for long-range dependencies. Also, it has attained better results for sequence-based semantic approaches. However, it fails to extract the contextual information and has less capability to extract the local context information. Therefore, Bi-LSTM [29] is a hybrid with an attention mechanism called attention-based Bi-LSTM (Attn_BiLSTM) to enhance performance and accuracy. The primary idea of Bi-LSTM is to acquire the context information from the input sequence by 2 hidden layers of LSTM. Subsequently, by concatenating two hidden vectors of LSTM, the context vector has been constructed as $H_u = [\overrightarrow{H_u}; \overleftarrow{H_u}]$. In Attn_BiLSTM, the BiLSTM has been employed to extract the features in the context. The general BiLSTM function is given below as follows,

$$\begin{bmatrix} \tilde{d}_u \\ g_u \\ p_u \\ j_u \end{bmatrix} = \begin{bmatrix} \varsigma \\ \varsigma \\ \varsigma \\ \tanh \end{bmatrix} \left(\mathrm{N}^U \begin{bmatrix} Y_u \\ i_{u-1} \end{bmatrix} + c \right) \quad (5)$$

$$d_u = j_u \odot \tilde{d}_u + g_u \odot d_{u-1} \quad (6)$$

$$i_u = p_u \odot \tanh(d_u) \quad (7)$$

Where, N^U and c represent the trainable parameters, $\odot$ indicates the dot product function, $\varsigma()$ signifies the sigmoid function, Y_u resembles the input vector, p_u, j_u and g_u imply the output, input and forget gates considerably.

The final feature representation from the BiLSTM has passed into self-attention for selecting the highly interdependent features. Initially, the attention layer acquires the weight of each feature by computing the relationship between each key matrix $L \in \Re^{p \times 2f_i}$ and feature matrix $R \in \Re^{p \times 2f_i}$, where f_i represents the hidden units' dimension of BiLSTM. Perceptron, concatenation and dot product were commonly used as similarity functions. Next, the similarity score from the previous level is normalized, and the weights are calculated using the softmax function. Lastly, the weighted summation of weights and value matrix $W \in \Re^{p \times 2f_i}$ are determined to attain the final attention. The scaled dot product function has been employed to evaluate the similarity. Then, the attention has been mathematically formulated as,

$$Atten(R, L, W) = Soft\max\left(\frac{RL^U}{\sqrt{f}}\right)W \quad (8)$$

The multi-head attention can be termed as the perfection of the attention mechanism. Initially, R, L and W execute i times linearly mapping by utilizing various weight matrices. After that, it executes the generation output of the attention function for R, L and W acquired from every linear mapping in parallel. Subsequently, the new representation has been considered a linear mapping of the concatenation. The function has been specified as,

$$Hd_j = Atten\left(RX_j^R, LX_j^L, WX_j^W\right) \quad (9)$$

$$I' = \left(Hd_j \oplus ..... \oplus Hd_j\right)X_p \quad (10)$$

The trainable projection parameters are indicated as $X_j^R \in \Re^{2f_i \times f_l}$, $X_j^L \in \Re^{2f_i \times f_l}$ and $X_j^W \in \Re^{2f_i \times f_l}$, and $f_l = {}^{2f_i}\!/_{i}$. $X_p \in \Re^{2f_i \times 2f_l}$ resembles the trainable parameter. In Attn_BiLSTM, the self-attention mechanism has been utilized and set $R = L = W = I$, which intends to acquire the long dependencies in the input, which I resembles the BiLSTM output, respectively.

3.3 Generation of Image modal

3.3.1 Pre-processing using Min-Max Gaussian filtering approach

In medical image processing, pre-processing has attained greater attention because of various factors like image acquisition, low contrast, complex background and illumination. Effective pre-processing plays a major role in determining better classification accuracy. The MR images hold large numbers of speckle noise. Various techniques have recently been employed to deal with denoising. However, the Gaussian filter cannot lose the peak signals and only reduces the variance between the fall and rise of signals. For the centre limit theorem, the combined effects of noise tend toward Gaussian. Henceforth a Gaussian filter is used to remove the speckle noise and other noise when present [30]. The Gaussian filter creates the edge blur, but this problem was solved using different mean and standard deviation ranges.

Here, Ω be the chosen data sets, $J(j,k)$ indicates the input image with dimension $N \times P$, where $J(j,k) \in \Omega$. The Gaussian filter can be stated as follows,

$$J^H(j,k) = J'\left(\frac{1}{2\pi\varsigma^2} e^{\left({}^{(-j^2-k^2)}\!/_{2\varsigma^2}\right)}\right) \quad (11)$$

$$\varsigma^2 = F(Z^2) - [F(Z)]^2 \quad (12)$$

Where $J^H(j,k)$ represents the denoising image by utilizing the Gaussian function, ς^2 specifies the variance of the noisy image, j and k state the distance from X and Y axes, and $Z \in (j,k)$, significantly.

Then the denoising image size has minimized depending on the range of image intensity. Size normalization is stated as a process in which the standard resolution of input images has been resized. This process converts the improved image $J^{H}(j,k)$ with different dimensions containing ranges of intensity from $J^{H}(j,k):(Z \subseteq \Re^{2}) \rightarrow \{J_{Mini} \ to \ J_{Maxi}\}$ to a new image with new intensities as $J_{new}(j,k):(Z \subseteq \Re^{2}) \rightarrow \{J_{Mini.new} \ to \ J_{Maxi.new}\}$. The process of normalization is linearly carried out and specified as,

$$J_{new}(j,k) = \frac{(J^{H} - J_{Mini})(J_{Maxi.new} - J_{Mini.new})}{J_{Maxi.} - J_{Mini.}} + J_{Mini.new} \quad (13)$$

Where $J_{Maxi.}$ indicates the maximum intensity before normalization, J_{Mini} specifies the minimum intensity before normalization and $J_{Maxi.new}$, $J_{Mini.new}$ implies the new maximum and minimum intensities after normalization. Afterwards, the resized image was fed to the feature extraction process for Alzheimer's disease classification.

3.3.2 Feature extraction using convolutional capsule *network_Chimp optimization algorithm (Conv_Capsnet_COA)*

For the generation of image modal, the features from the pre-processed images are extracted using the convolutional capsule network_Chimp optimization algorithm (Conv_Capsnet_COA). A capsule network is employed in Conv_Capsnet_COA to maintain the properties and position of objects in an image and model their hierarchical relationships. Considering the Convolutional Neural Network (CNN) [31], the significant information in the data only arises with the pooling layer because the data was forwarded to the subsequent layer through the pooling. However, knowing trivial details for the network may not be possible. In addition, CNN produces a scalar value in the neural output, and the capsule network [32] produces a vectorial output of similar size with different routings. The routings of a vector are mostly represented as image parameters. Also, CNN generally utilizes scalar input activation functions similar to tangent, sigmoid, and rectified linear units (ReLU). While the capsule network uses a vector action function term as squeezing, which is given in the following equation,

$$W_{k} = \frac{\|T_{k}\|^{2}}{1+\|T_{k}\|^{2}} \frac{T_{k}}{\|T_{k}\|} \quad (14)$$

Where W_{k} states the output of the capsule k and T_{k} states the total capsule input. The total input value of the capsule T_{k} is stated through the weighted sum of the predicted vectors $(V_{k|j})$ positioned in the lower layer of the capsule, excluding the initial layer of the capsule networks. Multiplying the capsule in the lower layer through its weight matrix (X_{jk}) and output (Q_{j}).

$$T_{k} = \sum_{j} c_{jk} v_{k|j} \quad (15)$$

$$v_{k|j} = X_{jk} Q_{j} \quad (16)$$

Where c_{jk} implies the coefficient obtained through the dynamic routing process, and it is given below as follows,

$$c_{jk} = \frac{\exp(b_{jk})}{\sum_{l} \exp(b_{jl})} \quad (17)$$

Where the log probability is specified as b_{jk}. The sum of the correlation coefficient between the top layer capsule and capsule j is considered 1, and softmax is used to determine the log prior probability.

A margin loss is introduced into the capsule network to determine whether the features are extracted. It is computed as follows:

$$M_{l} = U_{l} Maxi(0, n^{+} - \|w_{l}\|)^{2} + \kappa(1-U_{l}) Maxi(0, \|w_{l}\| - n^{-})^{2} \quad (18)$$

Where the value U_{l} is specified as 1 when the significant feature l exist. $n^{-} = 0.1$, $n^{+} = 0.1$ imply the hyper-parameters. In the capsule network, the calculated vector length represents the probability of being in the image part, and the direction of the vector encompasses the parameter information like size, position, colour, texture and so on. Figure 3 presents the architecture of the Conv_Capsnet_COA model. This Conv_Capsnet_COA model comprises several convolutional layers, one primary layer, three fully connected layers and one-digit layer. Including several convolution layers to present the primary layer with a more efficient feature map as input. The architecture of Conv_Capsnet_COA based feature extractor model is depicted in Figure 3.

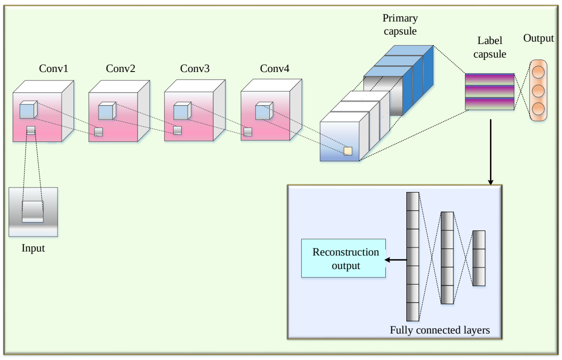


Fig 3: Architecture of convolutional capsule network model

Furthermore, to efficiently stimulate the detection accuracy, the resultant weights of the Conv_Capsnet_COA model were updated through the execution of chimp optimization algorithm (COA). The major benefits of COA [33] include accomplishing an optimal solution and scalability, alleviating trapping in local optima and resolving slow convergence speed for a high-dimensional neural network. Therefore, COA is implemented to update the output weights of the network model.

Here, the position weighting of each individual is updated using COA for accurate classification of CN/AD, MCI/AD and CN/MCI classes. The resulting expression signifies the updated form.

$$y_{chi}(u+1) = \begin{cases} y_{pre}(u) - b.e & if\ \eta < 0.5 \\ Chaotic\_value & if\ \eta > 0.5 \end{cases} \tag{19}$$

Where, η represents the random number in $[0,1]$, u specifies the current iteration, e implies the driving search agent, y_{prey} indicates the position of search agent and b represents the coefficient vector. $Chaotic\_value$ states the chaotic value obtain using the chaotic tent map. By implementing the Conv_Capsnet_COA model, the detection of AD classes is precisely detected with minimum time complexities. The algorithm of Conv_Capsnet_COA is described in Table 2.

Table 2: Algorithm of Conv_Capsnet_COA

```
Input: Preprocessed MRI image $Y$
Initialize the population $y_j(j = 1,2,3,...,p)$, training variable: loss, learning rate
For epoch = 0, 1, 2, 3,… do
        Sample min-batch $Y$
          Extract feature $Y^{'} \leftarrow Convolution$
           Update the variables by gradient: $Adam(loss, learning\ rate)$
 For weight updation, do
      Compute the location of each individual
        Separate the individuals randomly into independent sets
        Determine the fitness of each individual
         while $(u < \max imum number of\ iteration)$
    for each individual:
           Extract the individual's set
             Apply its group policy for updating the coefficient vector ($n$ and
              $d$ ), and $g$ (minimizes nonlinearly from 2.5 to 0)
            Compute $b$ and $e$ use $g$, $n$ and $d$
    end for
        for each search agent
          if $(\eta < 0.5)$
           if $(|b| < 1)$
                  Update the location of current search agent
                         else if $(|b| > 1)$
                              Choose a search agent randomly
                        end if
                      end if $(\eta > 0.5)$
                              Update the current search position
                   end if
                end for
        $u = u + 1$
      end while
     end for
 Stop
Output: Extracted feature
```

3.4 Feature fusion based on deep fusion strategy

After the feature extraction process, the extracted features from the IoT data model and the image model are fussed using Deep Fusion Strategy (DFS). The DFS comprises deep CNN (Deep_CNN) for learning the extracted feature vector and further residual modules employed to tackle gradient vanishing and overfitting problems. Lastly, the learned features are fused and fed to the softmax layer to classify AD. Deep_CNN can automatically learn the extracted features by training the dataset. The convolution layer, pooling layer, and fully-connected layers are three kinds of layers that generally exist in Deep_CNN. The convolutional layers are employed to determine the local relationships in their inputs, whereas the input dimensions are gradually minimized using the pooling layer. Eventually, the fully-connected layers were applied to classify the input patterns into different classes: CN/AD, MCI/AD and CN/MCI. Given the extracted input AD data be $Y = (y_1, y_2, ..., y_p)$. After convolution with fewer layers, the generated feature vector can be expressed as $M = (m_1, m_2, ..., m_p)$. M specifies the shallow features. The expression for convolution can be given as follows,

$$T = \operatorname{Re} LU\left(\sum_{k=1}^{p}\left(L_k * y_k + c_k\right)\right) \qquad (20)$$

Where T resembles the feature vector constructed from a single convolution, $\operatorname{Re} LU()$ specifies the activation function, C represents the bias vector and L states the convolution kernel.

For images, the generated feature vector after convolution with fewer layers can be expressed as $I = (i_1, i_2, ..., i_p)$. I signifies the deep features, and the formula for the convolution is provided in the below equation. The convolution layers minimize the training complexity and improve the model's generalization ability through transition shift, weight share and sparse connection. Assume the provided input data be $Y = (y_1, y_2, ..., y_p)$ with the size of $8\times8\times1$. The filter has comprised of a bias vector $C = (c_1, c_2, ..., c_p)$ and a weight vector $X = (x_1, x_2, ..., x_p)$. The convolution kernel (L) size has been set to 1×3; then after convolution, the j^{th} feature vector a_k is computed as follows.

$$a_k = \sum_{l=1}^{p} \operatorname{Re} LU\left(y_l * w_k + c_k\right) \qquad (21)$$

Then, in DFS, a residual network is used to solve gradient vanishing and overfitting problems. The residual network learns the deep features and use skip connection to the convergent loss function in training. The structure of DFS is demonstrated in Figure 4.

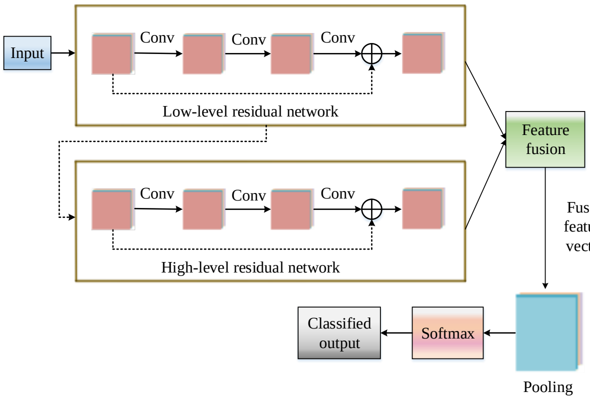


Fig 4: Structure of deep fusion strategy

Given the AD data Y, after the two-layer convolution Y, the obtained convolution outcome is $D(Y)$. Then, using the residual network, the obtained result is $S(Y)$. $S(Y)$ can be expressed as follows,

$$S(Y) = D(Y) + Y \qquad (22)$$

Here, ReLU is utilized as an activation function. Compared to conventional activation functions like tanh and logistic sigmoid, ReLU can eliminate gradient vanishing and gradient explosion. In addition, it can reduce computational complexity. $S(Y)$ can be specified as

$$S(Y) = \operatorname{Re} LU\left(\operatorname{Re} LU\left(Y * X_1 + C_1\right) * X_2 + C_2\right) + Y \qquad (23)$$

Where C_j represents the bias vector of j^{th} convolution layer and X_j characterizes the weight of j^{th} convolution layer.

The feature fusion process not only deliberates high semantic information but also considers more detailed information, allowing it to improve classification performance. Considering the learned features of IoT data and image models be $M = (m_1, m_2, ..., m_p)$ and $I = (i_1, i_2, ..., i_p)$, they were combined and fused into a new vector as $G = (g_1, g_2, ..., g_p)$. The expression for the fusion can be provided as,

$$g_j = m_j \ddagger i_j \qquad (24)$$

where ‡ is used to combine two features.

DFS initially combines two feature vectors as well as fuses them as $G = (g_1, g_2, ..., g_p)$. After that, G has pooled on the pooling layer. The main purpose of the pooling vector is to compress the input vector. In DFS, max pooling is used as the pooling function, and the constructed feature vector can be represented as $N = (n_1, n_2, ..., n_p)$. N is calculated using the below formula.

$$N_j = Maxi_{k=1}^{l}(G_j), \quad j = 1,2,3,...., p \qquad (25)$$

Where l indicates the size of pooling region.

Finally, the feature vector $Q = (q_1, q_2, ..., q_p)$, trained through Deep CNN, is reflected as the input of softmax for classification. In the training process, the training epoch is set to 50, the learning rate to 0.01 and the batch size 100. In addition, $Adam()$ is employed to optimize the training process, whereas the cross entropy loss function has employed as the loss function.

4. Experimental Setup

In this subdivision, the results of the proposed DL-ADDM are estimated for detecting AD at an early stage based on multi-model feature processing. The proposed DL-ADDM for AD detection using IoT data and images is being simulated on the standalone computer through the python platform. It has been configured with an Intel® core (TM) i5-4670S – CPU @ 3.10 GHz processor with 16 GB of main memory over a 64-bit windows 10 operating system. Early detection of AD intends to discover a better chance of benefiting from treatment. The data has separated into training, testing, and validation before constructing the model. In the initial stage, pre-processing is carried out for IoT data and images; then, the pre-processed data is fed into the feature extraction process. Subsequently, the proposed model fuses the extracted feature vectors and classifies them into various classes as MCI/AD, CN/MCI and CN/AD. The following sections designate the dataset description, evaluation metrics, and performance analysis of the AD detection model. Table 3 describes the hyper-parameter settings of proposed model.

Table 3: Hyper-parameters settings

S. No	Hyper-parameters	Values
1	Initial learning rate	0.01
2	Max epochs	50
3	Batch size	100
4	Optimizer	Adam
5	Activation function	ReLU

The initial learning rate is set to be 0.01 as it regulates the neural network weight on concerning the loss gradient. The Adam optimizer is chosen that updates the learning weight for every network weight individually. The optimizer helps to choose an ideal solution that enhances the classification accuracy. The epochs define the total number of iterations that the data takes place to maximize the network training ability. The ReLU activation function is employed to introduce the non-linearity property that solves the vanishing gradient issue. Because of considering the batch size, the training complexity gets reduced.

5. Results and Discussion

This section reveals the evaluation metrics used and baseline model comparison analysis for proposed and existing methods. The analysis were done in terms of confusion matrix, ROC and so on to prove the model superiority of proposed method.

5.1 Evaluation metrics

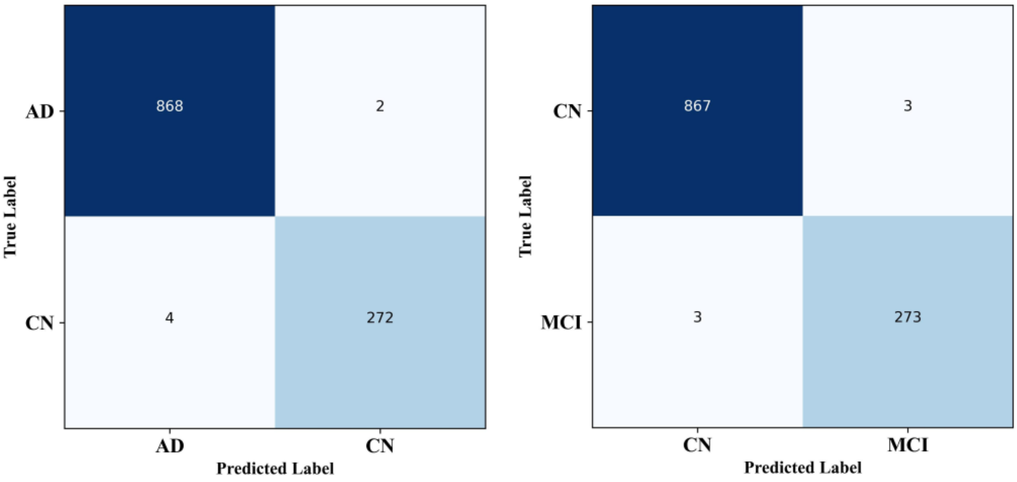
The proposed DL-ADDM uses performance metrics, including accuracy, specificity, recall, precision, and f-score [34-35], to define the performance. The obtained performance of the proposed DL-ADDM is investigated with state-of-art methods to reveal the effectiveness of the advanced model.

5.2 Baseline model comparison analysis

The investigation implications of the proposed DL-ADDM are discussed in this section. The results of the proposed DL-ADDM will be compared to existing architectures to validate the rate of improvement in terms of different performance metrics. This resulted in a brief discussion of the assessment in the following subsections

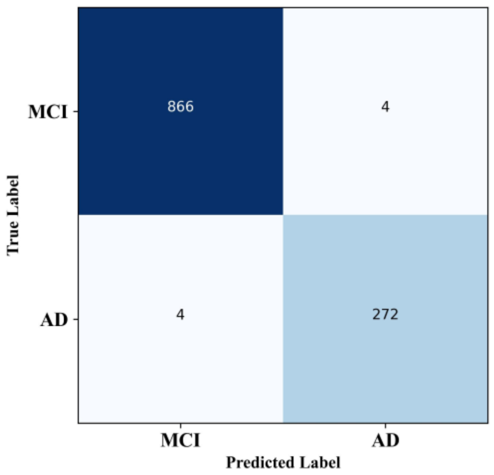
5.2.1 Confusion matrix analysis

The proposed DL-ADDM will be implemented significantly using the main dataset for cognitive testing and the Kaggle dataset to detect AD. Current models have accompanied DFS to offer performance efficiency. Figure 5 shows the confusion matrix of the proposed DL-ADDM to differentiate from AD. The proposed model has applied the dataset's data to detect AD at an early stage considerably. The confusion matrix delivers a 2×2 matrix with Tp''', Tn''', Fp''' and Fn'''. The figure shows that the proposed DL-ADDM detected the class labels of AD with extreme accuracy and deteriorated the false prediction rate.



(a) CN/AD

(b) CN/MCI

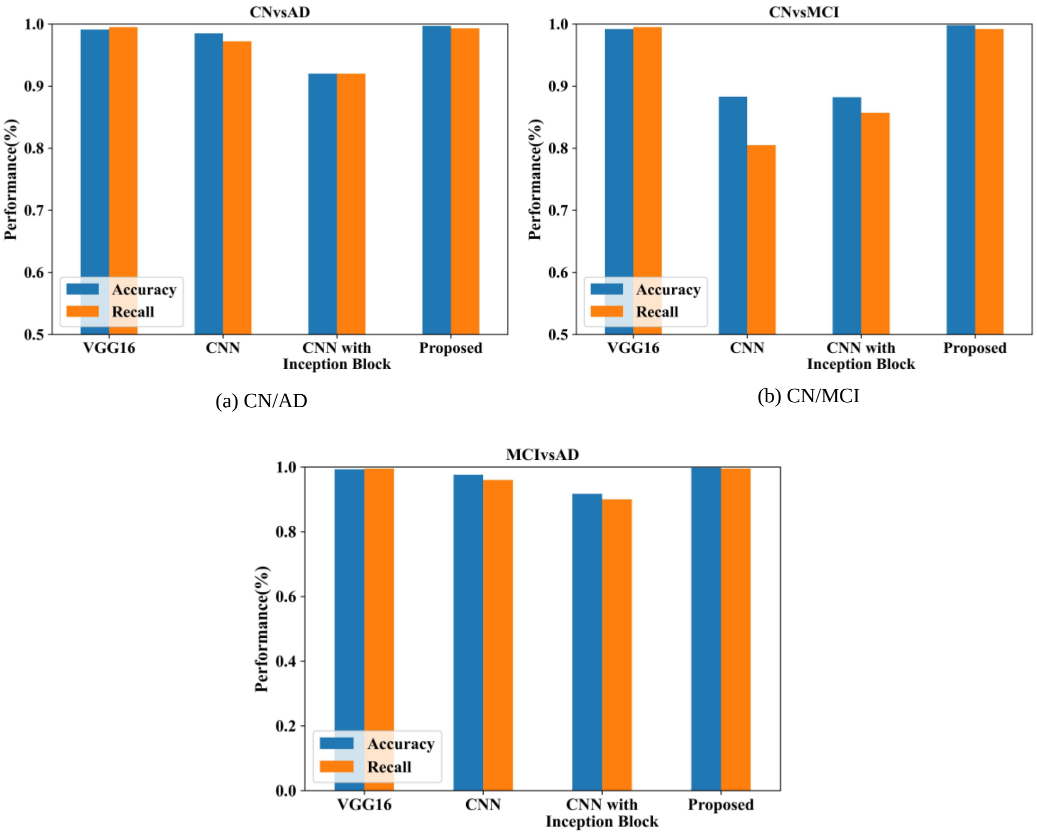


(c) MCI/AD

Fig 5: Confusion matrix of the proposed AD detection model

5.2.2 Comparison with DL classifiers:

The comparative analysis of accuracy, recall and specificity of the proposed DL-ADDM for CN/AD, MCI/AD and CN/MCI using DL classifiers is presented in Figure 6. The proposed DL-ADDM also compared the performance of recent DL architectures [36] such as VGG-16, CNN and CNN with inception block. Correspondingly, it is demonstrated that the proposed DL-ADDM has the proficiency to discriminate the false rates from the input data.



(a) CN/AD

(b) CN/MCI

(c) MCI/AD

Fig 6: Comparative analysis of accuracy and recall of proposed DL-ADDM with DL classifiers

In addition, it is perceived that the proposed methodology can potentially discover AD by mapping the target values and the input considerably. On the other hand, the proposed DL-ADDM achieved better accuracy due to selecting the best feature extraction and feature fusion strategy, which includes Deep_CNN and Residual Network. Also, it is understood that the proposed DL-ADDM has appropriately predicted the wrong value rates for AD detection.

Furthermore, the proposed DL-ADDM can exactly predict the class labels of AD for the input as well as define the exact positive rates. Even through the accuracy of VGG-16 is almost similar to the proposed model, the feature learning complexity tends to be very high in VGG-16 compared to the proposed model. Table 4 indicates the comparison of different classes using DL methodologies.

Table 4: Comparison of CN/AD, MCI/AD and CN/MCI using DL architectures

Classifier	Class	Accuracy (%)	Recall (%)	Specificity (%)	Precision (%)
VGG16	CN/AD	99.14	99.5	-	99.5
	CN/MCI	99.2	99.5	-	99.5
	MCI/AD	99.3	99.5	-	99.5
CNN	CN/AD	98.59	97.22	100	-
	CN/MCI	88.37	80.56	94	-
	MCI/AD	97.62	96	100	-
CNN with inception block	CN/AD	92	92	92	93.1
	CN/MCI	88.23	85.71	85.71	91.665
	MCI/AD	91.76	90	90	93.85
Proposed	**CN/AD**	**99.5**	**99.4**	**98.3**	**99.3**
	CN/MCI	**99.3**	**98.5**	**99.5**	**99.2**
	MCI/AD	**99.6**	**95.8**	**99.8**	**100**

5.2.3 Accuracy comparison with and without fusion

The accuracy of the proposed model with respect to IoT data is obtained to be 98.56% whereas the accuracy in terms of Image data is obtained to be 97.96% before fusion. On merging the features extracted from IoT and image data, the overall accuracy of the proposed model is obtained to be 99.5% for CN/AD classification, 99.3% for CN/MCI classification and 99.6% for MCI/AD classification. The fundamental benefit of merging the multimodal features is, identification of compact salient features set that effectively maximizes the classification accuracy with minimal rates of false positives. Through the adoption of DFS strategy, most of the irrelevant features can be diminished and thereby the overall classification modal efficiency can be enhanced.

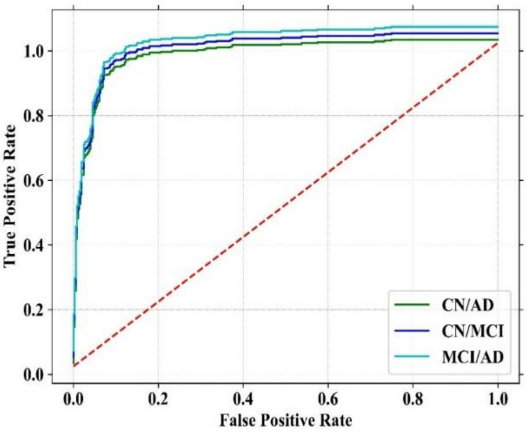


Fig 7: ROC analysis of proposed DL-ADDM

5.2.4 Evaluation of ROC

In this section, the receiver operating curve (ROC) of the proposed DL-ADDM has been analyzed. In AD detection, the ROC analysis has executed against a true positive rate (PR) and false PR to discriminate the value of the area under the curve (AUC). Figure 7 demonstrates the ROC evaluation for the proposed AD detection model. In the ROC graph, the false PR is designated by the x-axis, and the y-axis shows the true PR. The proposed DL-ADDM can differentiate among the target classes of AD. The proposed DL-ADDM has obtained a better AUC value by deepening the probability of discovering the class labels resulting from the input parameters. Henceforth, the proposed DL-ADDM is applicable for significantly classifying AD.

5.2.5 Evaluation in terms of model accuracy

In AD detection, assessing accuracy and loss helps estimate the efficiency of the AD detection model. For training and testing the dataset, the proposed DL-ADDM is better illustrated in terms of accuracy and loss. The proposed DL-ADDM has achieved maximum accuracy and is better than traditional approaches. Figure 8 shows the training and testing accuracy validation for the proposed DL-ADDM in terms of MCI/AD, CN/MCI and CN/AD. The proposed DL-ADDM has achieved excellent results for both training and testing instances. The accuracy score obtained for the proposed DL-ADDM is analyzed with the importance of training and testing samples. The figure shows that the training and testing curves are comparable, indicating the effectiveness of the proposed DL-ADDM in influencing the test patterns.

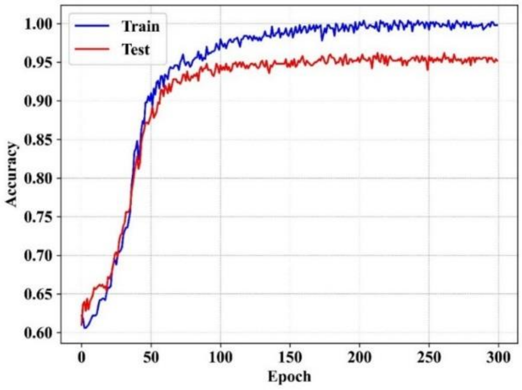


Fig 8: Training and testing accuracy of the proposed AD detection model

As seen in figure 8, the error value leads to the gap between the training and testing curve. For maximum epochs, the non-conformity between the training and testing curves has diminished, and it brings to increase the performance rate of AD detection. Furthermore, the proposed DL-ADDM has adept less over-fitting concerns. The proposed DL-ADDM has trained for different epochs to assess AD detection and classification ability. Additionally, the proposed DL-ADDM has trained well and accomplished better outcomes while maximizing the number of epochs.

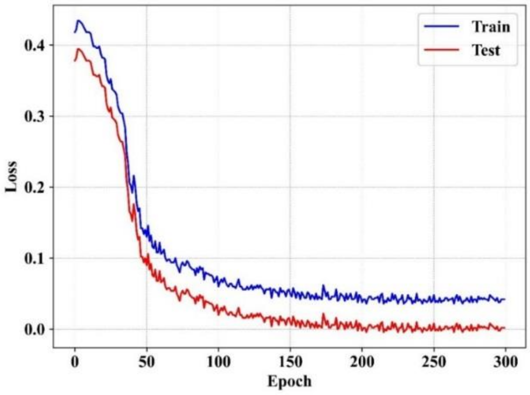


Fig 9: Training and testing loss of proposed AD detection model

Then, the obtained loss values of the proposed DL-ADDM were significantly evaluated with the consequences of training and testing samples. Figure 9 validates the training and testing loss of the proposed DL-ADDM by varying the number of epochs. The figure demonstrates that the training and validation loss of the proposed DL-ADDM has contracted over time, and the test loss holds slightly above the training loss. As a result, the proposed DL-ADDM has accomplished a balance between the underfitting and overfitting complications.

5.2.6 Discussion

To promote early treatment, exact AD classification using multimodal data possess a major requirement in medical field. The classification of multimodal data systems are contemplated as a significant field in AD classification. In the proposed DL-ADDM model, novel methodologies are implemented for AD classification through the acquisition of IoT and Image data. The accurate classification outcomes are promoted whereas 99.5% of accuracy is attained for CN/AD classification, 99.3% for CN/MCI classification and 99.6% for MCI/AD classification. The pre-processing step is carried out using min-max normalization, missing value imputation and data validation in case of IoT data. The most discriminant features of IoT data are extracted using WAE and Attn-BiLSTM. In case of image data, pre-processing step is undertaken using min-max gaussian filter whereas the features are extracted using Conv_Capsnet_COA model. The obtained features of multimodal data are fused together to generate a feature vector for the further classification of AD using Softmax classifier.

Precise classification outcomes can be obtained by the proposed model on considering the performance metrics like accuracy, precision, recall and specificity. Significant outcomes can be attained by the proposed model because of certain merits like improved network performance, less error rates, higher training capability, prioritized significant features and efficient fusion of features. The proposed outcomes are assessed through comparison with existing DL based classifier models like CNN, VGG16 and CNN with inception block. The outcomes exposed that the proposed DL-ADDM model outperformed the existing classifiers. The surveyed method in Qiu, Shangran et al. [18] does not describe a suitable solution for precise AD classification. In El-Sappagh et al. [19], the drawback of non-interpretable decisions and increased false positives are noticed. The accuracy is improved in Divya and Shantha Selva Kumari [20] but imbalance outcomes are noticed because of huge dataset consideration. Due to the accumulation of larger datasets, huge computational complexity is noticed in El-Sappagh et al. [21]. Because of higher training complexity in Helaly et al. [22], the training time tends to be high. As the consideration of features are ineffective in Zhang et al. [23], the model tends to be computationally expensive. Less training ability is noticed in Sun et al. [24] because of considering non-relevant features. Because of these complexities, the existing models promotes degraded performance whereas the proposed model tends to be highly superior by conquering existing drawbacks.

6. Conclusion

This paper proposes a new multi-model feature processing approach using DL for detecting AD. Primarily, the sensed medical-related IoT data and images are provided for pre-processing. Next, feature extraction is performed using a weighted auto encoder (WAE) with attention-based bi-directional long short term memory (Attn_BiLSTM) for IoT data and convolutional capsule network_chimp optimization algorithm (Conv_Capsnet_COA) for an image. Subsequently, the extracted features from IoT data and image model are fused using deep fusion strategy (DFS), and fused feature vectors are provided into softmax for classification. The performance of DL-ADDM is compared with recent existing ML and DL classifiers to determine the efficiency of AD detection. The proposed multimodal AD detection model has accomplished better accuracy of 99.5% for CN/AD, 99.3% for CN/MCI and 99.6% for MCI/AD; recall of 99.4% for CN/AD, 98.5% for CN/MCI and 95.8% for MCI/AD; and specificity of 98.3% for CN/AD, 99.5% for CN/MCI and 99.8% for MCI/AD, respectively. Even with better performance, the proposed method must search for the most discrimination features to improve the detection performance. In future, the most discriminating features will be analysed using hybrid optimization algorithms.

Compliance with Ethical Standards

Funding: No funding is provided for the preparation of manuscript.

Conflict of Interest: Authors declare that they have no conflict of interest.

Ethical Approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Consent to participate: All the authors involved have agreed to participate in this submitted article.

Consent to Publish: All the authors involved in this manuscript give full consent for publication of this submitted article.

Authors Contributions: All authors read and approved the final manuscript.

Data Availability Statement: Data sharing is not applicable to this article.

References

[1] J. Wen, E. Thibeau-Sutre, M. Diaz-Melo, J. Samper-González, A. Routier, S. Bottani, D. Dormont, S. Durrleman, N. Burgos, O. Colliot, "Alzheimer's Disease Neuroimaging Initiative. Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation". *Medical image analysis* vol. 63, pp. 101694, 2020.

[2] M. Liu, F. Li, H. Yan, K. Wang, Y. Ma, L. Shen, M. Xu, "Alzheimer's Disease Neuroimaging Initiative. A multi-model deep convolutional neural network for automatic hippocampus segmentation and classification in Alzheimer's disease". *Neuroimage* vol. 208, pp. 116459, 2020.

[3] N. Yamanakkanavar, J.Y. Choi, B. Lee, "MRI segmentation and classification of human brain using deep learning for diagnosis of Alzheimer's disease: a survey". *Sensors* vol. 20, no. 11, pp. 3243, 2020.

[4] L.F. Samhan, A.H. Alfarra, S.S. Abu-Naser, "Classification of Alzheimer's disease using convolutional neural networks".

A. Mehmood, M. Maqsood, M. Bashir, Y. Shuyuan, "A deep Siamese convolution neural network for multi-class classification of Alzheimer disease". *Brain sciences* vol. 10, no. 2, pp. 84, 2020.

[5] H.R. Almadhoun, S.S. Abu-Naser, "Classification of Alzheimer's disease using traditional classifiers with pre-trained CNN".

[6] J. Venugopalan, L. Tong, H.R. Hassanzadeh, M.D. Wang, "Multimodal deep learning models for early detection of Alzheimer's disease stage". *Scientific reports* vol. 11, no. 1, pp. 3254.

[7] E.E. Bron, S. Klein, J.M. Papma, L.C. Jiskoot, V. Venkatraghavan, J. Linders, P. Aalten, P.P. De Deyn, G.J. Biessels, J.A. Claassen, H.A. Middelkoop, "Cross-cohort generalizability of deep and conventional machine learning for MRI-based diagnosis and prediction of Alzheimer's disease". *NeuroImage: Clinical* vol. 31, pp. 102712, 2021.

[8] Y. AbdulAzeem, W.M. Bahgat, M. Badawy, "A CNN based framework for classification of Alzheimer's disease". *Neural Computing and Applications* vol. 33, pp. 10415-28, 2021.

[9] M. Shahbaz, S. Ali, A. Guergachi, A. Niazi, A. Umer, "Classification of Alzheimer's Disease using Machine Learning Techniques". *In Data* pp. 296-303, 2019.

[10] T. Jo, K. Nho, A.J. Saykin, "Deep learning in Alzheimer's disease: diagnostic classification and prognostic prediction using neuroimaging data". *Frontiers in aging neuroscience* vol. 11, pp. 220, 2019.

[11] S. Basaia, F. Agosta, L. Wagner, E. Canu, G. Magnani, R. Santangelo, M. Filippi, "Alzheimer's Disease Neuroimaging Initiative. Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks". *NeuroImage: Clinical* vol. 21, pp. 101645, 2019.

[12] S. Spasov, L. Passamonti, A. Duggento, P. Lio, N. Toschi, "Alzheimer's Disease Neuroimaging Initiative. A parameter-efficient deep learning approach to predict conversion from mild cognitive impairment to Alzheimer's disease". *Neuroimage* vol. 189, pp. 276-87, 2019.

[13] F.J. Martinez-Murcia, A. Ortiz, J.M. Gorriz, J. Ramirez, D. Castillo-Barnes, "Studying the manifold structure of Alzheimer's disease: a deep learning approach using convolutional autoencoders". *IEEE journal of biomedical and health informatics* vol. 24, no. 1, pp. 17-26, 2019.

[14] K. Oh, Y.C. Chung, K.W. Kim, W.S. Kim, I.S. Oh, "Classification and visualization of Alzheimer's disease using volumetric convolutional neural network

and transfer learning". *Scientific Reports* vol. 9, no. 1, pp. 18150, 2019.

[15] R. Jain, N. Jain, A. Aggarwal, D.J. Hemanth, "Convolutional neural network based Alzheimer's disease classification from magnetic resonance brain images". *Cognitive Systems Research* vol. 57, pp. 147-59, 2019.

[16] Z. Tang, K.V. Chuang, C. DeCarli, L.W. Jin, L. Beckett, M.J. Keiser, B.N. Dugger, "Interpretable classification of Alzheimer's disease pathologies with a convolutional neural network pipeline". *Nature communications* vol. 10, no. 1, pp. 2173, 2019.

[17] S. Qiu, G.H. Chang, M. Panagia, D.M. Gopal, R. Au, V.B. Kolachalama, "Fusion of deep learning models of MRI scans, Mini–Mental State Examination, and logical memory test enhances diagnosis of mild cognitive impairment". *Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring* vol. 10, pp. 737-49, 2018.

[18] S. El-Sappagh, H. Saleh, F. Ali, E. Amer, T. Abuhmed, "Two-stage deep learning model for Alzheimer's disease detection and prediction of the mild cognitive impairment time". *Neural Computing and Applications* vol. 34, no. 17, pp. 14487-509, 2022.

[19] R. Divya, R.S.S. Kumari, "Alzheimer's disease Neuroimaging Initiative. Genetic algorithm with logistic regression feature selection for Alzheimer's disease classification". *Neural Computing and Applications* vol. 33, no. 14, pp. 8435-44, 2021.

[20] S. El-Sappagh, H. Saleh, R. Sahal, T. Abuhmed, S.R. Islam, F. Ali, E. Amer, "Alzheimer's disease progression detection model based on an early fusion of cost-effective multimodal data". *Future Generation Computer Systems* vol. 115, pp. 680-99, 2021.

[21] H.A. Helaly, M. Badawy, A.Y. Haikal, "Deep learning approach for early detection of Alzheimer's disease". *Cognitive computation* pp. 1-7, 2021.

[22] J. Zhang, B. Zheng, A. Gao, X. Feng, D. Liang, X. Long, "A 3D densely connected convolution neural network with connection-wise attention mechanism for Alzheimer's disease classification". *Magnetic Resonance Imaging* vol. 78, pp. 119-26, 2021.

[23] H. Sun, A. Wang, W. Wang, C. Liu, "An improved deep residual network prediction model for the early diagnosis of Alzheimer's disease". *Sensors* vol. 21, no. 12, pp. 4182, 2021.

[24] Pirani, Z. Nasreddine, F. Neviani, A. Fabbo, M.B. Rocchi, M. Bertolotti, C. Tulipani, M. Galassi, M.B. Murri, M. Neri, "MoCA 7.1: multicenter validation of the first Italian version of Montreal cognitive assessment". Journal of Alzheimer's Disease Reports vol. 6, no. 1, pp. 509-20, 2022.

[25] https://www.kaggle.com/datasets/sachinkumar413/alzheimer-mri-dataset

[26] A.H. Alamoodi, B.B. Zaidan, A.A. Zaidan, O.S. Albahri, J. Chen, M.A. Chyad, S. Garfan, A.M. Aleesa, "Machine learning-based imputation soft computing approach for large missing scale and non-reference data imputation". *Chaos, Solitons & Fractals* vol. 151, pp. 111236, 2021.

[27] K.P. Chan, M.I. Solihin, C.K. Ang, L.P. Pui, "Experimentation on Spectra Data Regression Using Dense Multilayer Neural Networks with Common Pre-processing". *InEnabling Industry 4.0 through Advances in Mechatronics: Selected Articles from iM3F 2021, Malaysia*. *Singapore: Springer Nature Singapore* pp. 97-112, 2022.

[28] S. Nurmaini, A.E. Tondas, A. Darmawahyuni, M.N. Rachmatullah, J. Effendi, F. Firdaus, B. Tutuko, "Electrocardiogram signal classification for automated delineation using bidirectional long short-term memory". *Informatics in Medicine Unlocked* vol. 22, pp. 100507, 2021.

[29] R.F. Mansour, N.M. Alfar, S. Abdel-Khalek, M. Abdelhaq, R.A. Saeed, R. Alsaqour, Optimal deep learning based fusion model for biomedical image classification. *Expert Systems* vol. 39, no. 3, pp. e12764, 2022.

[30] L. Gaur, U. Bhatia, N.Z. Jhanjhi, G. Muhammad, M. Masud, "Medical image-based detection of COVID-19 using deep convolution neural networks". *Multimedia systems* vol. 29, no. 3, pp. 1729-38, 2023.

[31] T. Kavitha, P.P. Mathai, C. Karthikeyan, M. Ashok, R. Kohar, J. Avanija, S. Neelakandan, "Deep learning based capsule neural network model for breast cancer diagnosis using mammogram images". *Interdisciplinary Sciences: Computational Life Sciences* pp. 1-7, 2021.

[32] M. Khishe, M.R. Mosavi, "Chimp optimization algorithm". *Expert systems with applications* vol. 149, pp. 113338, 2020.

[33] F. Ramzan, M.U. Khan, A. Rehmat, S. Iqbal, T. Saba, A. Rehman, Z. Mehmood, "A deep learning approach for automated diagnosis and multi-class classification of Alzheimer's disease stages using resting-state fMRI and residual neural networks". *Journal of medical systems* vol. 44, pp. 1-6, 2020.

[34] S. Farhan, M.A. Fahiem, H. Tauseef, "An ensemble-of-classifiers based approach for early diagnosis of Alzheimer's disease: classification using structural features of brain images". *Computational and mathematical methods in medicine* vol. 2014, 2014.

[35] S. Basheera, M.S. Ram, "Deep learning based Alzheimer's disease early diagnosis using T2w segmented gray matter MRI". *International Journal of Imaging Systems and Technology* vol. 31, no. 3, pp. 1692-710, 2021.