IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 2, 2022, accepted March 17, 2022, date of publication March 23, 2022, date of current version March 30, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3161749

Stacked Deep Dense Neural Network Model to Predict Alzheimer's Dementia Using Audio Transcript Data

YUSERA FAROOQ KHAN[1], BAIJNATH KAUSHIK[1], MOHAMMAD KHALID IMAM RAHMANI[2], (Senior Member, IEEE), AND MD EZAZ AHMED[2]

[1]School of Computer Science and Engineering, Shri Mata Vaishno Devi University, Katra, Jammu and Kashmir 182320, India
[2]College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia

Corresponding authors: Md Ezaz Ahmed (m.ezaz@seu.edu.sa) and Mohammad Khalid Imam Rahmani (m.rahmani@seu.edu.sa)

ABSTRACT Alzheimer's disease (AD) is caused by cortical degeneration leading to memory loss and dementia. A possible criterion for the early identification of Alzheimer's dementia is to identify the difference between positive and negative linguistic and cognitive abilities of the patients. This study involves the use of Convolutional Neural Network (CNN), designed a hybrid model with CNN & Bidirectional Long-Short Term Memory (Bidirectional LSTM), and proposed a Stacked Deep Dense Neural Network (SDDNN) model for text classification and prediction of Alzheimer's dementia. These models were trained end-to-end using DementiaBank clinical transcript dataset. The transcripts consisted of recorded interviews of Alzheimer's patients with clinical experts. The models were investigated under two settings: Randomly initialized and Glove embedding. Further, hyperparameter optimization was accomplished using GridSearch, which yielded optimal parameters for the design of suitable learning models for most accurate predictions. Other parameters were computed and compared based on AUC, accuracy, specificity, precision, F1 score, and recall. To ensure performance generalization, the classification accuracy was tested using 10-fold cross-validation approach. The performance and classification accuracy of the proposed model was significantly improved to 93.31% when applied with Glove embedding and hyperparameter tuning. This research work will considerably help the clinical experts in early detection and diagnosis of AD.

INDEX TERMS Dementia, audio transcript data, deep learning, convolutional neural network, Bi-LSTM, stacked deep dense neural network.

I. INTRODUCTION

Alzheimer's disease (AD) is the most prevalent form of dementia, a degenerative brain condition primarily affecting older individuals [1]. By comparing the patient's brain functionality at different visits, it has been found that decline in memory and other cognitive function is responsible for primary dementia syndrome [2]. In 2006, 26.6 million patients suffered from AD worldwide. It has been predicted that AD will affect 1.2% of the global population by 2046 [3]. Hence it is a need of the hour to detect AD at an early stage to effectively treat patients suffering from the disease. With the help of machine learning techniques, early detection of Alzheimer's is possible by identifying patient's linguistic patterns within their health care records. In this paper, we investigated effective computational diagnostic models for predicting AD from verbal utterances in transcript data. A potential clinical usefulness of these models is their ability to predict the probable AD. For each patient, local weighted learning is used to modify a classifier model, and sequence computing of biomarkers is highly cost-effective and informative for their diagnosis. The whole data is considered at once by decreasing the cost and number of biomarkers required to attain a positive diagnosis for individual patients. The clinical setup is proved useful along with the contributing effective and personalized detection of AD [4]. Recognition of dementia using automated speech recognition, feature extraction/selection techniques, and classification algorithms has lately become mainstream practise in the field of dementia detection and characterization.

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Alawneh.

Based on some limitations of existing techniques, there is a need to enhance and improvise the state-of-the-art models to effectively deal with existing problem. In this study, we have implemented Deep learning models-CNN and combination of CNN & Bidirectional LSTM and proposed a novel SDDNN model for predicting AD and support normal individuals by identifying their linguistic patterns. We have demonstrated the effectiveness of the proposed model by using stacking of multiple sequence learning models such as CNN, CNN + Bidirectional LSTM and Bidirectional LSTM with attention. The proposed stacking model is used to observe deeper characteristics of conversational patterns and aids in differentiating between two diagnostic groups with extremely similar symptoms. The proposed stacked model has the ability to integrate the capabilities of many different sequence based deep neural network models to provide results that more accurate than existing individual models in ensemble.

A. MOTIVATION

AD in patients probably starts decades ago, prior to the onset of symptoms [5]. So there is a much potential either to prevent it or to slow down its progress when research advancement makes it possible to detect the disease by using biomarkers before its symptoms begin. The motive behind this work is to apply end-to-end deep neural network models to analyze linguistic markers in patients. These markers are analyzed using the spoken languages of AD patients and control normal and then classifying them using Natural Language Processing (NLP) techniques. For mental healthcare professionals, a model that can detect AD from clinical transcription data can be very useful in the trial, quantification and tracking of AD positive cases.

B. CONTRIBUTIONS

This work presents a Deep Dense Neural Network to detect language patterns for classifying AD (AD+) and non-AD (AD-). Following are the contributions of this article:

- In this article, we presented a stacked deep learning neural network models for the early detection and diagnosis of AD from DementiaBank clinical transcript data. The linguistic characteristics of AD patients have been analysed and classified using NLP techniques and deep learning models respectively.
- We investigated the role of "Glove (pretrained) word embedding" and "randomly initialized word embedding."
- Additionally, hyperparameter optimization is performed using Grid Search to get the most accurate predictions for each model.
- We have proposed and designed a novel SDDNN model stacking CNN, CNN + Bidirectional LSTM, and Bidirectional LSTM with attention approach, to detect AD form clinical transcript data using pre-trained Glove embedding and hyperparameter tuning that outperformed state-of-the-art models.
- In the proposed stacked model, we concatenated the outputs of the flattened layers obtained from CNN, CNN + Bidirectional LSTM and Bidirectional LSTM with attention model, and dense layers to obtain the final classification results.
- Our model's predictions were validated using a 10-fold cross-validation procedure.

C. ROAD MAP

The rest of this paper is structured as follows: Section 2 covers the causes of AD and includes an outline of NLP with deep learning for disease detection, as well as a review of previous studies. Section 3 gives a detailed description of the DementiaBank dataset. The technique and methodologies used in this study are discussed in Section 4. Section 5 addresses the experimental results obtained using deep learning models. Conclusion and future directions are covered in the last section.

II. BACKGROUND STUDY

In this section, we have studied the causes and risk factors of AD, framework of its diagnosis, and reviewed the existing work done by various researchers.

A. CAUSES AND RISK FACTORS OF ALZHEIMER DISEASE

Scientists believe that there is not a single cause of AD but it can be developed due to many factors such as the gradual loss of brain cells, changes in lifestyle, and environmental factors. Its risk factors may also include old age, family genetics and heredity that cannot be reversed. The risk of AD is high in cases with history of their parents with the disease [6]. On an average females are more prone to AD than males due to longer life expectancy of the former. People with head trauma are more likely to suffer from AD. People with poor sleep patterns have a higher risk of developing AD. Language difficulties are most prevalent among people having mental health issues such as cognitive impairments, dementia, hearing challenges, or any other congenital brain abnormalities [7], [8]. The damaging usually starts from the memory controlling area of the brain, but its process is initiated years before development of the first symptom. The disease in patients may lead to:

- Cognitive disorders
- Behavioral disorders
- Psychological disorders

AD can be diagnosed at an early stage using a technique known as computer-aided diagnosis (CAD) [9]–[11]. CAD has the capacity to comprehend and interpret massive volumes of natural language data for the purpose of disease diagnosis. The most demanding tasks in language processing includes: voice recognition, sentiment analysis, and language comprehension. The effective ways of taking measures may be explored from data for prediction of AD by comparing deep learning techniques using prognostic models [12], [13]. Various studies have aimed at predicting the time for AD progression under a time-to-event examination set-up and

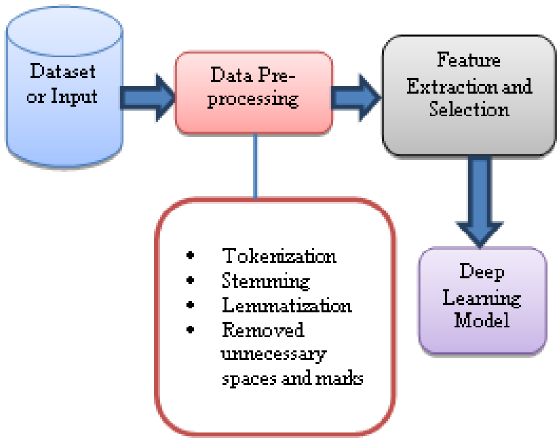

FIGURE 1. Steps for data pre-processing and feature extraction from audio transcript data.

effective performance has been achieved. Recently, deep learning methods have been developed based on CNN along with LSTM arrangement that attained significant advances.

B. NATURAL LANGUAGE PROCESSING (NLP) WITH DEEP LEARNING

This subsection introduces the process of NLP used by Deep Learning models. Disease prediction is a subset of NLP, which is a branch of Artificial Intelligence (AI) [14]. Fig. 1 depicts the NLP steps used for extracting tokens to be used by deep learning models.

Audio transcripts are fed to the neural network where identification of the linguistic pattern of AD+ and AD- patients takes place. Based on the observed data patterns, AD+ patients are classified from AD- [15].

III. EXISTING WORK

Accurate diagnosis of AD is still a challenging task in clinical practices. In recent years, various deep learning based models such as CNN, RNN, LSTM, CNN-RNN and CNN-LSTM were frequently used on transcripts of spoken languages of AD patients and control patients. Additionally, attention-based hybrid LSTM-CNN model has been employed in order to perform classifications in various healthcare domains. Liu *et al.* (2019) [16] used Attention-based Hybrid LSTM-CNN Model to detect various types of Arrhythmias. Cai *et al.* (2017) implemented hybrid CNN-LSTM model for Analyzing User Intent in Online Health Communities [17]. Current research has shown that high performance can be attained using longitudinal data that is utilized to make the classifiers [18]. Different subjects are required for prediction models based on longitudinal data. In the longitudinal study, missing data is a ubiquitous issue and this issue is conventionally circumvented by imputing lost data [19]. The effective means for informational measures can be learned from AD dementia data by examining the results of deep learning techniques and the prognosis of AD [12], [13].

Locke (1997) emphasized on lexical-semantic language components, part of which can be seen in the case of a younger utterance age. This study also underlined that lexical capability increased, automated syntactic processing, and hence lexical and syntactic language shifts [20]. Schwenk (2007), demonstrated the possibility of training an advanced neural network language model with a low error rate and ambiguity. The output was much better with only one hidden layer compared with the conventional language model. Deep learning has been strengthened because the error rate and ambiguity are lower [21]. Ball *et al.* (2009) showed positive results in syntactic interpretation in the context of an acquired language, such as Aphasia in adults, promoting further focus on efficient syntactic identification methods. We further explored deep models of neural networks to learn the language changes that differentiate the language of AD patients from healthy controls [22]. The effectiveness of using complex syntax for MCI classification features has been shown by Roark *et al.* (2011). In their study, speaking languages were used for the training of support vehicle machines (SVM) in 37 MCI patients and 37 NC with seven important pauses and syntactical language annotations. 86.1% of the area below the ROC curve was achieved by this approach [23]. Further, Sidorov *et al.* (2012) introduced the concept of syntactic sn-grams named it n-grams, which is different from traditional n-grams in terms of construction method [24]. In the proposed n-gram approach, the neighbours were using syntactic trees to track the linguistic correlations. Due to this, it was able to bring the syntactic knowledge into ML methods. Their proposed n-gram can be suitable for a wide range of tasks of NLP in which traditional n-grams were applied. They have described the way of applying authorship attention. For evaluation NB, J48, and SVM classifiers were traditionally applied in baseline n-grams of words, the POS letters and tags that showed the best results achieved using SVM. In comparison, Sidorov *et al.*(2014), showed that n-grams and skip-gram efficient class predictors for many tasks of language modelling sparse records [25]. Study proposed by Prud'hommeaux and Roark, 2015 was based on 'graph-based content word word-level score' to predict cognitive condition (AD) also followed by MCI, is Alzheimer's. The work was performed with the SVM in the same DementiaBank given an AUC of 82.3%. The techniques based on the graph however are important alignment models built separately and of adequate scale datasets [26].

Orimaye, *et al.* proposed models to differentiate between the likely AD group and the stable group through many linguistic biomarkers, syntactics, and n-grams. As such, the likely AD category of the stable elderly with a better ROC curve region (AUC = 0.82) using their diagnosis SVM model was considerably different from healthy elderly [27]. To distinguish between samples of vocabulary from AD and HC, Karlekar, *et al.* (2018), CNNs, LSTM, and CNN-RNNs were applied in the experiment. Their study achieved accuracies of 82.8%, 83.7%, 84.9%, and 91.1% for AD classification [28]. In this regard, Sheela, *et al.* (2020) [29] used DL and NLP approaches to evaluate language patterns of AD patients. Their proposed neural networks were trained

TABLE 1. DementiaBank dataset details.

	Subjects	Transcripts	Utterances/Sentences
Alzhimer's disease	208	1017	11458
Control normal	104	243	2904
Total	312	1260	14362

based on the transcripts of AD and CN. Sheela applied two models: CNN and CNN + bidirectional LSTM, and applied it to a dataset for comparison purposes. The experimental results showed that 72% accuracy was achieved using CNN + bidirectional LSTM [29].

IV. DementiaBank DATASET

The dataset is acquired from transcripts of spoken conversations of Alzheimer's patients (AD+) and control patients (AD-) from Pitt Corpus. Table 1 summarizes the summary of DementiaBank dataset. The University of Pittsburgh School of Medicine and the National Institute on Aging collaborated on a longitudinal research of AD and related dementias, which resulted in the collection of this data.[1]

DementiaBank dataset includes transcripts of patients with Alzheimer's dementia, other related dementia and healthy controls while having verbal interviews. These interviews were conducted based on a picture that contained different components depicting the "cookie-theft" scene at the Boston Diagnostic Aphasia Examination. These interviews were conducted in English. In the interview, "cookie-theft" picture was shown to the patients and told them to explore anything in the given picture they could see. The oral utterances of the patients were recorded and transcribed in the corresponding text in a transcription format. Other exercise includes the 'Recall Test', which requested patients to recall characteristics of the tale previously told or seen in the picture.

DementiaBank dataset contains 104 normal individuals and 208 dementia patients. It includes a total of 1,017 transcripts of Alzheimer's and 243 verified control transcripts. Each of the transcript in the data sample is broken into sentences or statements. Among the 14,362 samples of utterances, 11,458 are diagnosed with AD and 2,904 are the control patients [30].

The dataset used in this article is structured as a TSV file containing 3,245 records. The dataset is divided into training and testing sets in the ratio of 80:20. Further, 10% of the training data is used in each fold (K-fold) for validation purpose. The test and training set contain both 'AD positive' and 'AD negative' utterances. Each record and label is used as mentioned below: AD negative (AD^-) utterance as '0'control patients and AD positive (AD^+) utterance as '1'dementia patients.

Automated morpho-syntactic analysis is included with each transcript in DementiaBank, such as:

- Standard part-of-speech tagging
- The tense's description, and
- Markers for repeated words

[1] https://dementia.talkbank.org/access/English/Pitt.html

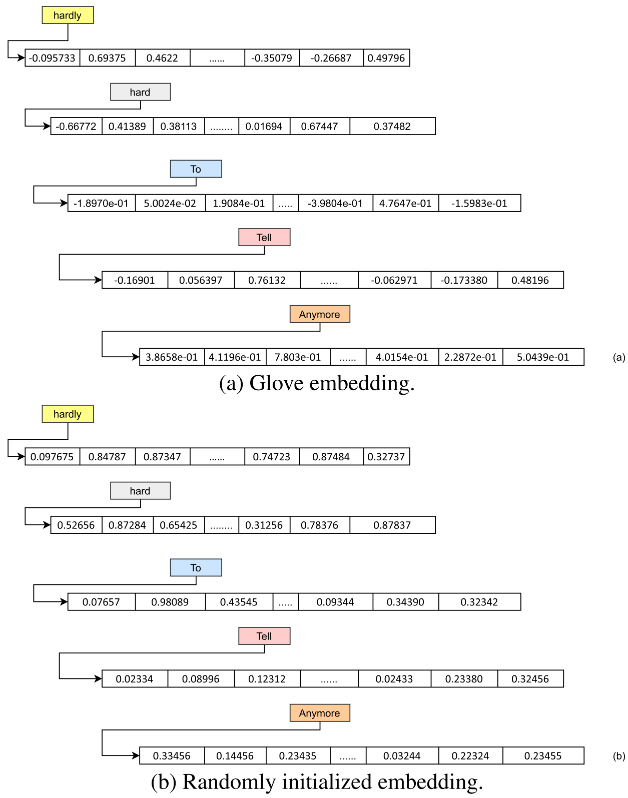

FIGURE 2. Text vector of size (100 dimensions) using: (a) Glove embedding (b) Randomly initialized embedding.

In other databases, such as CHILD Talk Bank, the same automated labelling is done, therefore it is not unique to DementiaBank.

V. METHODOLOGY

The framework for the proposed methodology for the early diagnosis of AD is addressed in this section, as shown in Fig. 2. The proposed methodology has different stages, as explained below.

A. PRE-PROCESSING OF AUDIO TRANSCRIPT DATA

Significant experience involves data that is fuel to the learning models. The dataset was obtained from DementiaBank. Prior to feed data into the learning models, pre-processing and feature extraction is done for obtaining better results. The simplest way to represent text data is to use a bag of words. It is a procedure in which a list is created to describe distinct terms in text data called vocabulary [31]. For each word in the phrase, 1 indicates that it is present in the vocabulary, whereas 0 indicates that it is not. Patients with possible AD are marked as positive, and the entries of the control patients are marked as negative. One of the major drawback of using "bag of words" is that it discards the order of words. Consequently, the sentence's context is ignored. The steps involved in the pre-processing of audio transcript data are discussed as follows:

- Data cleaning: Keeping only alphabets by removing all the symbols.

- Converting the whole sentence into the lower case for uniformity.
- Tokenization: Splitting a phrase, sentence, paragraph, or an entire text document into smaller units/words.
- Removing stop words: Designed custom stop words and removed them. Typical stop words from English are: "is", "an", "this", "the", etc.
- Performing Lemmatization: Lemmatization returns the dictionary form of a word, which must be a valid word.
- After performing Lemmatization, the words are rejoined to form sentences.

In the proposed work, lemmatization is chosen instead of stemming because stemming produce stems of words that might not be a twin of the morphological root of the word.

B. FEATURE EXTRACTION USING WORD EMBEDDING

The context of words is critical in NLP, thus we utilised a different method called word embeddings to capture the context of words [32], [33]. Word embedding is a representation of words where similar terms are denoted in the same way. Conceptually, it is a mathematical model from space with numerous dimensions per word to a continuous vector space with low dimensions.

One of the major challenges in NLP is the shortage of training data. Although NLP is a diversified area with a variety of different tasks, most task dataset only include a few thousand human-labeled exercises/examples for training [34]. However, advanced deep learning NLP-based models find benefits with a much larger volume of data such as millions or billions. To overcome this gap, researchers have developed a variety of model training techniques with a huge amount of un-annotated web text called pre-training. The pre-trained model can be refined on small NLP tasks, such as questionnaires and emotion analyses, leading to major accuracy improvements. In this study, two embedding techniques have been implemented as follows:

- Randomly initialized embedding
- Glove embedding

An example of Glove embedding and Randomly initialized embedding is elaborated using a sentence "Hardly hard to tell any more" as shown in Fig. 2a and 2b respectively.

C. METHODS USED

The methodology section describes the work of the techniques used, model design, implementation, and training. Three deep neural network models: CNN, CNN + Bidirectional LSTM, and the proposed SDDNN model are used for the classification task. These models are implemented using Glove embedding and Randomly initialized embedding techniques.

1) CONVOLUTIONAL NEURAL NETWORK (CNN)

The first model used is one dimensional (1D) CNN. The fundamental architecture of CNN enables network layers to learn numerous complex characteristics that a simple neural network cannot [35], [36]. Every input is passed through the embedding layer and then through 1D convolution layer followed by max-pooling layer. After carrying out convolution, the result is passed through two dense layers. The activation function used in the last dense layer is sigmoid, which outputs probabilities of both the classes. Dropout is used between convolution 1D layer and max pooling layer and then between last two dense layers [37]. Its step-wise description is given in Algorithm 1.

Algorithm 1 CNN Model

procedure *CNN*()
Input: Training data, the number of training epochs, number of iterations;
Output: Classification (AD^+ or AD^-)
  $Iter = 0$
  for every data sample x_i in $(x_1, x_2, \ldots, x_n \in R$, where R is a vector of size 2596*100) **do**
    $Iter = Iter + 1$
    if $(iter < Epoch)$
      E = Embedding(v_len, i_length = 100, weights = [embedding_matrix]) //Create word embedding of x_i
      $conv1 = Conv1D(filters = 256, kernel\_size = 5, activation =' relu')$
      $m\_pool = Max\_pool(pool\_size = 2)$
      $fnn1$ = Dense(units = neurons, activation = 'relu') //feed the extracted features from previous layer to FFN
      $fnn2$ = Feedforward neural network (fnn1) //Classify whether the sample is AD^+ or AD^-
    end if
  end for
end procedure

The detailed architecture of this model is shown in Fig.3 and its mathematical operations are summarized in Table 2.

Following are the types of embedded layers in deep CNN model:

- **Convolution layer**: New feature values are computed by applying a convolution operation among the input data and filters (convolution kernels). A filter can be perceived as a small window that includes the coefficient values. A number of convolved features are generated using different convolution kernels on the given data. These features are typically more useful than the original input parameters, thereby increasing the efficiency of the model [38].
- **Downsampling (MaxPooling) layer**: A pooling layer is a downsampling technique that takes some values out of the convolved values and creates a matrix of lower-dimensions. Thus the pooling layer computes a lower-dimension matrix, which is considered as simplified version of convolved features [39]. Thus the downsampling technique will make the system more

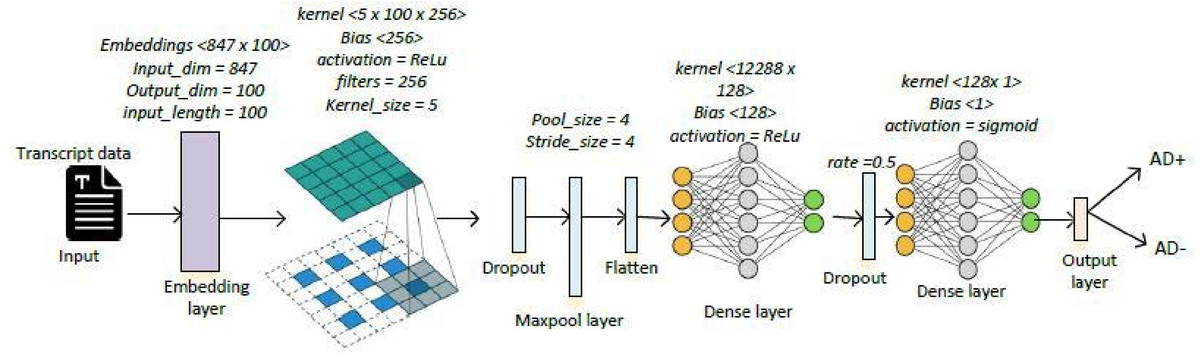

FIGURE 3. Architectural description of CNN model implemented in this study.

TABLE 2. Summary of mathematical operations in CNN model.

Operation	Mathematically given as	Variable Description
Convolution	$z^l = h^{(l-1)} * w^l$	z^l is pre-activation output of layer l $*$ is the discrete convolution operation
MaxPooling	$h^l_{xy} = max_{(i=0,..,s,j=0,..s)} h^{l-1}(x+i)(y+j)$	h^l is activation of layer l
Fully connected	$z^l = w^l h^{l-1}$	w signifies kernel
Activation function (Sigmoid)	$\sigma\,(z_i) = 1/(1+e^{-z_i})$	σ represents the Sigmoid function

stable as the pooled results are not altered by minor changes in the input.

- **Fully connected (Normal Flat feed-forward neural network) layer**: In the fully connected layer, every input is connected to every output by the weight. It serves the purpose of performing an actual classification task. Without this layer, a classical CNN cannot split out the predicted classes. The flattened feature map or feature vector is passed through a fully connected layer, which is similar to hidden layers in ANN [40], [41].

2) HYBRID OF CONVOLUTIONAL NEURAL NETWORK AND BIDIRECTIONAL LONG SHORT TERM MEMORY (CNN + BIDIRECTIONAL LSTM)

In this model, 1D convolution and 1D Bidirectional LSTM layers are used. Every input is passed through the embedding layer and then through the 1D convolution layer followed by max-pooling layer. With the use of feedback connections, RNNs have the ability to comprehend long-term connections [42]–[44]. LSTM comprises of three gates, and one cell state, each having a sigmoid activation function. Sigmoid has smooth curves from 0 to 1 and is differentiable. Each cell has a state vector and it can read, write, or reset it at any time step [45]. The main advantage of this network is that it remembers the past. In more detail, the output of LSTM is passed to the final dense layer. The detailed architecture of this model is shown in Fig. 4.

The activation used in the dense layer is sigmoid [46]–[48], which outputs the probabilities of both classes. A dropout of 0.2 is used between the LSTM layer and dense layer. The step-wise details of the model are discussed in Algorithm 2.

Algorithm 2 CNN + Bidirectional LSTM Model

1: **procedure** CNN_LSTM()
2: **Input**: Training data, the number of training epochs, number of iterations;
3: **Output**: Classification (AD^+ or AD^-)
4: $Iter = 0$
5: **for** every data sample x_i in ($x_1, x_2, \ldots, x_n \in R$, where R is a vector of size 2596*100) **do**
6: $Iter = Iter + 1$
7: **if** ($iter < Epoch$)
8: E = Embedding(v_len, i_length, weights = [embedding_matrix]) //Create word embedding of x_i
9: $conv1 = Conv1D(filters = 256, kernel\_size = 5, activation =' relu')$
10: $m\_pool = Max\_pool(pool\_size = 2)$
11: $lstm1$ = Bidirectional_LSTM(m_pool) //Feed the extracted features from 1D Convolution to Bidirectional LSTM
12: $fnn1$ = Feedforward neural network (lstm1) //Classify whether the sample is AD^+ or AD^-
13: **end if**
14: **end for**
15: **end procedure**

Equations (1)-(6) describe the operations performed by an LSTM that comprises of three gates and one cell state.

Forget Gate: f_t

$$f_t = \sigma(w_f s_{(t-1)} + w_f x_t) \tag{1}$$

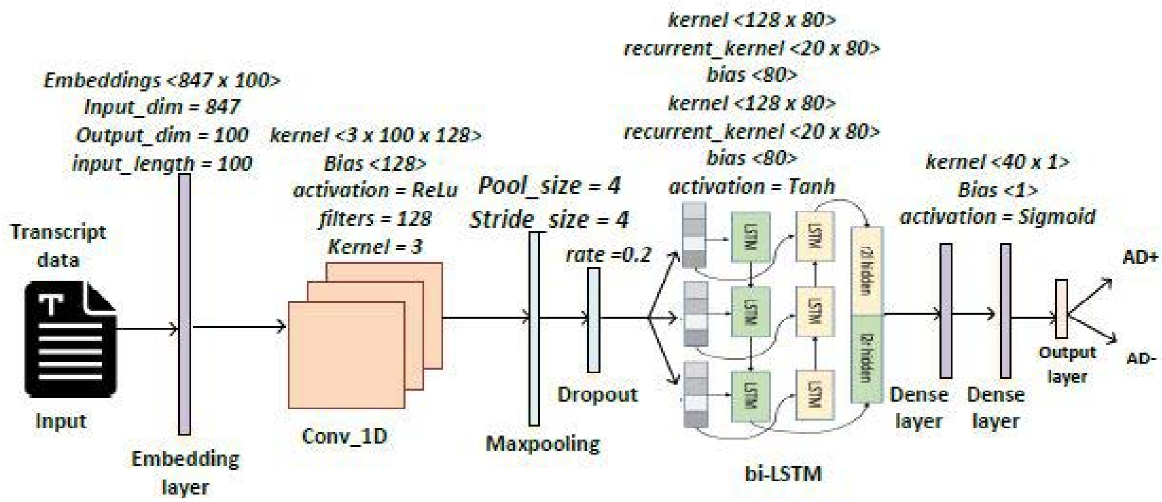

FIGURE 4. Architectural description of CNN + Bidirectional LSTM model implemented.

Input Gate: i_t

$$i_t = \sigma(w_i s_{(t-1)} + w_i x_t) \tag{2}$$

Output Gate: o_t

$$o_t = \sigma(w_o s_{(t-1)} + w_o x_t) \tag{3}$$

Each gate has different sets of weights. Here $c^{`}$ is called as intermediate cell state. $c_t^{`}$ is calculated using the given equation

Intermediate cell state $c_t^{`}$

$$c_t^{`} = tanh(w_c s_{(t-1)} + w_c x_t) \tag{4}$$

Cell state c_t is computed below as a result of adding the input gate and intermediate cell state with the old cell state and the forget gate as:

Cell state c_t

$$c_t = (i_t \times c_t^{`}) + (f_t \times c_{(t-1)}) \tag{5}$$

Then the cell state, the *tanh* activation multiplied with the output gate to get the new state

New State h_t constitutes the output of the memory cell and is calculated by

$$h_t = o_t \times tanh(c_t) \tag{6}$$

In the equations given above S_0 represents old state and X_1 denote the input. Here is also the previous cell state c_o. S_0 and X_1 are the inputs and w_c, w_o,w_i and w_f are separate weight vectors. First the previous state is calculated then the previous state and input is passed through sigmoid (σ) activation function. We have used the variants of the CNN + Bidirectional LSTM model in our implementation.

3) PROPOSED ARCHITECTURE OF STACKED DEEP DENSE NEURAL NETWORK (SDDNN) MODEL

Stacked model is a distributed predictive model that combines two or more hierarchical units to construct an individual weight and bias matrix. After computing these measures, they are processed through an artificial neural network, which generates a matrix with a uniform dimension. Finally, depending on the application, the dense output or the new configuration of weights and biases is passed to the concatenation layer. The combined output is then processed using a combination of neural functions and dense layers to arrive at an appropriate class prediction. In the proposed SDDNN model, a CNN is used as the first unit followed by Max pooling and a dense layer. A combination of CNN with Bidirectional LSTM (CNN + Bidirectional LSTM) memory networks is used as the second unit and a convolution network and Bidirectional LSTM with attention (Bidirectional LSTM with attention) is used as the third unit. Rather than using the additional function to create a vector segment, this study suggests using a concatenation layer. This not only reduces the computation but is also capable of extracting feature vectors in a more precise and efficient manner. Finally, three dense layers are used to make the prediction. The step-wise details of the proposed novel SDDNN (CNN, CNN + Bidirectional LSTM, Bidirectional LSTM with attention) model are discussed in Algorithm 3 and its architecture is depicted in Fig. 5.

DementiaBank dataset is trained using the proposed SDDNN model to identify Alzheimer's dementia using audio transcript data. The basic requirements for the units in this model are determined by the dataset's corpus tension. The data is widely dispersed and has a broad range in terms of standard deviation and raw manifestation. A simple recurrent technique is not efficient for such type of data.

To offer suitable gradients and functional modifications to the data set, initially CNN is used to induce the neighbour kernel association inside the embedding's. Following this, bidirectional LSTM network is then used to generate the universal association function. Finally, the incorporation of an attention layer to a collinear distribution function leads to optimization of gradients and relative class-wise

Algorithm 3 Proposed SDDNN Model

procedure *SDDNN*()
Input: Training data (d_{train}), test dataset (d_{test}), training epochs (e), input shape (inp), and batch size (b)
Output: Classification for AD^+ and AD^-
Embedding = Emb(v_len, i_len, $weights$ = [embedding matrix]) // create word embedding
model_CNN ← Conv1D(filters = 256, $kernel_{size}$, activation = 'relu')(Embedding)
Maxpool_CNN = Maxpool(pool_size = 2)(model_CNN)
Transform_CNN = Dense(2048)(Maxpool_CNN)
model_CNN ← Conv1D(filters = 256, $kernel_{size}$ = 5, activation = 'relu')(Embedding)
Maxpool_CNN = Maxpool(pool_size = 2)(model_CNN)
LSTM = Bidirectional_LSTM(64)(Maxpool_CNN)
Transform_CNN_BiLSTM = Dense(2048)(LSTM)
model_CNN ← Conv1D(filters = 256, $kernel_{size}$ = 5, activation = 'relu')(Embedding)
LSTM = Bidirectional_LSTM(64)(Maxpool_CNN)
Attn. = Attention(LSTM)
Transform_CNN_BiLSTM_Attention = Dense(2048)(Attn)
Converge = Concatenate () [Transform_CNN, Transform_CNN_BiLSTM, Transform_CNN_BiLSTM_Attention
Dense = Dense(1024)(Converge)
Dense = Dropout (0.28)(Dense)
Dense = Dense(256)(Dense)
Dense = Dense(2)(Dense)
i = 0, For every data sample x_i in X where R is the vector size of 2596⋆100
i = i + 1, i ← 1
while i ≤ e do
model_hybrid1_Fit(d_train,b,e) ← Train the Hybrid Model-1
end while
accuracy, precision, recall, fbeta_score, auc ← model_hybrid1_Evaluate(d_{test})
return accuracy, precision, recall, fbeta_score, auc
end procedure

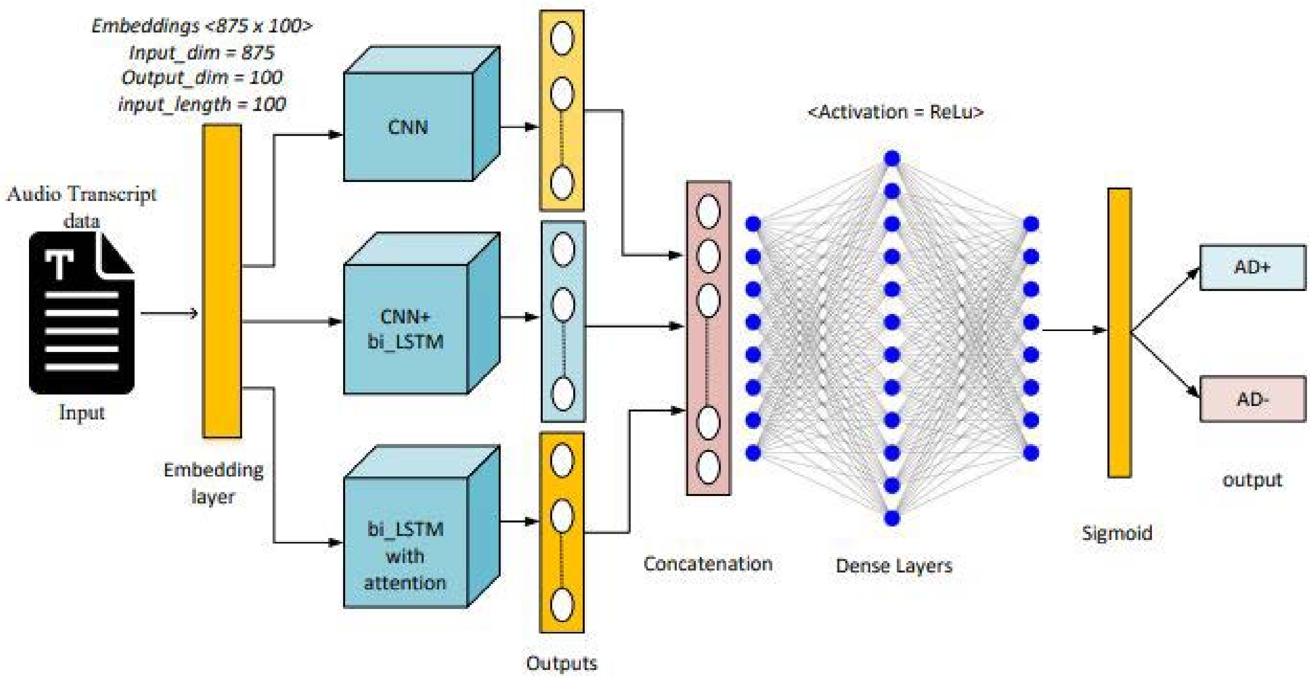

FIGURE 5. Proposed Stacked Deep Dense Neural Network (SDDNN) model: 1D CNN, CNN + Bidirectional LSTM, Bidirectional LSTM with attention, fully connected dense and output layers.

accuracy. Thus, activation optimization functions like Relu and softmax are embedded into individual layer structures. According to the current linguistic modelling principles, attention is of higher importance. In addition, it is important to emphasise on initially selected weights and biases that are entirely random and tend to choose constructual issues. Thus, it is critical to maintain a particular range for the weighted mean and standard deviation. Finally, after all of these modifications and procedures, the optimisation function Adam is selected, with a learning rate of 0.0001. Adam features have a dynamic learning rate and modification function that is linked with the model's mathematical origin.

VI. EXPERIMENTAL ANALYSIS

A. EXPERIMENTAL SETUP

We implemented three models: CNN, CNN + Bidirectional LSTM, and the proposed SDDNN model with exploration in terms of Randomly initiated and Glove embedding using Google Colaboratory on the top of TensorFlow gpu 2.2.0. The Google Colab is an open source services provided by Google [49] and provides a GUI to researchers with limited resources [50].

B. EVALUATION PARAMETERS

In order to evaluate the performance of these models, classes may be assessed in terms of several indices such as true positive (TP), true negative (TN), false positive (FP), and false negative (FN) [51]–[53]. TP depicts the accurately identified positive samples whereas TN depicts the accurately identified negative samples.

Samples FP indicates wrongly identified positive samples whereas FN shows wrongly classified negative samples. It is also known as type I error. The various performance metrics such as accuracy (Eq. 7), precision (Eq. 8), recall (Eq. 9), specificity (Eq. 10) and F_1 score, (Eq. 11) used for evaluating the effectiveness of these models are provided below.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{7}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{8}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{9}$$

$$Specificity = \frac{TN}{(TN + FP)} \tag{10}$$

$$F_1Score = \frac{(2 * (Recall * Precision))}{(Recall + Precision)} \tag{11}$$

Accuracy is one of the most often used measures to determine the performance of correctly recognized samples. The accuracy of a model is its ability to properly classify positive samples out of total positive samples [54]. Recall determines the exact positive samples (TP + FN) accessible. Specificity determines the actual negative samples. The F_1 score is calculated from the recall and precision. We also obtained the results of these three models with two embedding techniques.

C. HYPERPARAMETER TUNING TO OBTAIN OPTIMIZED PARAMETERS USING GridSearch

The second objective is to optimize the hyperparameters of the model. Hyperparameter tuning can reduce the error in the test dataset. The method of estimation for each model is distinct, and therefore each model has its own hyperparameters that need to be optimized. There are numerous parameters to be configured as discussed in [55], [56]. Hyperparameter tuning of all the models is summarized in Table 3.

- **Learning Rate:** This controls the amount of weight required to be updated in the optimization algorithm. Constant learning rate, momentum-based techniques, gradually decreasing learning rate, and adaptive learning rates can be used depending on the choice of optimizer [57].
- **Number of epochs:** The total number of times the training data is sent through the neural network model. The number of iterations is raised until the test-training error difference decreases.
- **Batch size:** The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters. A small group is preferable in the learning process for simplicity. A range of 15-130 can be a sensible choice [58].
- **Number of hidden layers:** It is sometimes practical to make variations in the number of hidden layers to decrease the error rate between testing and validation accuracy. The errors do not improve. Due to this, the computational cost to train a DNN increases. Having a variety of layers might lead to under-fitting, whereas having many layers may lead to over-fitting but is not harmful with regularization technique.
- **Dropout:** Dropout is a desirable regularization technique for overfitting in neural networks. The technique merely drops units in deep neural models to maintain the specified likelihood. A default value of 0.5, could be a sensible choice for training [59], [60].

We implemented GridSearch, an optimization tool used for tuning of hyperparameters. The concept is that various hyperparameters represent a certain combination of values to minimize the error of our predictive model. Our purpose is to locate this particular parameter combination using GridSearch. This approach tests each possible parameter and chooses the optimal parameters for each model.

VII. RESULTS AND DISCUSSIONS

All three models yielded approximately the same results for all the parameters considered in the analysis. The models achieved different classification accuracy levels for the same testing data for Randomly initialized and Glove embedding.

A. RESULTS WITH HYPERPARAMETER TUNING

Performance scores for all three models (CNN, CNN + bidirectional LSTM, and SDDNN) evaluated with optimized parameters obtained using GridSearch are given in Table 4.

TABLE 3. Hyperparameter tuning to obtain optimized parameters using GridSearch.

Models	Embedding Technique	Parameters Grid	Best Parameters
CNN	Randomly initialized embedding	optimizer = [RMSprop, Adam, SGD] dropout rate = [0.1, 0.2, 0.3, 0.4, 0.5] weight constraint = [1,2,3,4,5]	optimizer = ’Adam’ dropout rate = ’0.5’ weight constraint = 4
	Glove embedding	optimizer = [RMSprop, Adam, SGD] neurons = [128, 256] nb_filters = [128, 256] kernel size = [3, 5] dropout rate = [0.1, 0.2, 0.3, 0.4, 0.5] weight constraint = [1,2,3,4,5] batch size = [32, 64, 128, 256]	optimizer = ’Adam’ neurons = ’256’ nb_filters = 256 kernel size = 3 dropout rate = 0.5 weight constraint = 1 batch size = 32
CNN + bidirectional LSTM	Randomly initialized embedding	units = [10,20,30,40] optimizer = [RMSprop, Adam, SGD] nb_filters = [128, 256] kernel size = [3 , 5] dropout rate = [0.1,0.2,0.3,0.4,0.5] batch size = [32,64,128,256]	units = ‘30’ optimizer = ‘Adam’ nb_filters = ‘256’ kernel size = ‘3’ dropout rate = ‘0.1’ batch size = ‘64’
	Glove embedding	units = [10,20,64,128] optimizer = [RMSprop, Adam, SGD] nb_filters = [128, 256] kernel size = [3 , 5] dropout rate = [0.1,0.2,0.3,0.4,0.5] batch size = [32,64,128,256]	units = ‘20’ optimizer = ‘Adam’ nb_filters = ‘256’ kernel size = ‘5’ dropout rate = ‘0.5’ batch size = ‘128’
Proposed SDDNN Model	Randomly initialized embedding	units = [10,20,64,128] optimizer = [RMSprop, Adam, SGD] nb_filters = [128, 256] kernel size = [3, 5] pool size = [2,4] neurons = [128, 256,512] dropout rate = [0.1, 0.2, 0.3, 0.4, 0.5] batch size = [32, 64, 128, 256] weight constraint =[1,2,3,4,5]	units = ’64’ optimizer = ‘Adam’ nb_filters = ’265’ kernel size = 5 pool size = 4 neurons = ’128’ dropout rate = ’0.5’ batch size = ’32’ weight constraint = ‘3’
	Glove embedding	units = [10,20,32,64,128] optimizer = [RMSprop, Adam, SGD] nb_filters = [128, 256] kernel size = [3, 5] pool size = [2,4] neurons = [10,20,64,128] dropout rate = [0.1, 0.2, 0.3, 0.4, 0.5] batch size = [32, 64, 128, 256] weight constraint = [1,2,3,4,5]	units = ’32” optimizer = ‘Adam’ nb_filters = ’256’ kernel size = ’5’ pool size = ’4’ neurons = ’128’ dropout rate = 0.5 batch size = ’32’ weight constraint = ‘3’

TABLE 4. Performance scores (in %) of three models for binary classification on DementiaBank dataset with optimized parameters using GridSearch (hyperparameter tuning).

Model	Embedding Techniques	Accuracy	Specificity	precision	Recall	F1 score	AUC
CNN	Randomly initialized embedding	84.44	81.87	81.21	87.19	84.14	91.05
	Glove embedding	85.05	85.67	84.09	84.36	84.22	91.36
CNN + bidirectional LSTM	Randomly initialized embedding	84.43	86.25	84.93	86.31	85.62	91.28
	Glove embedding	84.89	86.54	84.71	83.06	83.88	91.22
Proposed SDDNN model	Randomly initialized embedding	92.01	90.27	88.23	83.66	84.96	91.76
	Glove embedding	**93.31**	91.23	88.22	87.23	85.69	92.22

TABLE 5. Performance evaluation of models with optimized parameter using 10-Fold CV.

Model	Variations/techniques	1-fold	2-fold	3-fold	4-fold	5-fold	6-fold	7-fold	8-fold	9-fold	10-fold	Mean
CNN	Randomly initialized embedding	99.0915	98.7215	99.3096	99.2419	99.0486	99.5293	99.1801	99.8478	99.1094	99.1645	99.22
	Glove embedding	99.2572	98.8340	99.2740	99.2716	99.1026	99.4874	99.2464	99.7345	99.2207	99.2123	99.26
CNN + bidirectional LSTM	Randomly initialized embedding	99.2453	99.1003	99.5822	99.4143	99.1746	99.5713	99.2163	99.8598	99.0785	99.1705	99.34
	Glove embedding	98.6535	99.0175	98.7348	99.1765	99.0786	99.5533	99.2705	99.8240	98.6641	98.8602	99.08
Proposed SDDNN Model	Randomly initialized embedding	99.2365	97.5698	98.3256	99.3652	98.1458	99.3698	98.1478	99.1475	99.2563	99.7619	99.23
	Glove embedding	99.2585	99.1236	99.6547	99.6589	99.4758	99.9764	99.8561	99.6532	99.1212	99.6678	99.78

GridSearch carries out an exhaustive search, resulting in several possible best parameters being selected from a grid of parameter values. The configuration of hyperparameters, especially in deep learning, influences the performance of the optimizer. The performances of these models improved significantly after performing hyoerparameter tuning.

As shown in Table 4, our proposed SDDNN model with Glove embedding has outperformed other models with a classification accuracy of 93.31%. Stacked approach with optimized parameters has proven to make most accurate predictions. However, CNN achieved an accuracy of 84.44% and 85.05% with Randomly initialized embedding and Glove

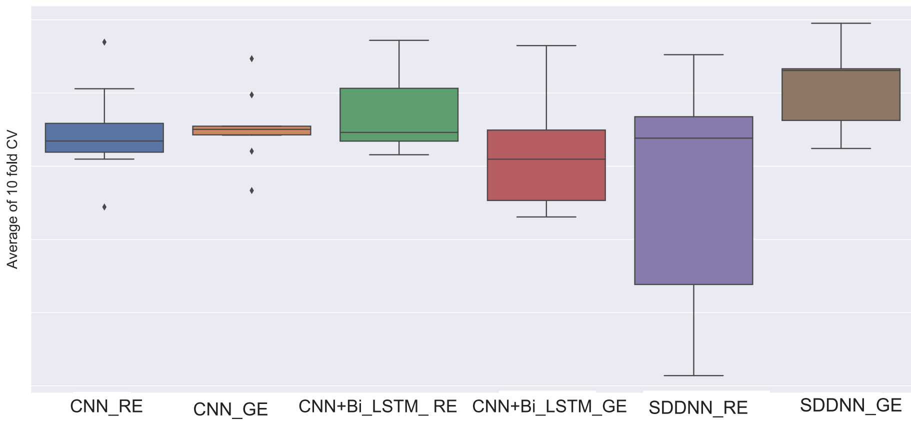

FIGURE 6. 10-fold cross validation performance of three models with two embedding techniques after hyperparameter tuning.

embedding respectively. Whereas, Hybrid of CNN + Bidirectional LSTM achieved 84.43% and 84.89% of accuracy with Randomly initialized embedding and Glove embedding respectively. the proposed model also showed better results for specificity, precision, recall, and AUC compared to CNN and CNN + Bidirectional LSTM models.

B. CROSS VALIDATION: EVALUATING ESTIMATOR EFFICIENCY

During the evaluation of hyperparameters for the estimator, there is a risk of overfitting in the test dataset. This is because the values can be modified until the estimator performs ideally. By doing this, there is a possibility of data leakage. A validation test is required to determine the final validity of the model's predictions. Validation is the process of determining whether the numerical findings derived from hypothesis testing are acceptable as per the description of data. K-folds cross-validation (CV) ensures that each original dataset observation has the possibility of showing the performance of any model. Shuffling of dataset before running K-fold CV technique ensures that the order of the input and output data is completely randomized. To ensure that the input data is unbiased, shuffling is performed. Finally, the dataset is divided into k equal-sized partitions.

In this analysis, a 10-fold CV technique is implemented for all three models for both Randomly initialized and Glove embedding. Therefore, the first step is to break the dataset into 10 equal groups. After determining the empirical square loss, first fold is used to test the models and compute the loss.

For the k^{th} part, we fit the model to the other $K-1$ parts of the data and determine the prediction error of the fitted model while predicting the k^{th} part of the data. This is repeated for $k = 1, 2, \ldots, K$, and the K estimates of the prediction error are combined. Let $k : 1, 2, \ldots, N$ be an indexing function indicating the partition to which observation i is randomly assigned. The fitted function computed with the k^{th} part of the deleted data, is denoted by $\hat{f}^{-k}(x)$. The cross-validation estimation of the prediction error was defined in Eq. 12:

$$CV(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i)) \tag{12}$$

where $N = 10$ and $k(i) = i$, for i^{th} observation. The fit is determined for the i^{th} observation using all of the data except the i^{th}. These settings have been demonstrated empirically to produce test error rate estimates that are neither excessively high in bias nor extremely high in variance when $k = 10$ CV is used. CV is done for three models with two embedding techniques (Randomly Initialized Embedding and Glove Embedding). The average of the 10 10-fold CV is then provided as the performance metric for all the three models. 10 rounds of CV are calculated using different divisions and the validation results are averaged across the rounds to determine the model's prediction performance as shown in Table 5. The proposed SDDNN model using Randomly initialized and Glove embedding has shown the highest mean value of 99.23 and 99.78 respectively after hyperparameter tuning. The mean of 10 folds for each model represents the fitness metrics used in prediction in order to obtain a more precise approximation of model prediction performance. The Boxplot [61] of the 10-fold CV for CNN, CNN + Bidirectional LSTM, SDDNN model with the two embedding techniques is shown in Fig. 6 which helps in drawing the following inference:

(a) CNN + Bidirectional LSTM with Glove embedding (CNN+Bi_LSTM+GE) shows poor behaviour.

(b) The SDDNN model in combination with Randomly initialized and Glove embedding (SDDNN+RE, SDDNN+GE) shows better performance.

C. CONFUSION MATRICES

To evaluate and compare the performance of the three models, confusion matrices [62], [63] are computed for both Randomly initialized and Glove embedding types.

For all the three models, a total of 649 samples are used to test the confusion matrix. The confusion matrices for

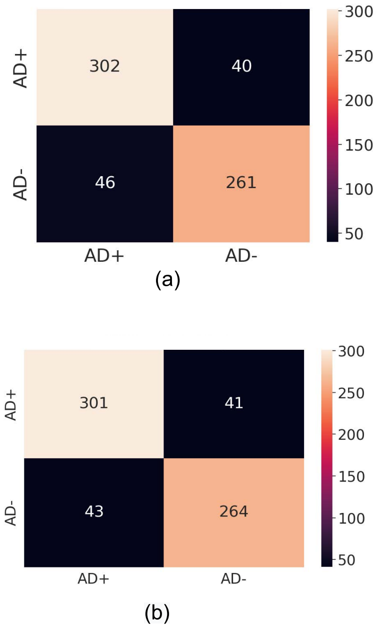

FIGURE 7. **Confusion matrices [64] for 2-Class classification task using (a) Randomly initialized embedding and (b) Glove embedding for CNN model.**

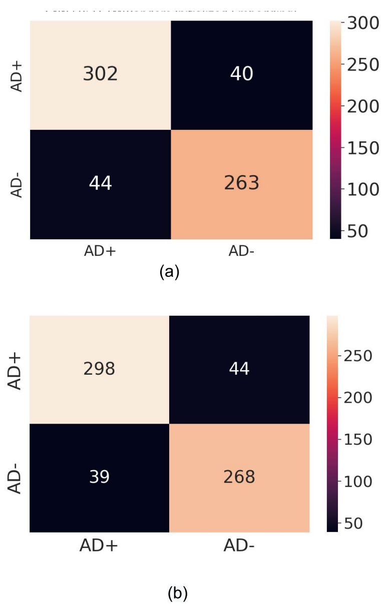

FIGURE 8. **Confusion matrices for 2-Class classification task using (a) Randomly initialized embedding and (b) Glove embedding for CNN + Bidirectional LSTM model.**

the binary classification problem of detecting the linguistic cues for AD+ and AD- are provided in Figs. 7, 8 and 9 respectively. For CNN model: The confusion matrix specifies a table that is used to describe the classification performance on a test dataset for known true values. For randomly initialized embedding, 565 samples represent true (predicted) values, whereas only 84 samples represent false (error) values. Confusion Matrix for CNN + Bidirectional LSTM hybrid model: For randomly initialized embedding, 565 samples represent true (predicted) values,

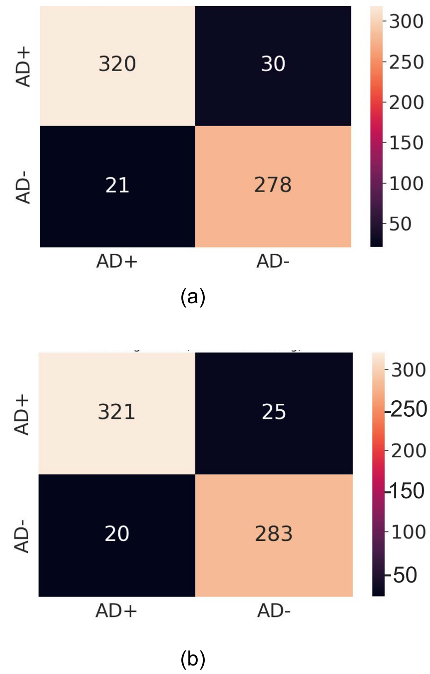

FIGURE 9. **Confusion matrices for 2-Class classification task using (a) Randomly initialized embedding and (b) Glove embedding for the proposed SDDNN model.**

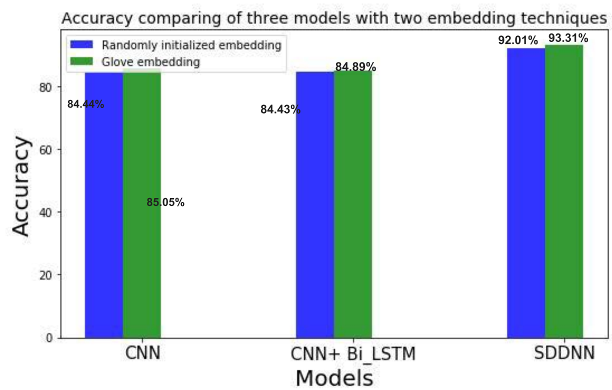

FIGURE 10. **Accuracy comparison for CNN, CNN + Bidirectional LSTM and SDDNN with optimized parameters using two embedding technique.**

whereas only 94 samples represent false (error) values. For glove embedding, 568 samples represent true (predicted) values, and 84 samples represent false (error) values. Whereas only 51 samples represent false (error) values. For glove embedding, 604 samples represent true (predicted) values and 45 samples represent false (error) values. Confusion Matrix for SDDNN: For randomly initialized embedding, 598 samples represent true (predicted) values.

D. COMPARATIVE ANALYSIS

In this subsection, performance-based comparison of the proposed model with CNN and CNN + Bidirectional LSTM is carried out using a variety of performance metrics. The graphs in Fig. 10 represents the comparison among CNN, CNN + Bidirectional LSTM, and the proposed SDDNN model based on testing accuracy levels.

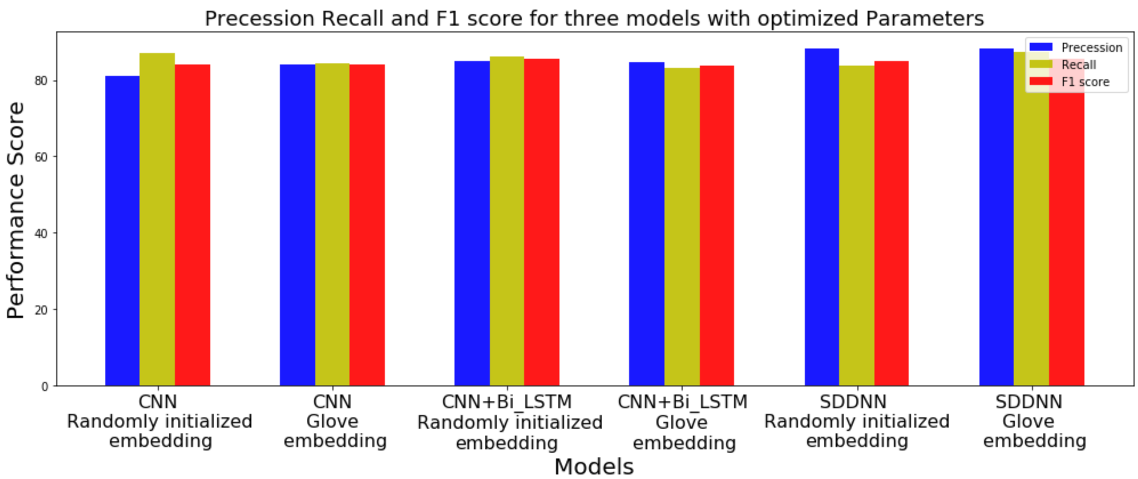

FIGURE 11. Comparison based on Precision, Recall and F_1 score for CNN, CNN + Bidirectional LSTM and SDDNN using two embedding technique.

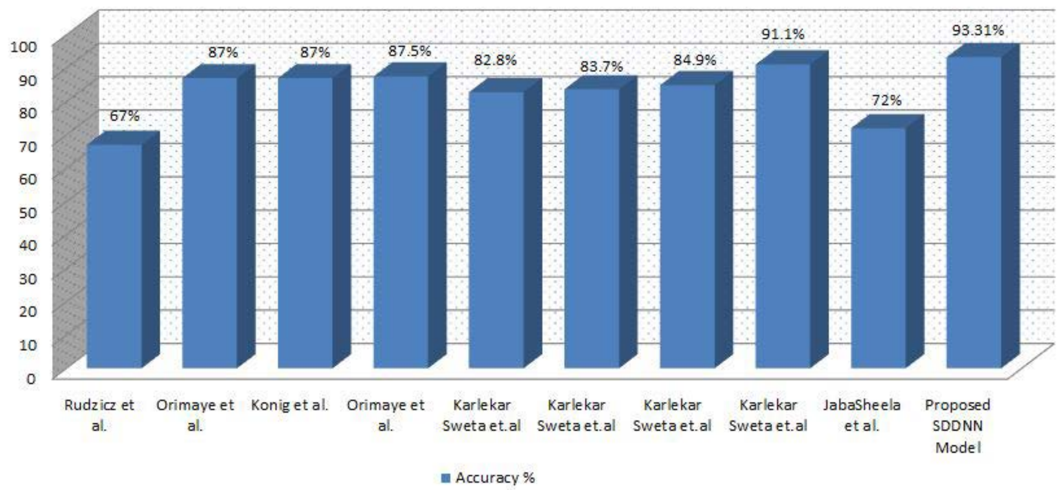

FIGURE 12. Accuracy comparison of proposed model with existing methods.

TABLE 6. Comparison of the proposed model with existing works.

Author	Model	Description	Accuracy
Rudzicz et al. [55]	Machine Learning	Extracted over 200+ lexical features	67.00%
Orimaye et al. [56]	Machine Learning	Used syntactic, lexical, and n-gram features	87.00%
Konig et al. [57]	Machine Learning	Analyzed speech audio	87.00%
Orimaye et al. [58]	Deep Learning	Deep Neural + Language Model	87.50%
Karlekar Sweta et al. [24]	Hybrid Deep Learning	2D-CNN Non-Tagged Utterances	82.80%
		LSTM Non-Tagged Utterances	83.70%
		CNN-RNN Non-Tagged Utterances	84.90%
		CNN-RNN POS-Tagged Utterances	91.10%
JabaSheela et al. [25]	Hybrid Deep Learning	CNN + bidirectional LSTM	72.00%
Proposed SDDNN Model	Stacked Deep Dense Neural Network + NLP	Analyzed Audio Transcripts	**93.31%**

The graph shows that parameter tuning significantly improves the performance of all three models. However, among all the three models, Glove embedded Stacked SDDNN model with optimized parameters achieves maximum accuracy (93.31%). The results shown in Fig. 11 provide a comparison based on the precision, recall, and F_1 score. The graph shows that maximum precision of 88.23% is obtained by Randomly initialize embedded proposed SDDNN model with optimized parameters. The maximum recall value of 87.23% is obtained by Glove embedded proposed SDDNN model, and maximum F_1 score of 85.69% is obtained by Glove embedded SDDNN model. Table 6 compares the benchmark accuracies achieved by the proposed model with the accuracy achieved by previous

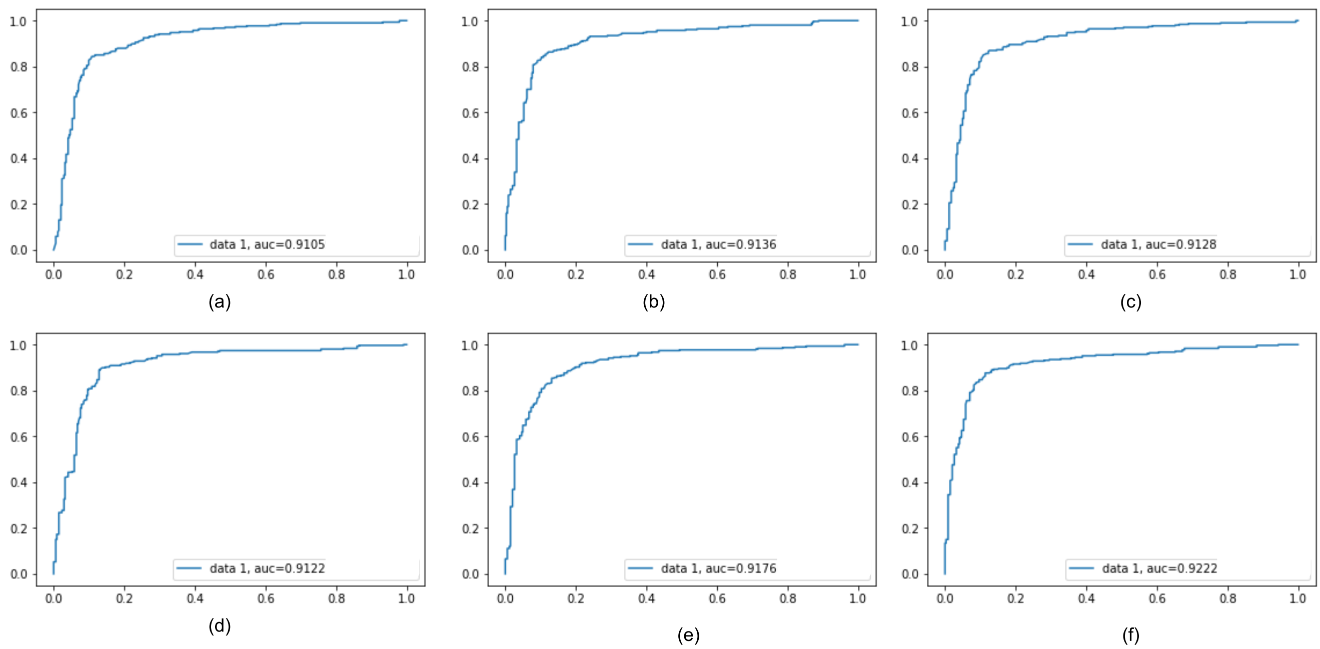

FIGURE 13. Plots of AUC for: (a) CNN (Randomly initiated embedding) (b) CNN (Glove embedding) (c) CNN + Bidirectional LSTM (Randomly initiated embedding) (d) CNN + Bidirectional LSTM (Glove embedding) (e) SDDNN (Randomly initiated embedding) (f) SDDNN (Glove embedding).

studies. The graph shown in Fig. 12 represents the comparative performance of the proposed model with those of previous studies.

E. AREA UNDER THE ROC CURVE (AUC) PLOTS

The AUC plots of the three models, CNN, CNN + Bidirectional LSTM, and the proposed SDDNN model are shown in Fig. 13 for both Randomly initiated and Glove embedding. The output curve shows that the maximum AUC is achieved by Glove embedded SDDNN model with a value of 0.9222 that outperformed CNN and CNN + Bidirectional LSTM with two embedding techniques.

VIII. CONCLUSION AND FUTURE SCOPE

For neurodegenerative conditions, such as Alzheimer's disease (AD) and associated dementias, the manual identification of these disorders has proven difficult. Currently, specific clinical diagnostic criteria and cognitive examinations are used to diagnose these illnesses. By using various deep learning algorithms, together with lower-level linguistic data produced from spoken utterances, it is possible to construct automated diagnostic models for the detection of people with suspected Alzheimer's disease from a larger population. DementiaBank language transcript clinical dataset allowed us to create multiple hybrid deep learning models to improve results.

In this article, three deep neural network models namely CNN, CNN + bidirectional LSTM, and SDDNN are used for 2-class (AD+ and AD-) classification purposes using two approaches: Randomly initialized and Glove embedding. Attention mechanism has been applied to a computational model for improving classification in the prediction of Alzheimer's dementia using audio transcript data.

For clinical AD datasets and the NLP viewpoint, we discussed and described the applicability of deep neural networks with two embedding techniques to classify AD patients. Of all the approaches, SDDNN using Glove embedding performed better than other models. Different performance parameters like accuracy, precision, specificity, recall, AUC, and F_1 score were used for testing and comparing these models. Glove embedding SDDNN outperformed state-of-the-art methods by achieving an accuracy, precision, specificity, recall, F_1 score and AUC of 93.31%, 88.22, 91.23, 87.23, 85.69 and 92.22 respectively.

In future, work can explore some more stages of Alzheimer's disease and apply more robust classification models for multi-class classification. These models can be used in several healthcare domains. However, it is also necessary to improve the diagnostic model's performance on larger datasets, thereby, enhancing the Accuracy.

A. DATA AVAILABILITY

The python code of this article will be shared on request to the corresponding author. The data underlying this article is available at URL: https://dementia.talkbank.org/access/English/Pitt.html

B. FUNDING STATEMENT

The authors received no specific funding for this study.

C. CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

[1] G. McKhann, D. Drachman, M. Folstein, R. Katzman, D. Price, and E. M. Stadlan, ''Clinical diagnosis of Alzheimer's disease: Report of the NINCDS–ADRDA work group* under the auspices of department of health and human services task force on Alzheimer's disease,'' *Neurology*, vol. 34, no. 7, p. 939, 1984.

[2] J. Xiao, N. Katsumata, F. Bernier, K. Ohno, Y. Yamauchi, T. Odamaki, K. Yoshikawa, K. Ito, and T. Kaneko, ''Probiotic bifidobacterium breve in improving cognitive functions of older adults with suspected mild cognitive impairment: A randomized, double-blind, placebo-controlled trial,'' *J. Alzheimer's Disease*, vol. 77, no. 1, pp. 139–147, Sep. 2020.

[3] R. Brookmeyer, E. Johnson, K. Ziegler-Graham, and H. M. Arrighi, ''Forecasting the global burden of Alzheimer's disease,'' *Alzheimer's Dementia*, vol. 3, no. 3, pp. 186–191, Jul. 2007.

[4] A.-D. Pham, A. Névéol, T. Lavergne, D. Yasunaga, O. Clément, G. Meyer, R. Morello, and A. Burgun, ''Natural language processing of radiology reports for the detection of thromboembolic diseases and clinically relevant incidental findings,'' *BMC Bioinf.*, vol. 15, no. 1, pp. 1–10, Dec. 2014.

[5] R. Beardmore, R. Hou, A. Darekar, C. Holmes, and D. Boche, ''The locus coeruleus in aging and Alzheimer's disease: A postmortem and brain imaging review,'' *J. Alzheimer's Disease*, vol. 83, no. 1, pp. 5–22, Aug. 2021.

[6] L. G. Apostolova, S. L. Risacher, T. Duran, E. C. Stage, N. Goukasian, J. D. West, T. M. Do, J. Grotts, H. Wilhalme, K. Nho, and M. Phillips, ''Associations of the top 20 Alzheimer disease risk variants with brain amyloidosis,'' *JAMA Neurol.*, vol. 75, no. 3, pp. 328–341, 2018.

[7] M. Amiribesheli and H. Bouchachia, ''A tailored smart home for dementia care,'' *J. Ambient Intell. Humanized Comput.*, vol. 9, no. 6, pp. 1755–1782, Nov. 2018.

[8] C. Reitz, C. Brayne, and R. Mayeux, ''Epidemiology of Alzheimer disease,'' *Nature Rev. Neurol.*, vol. 7, no. 3, pp. 137–152, 2011.

[9] I. A. Illán, J. M. Górriz, M. M. López, J. Ramírez, D. Salas-Gonzalez, F. Segovia, R. Chaves, and C. G. Puntonet, ''Computer aided diagnosis of Alzheimer's disease using component based SVM,'' *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2376–2382, Mar. 2011.

[10] Y. F. Khan and B. Kaushik, ''Neuro-image classification for the prediction of alzheimer's disease using machine learning techniques,'' in *Proc. Int. Conf. Mach. Intell. Data Sci. Appl.*, Cham, Switzerland: Springer, 2021, pp. 483–493.

[11] Y. F. Khan and B. Kaushik, ''Computer vision technique for neuro-image analysis in neurodegenerative diseases: A survey,'' in *Proc. Int. Conf. Emerg. Smart Comput. Informat. (ESCI)*, Mar. 2020, pp. 346–350.

[12] S. G. Costafreda, I. D. Dinov, Z. Tu, Y. Shi, C.-Y. Liu, I. Kloszewska, P. Mecocci, H. Soininen, M. Tsolaki, B. Vellas, L.-O. Wahlund, C. Spenger, A. W. Toga, S. Lovestone, and A. Simmons, ''Automated hippocampal shape analysis predicts the onset of dementia in mild cognitive impairment,'' *NeuroImage*, vol. 56, no. 1, pp. 212–219, May 2011.

[13] K. Li, R. O'Brien, M. Lutz, S. Luo, and T. Alzheimer's Disease Neuroimaging Initiative, ''A prognostic model of Alzheimer's disease relying on multiple longitudinal measures and time-to-event data,'' *Alzheimer's Dementia*, vol. 14, no. 5, pp. 644–651, May 2018.

[14] M. J. Ball, M. R. Perkins, N. Müller, and S. Howard, *The Handbook of Clinical Linguistics*. Hoboken, NJ, USA: Wiley, 2009.

[15] L. JabaSheela, S. Vasudevan, and V. Yazhini, ''A hybrid model for detecting linguistic cues in alzheimer's disease patients,'' *J. Inf. Comput. Sci.*, vol. 10, no. 1, pp. 85–90, 2020.

[16] F. Liu, X. Zhou, T. Wang, J. Cao, Z. Wang, H. Wang, and Y. Zhang, ''An attention-based hybrid LSTM-CNN model for arrhythmias classification,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[17] R. Cai, B. Zhu, L. Ji, T. Hao, J. Yan, and W. Liu, ''An CNN-LSTM attention approach to understanding user query intent from online health communities,'' in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2017, pp. 430–437.

[18] J. Escudero, E. Ifeachor, J. P. Zajicek, C. Green, J. Shearer, and S. Pearson, ''Machine learning-based method for personalized and cost-effective detection of Alzheimer's disease,'' *IEEE Trans. Biomed. Eng.*, vol. 60, no. 1, pp. 164–168, Jan. 2013.

[19] P. Padilla, M. López, J. M. Górriz, J. Ramírez, D. Salas-Gonzalez, and I. Álvarez, ''NMF-SVM based CAD tool applied to functional brain images for the diagnosis of Alzheimer's disease,'' *IEEE Trans. Med. Imag.*, vol. 31, no. 2, pp. 207–216, Feb. 2012.

[20] J. L. Locke, ''A theory of neurolinguistic development,'' *Brain Lang.*, vol. 58, no. 2, pp. 265–326, Jun. 1997.

[21] H. Schwenk, ''Continuous space language models,'' *Comput. Speech Lang.*, vol. 21, no. 3, pp. 492–518, Jul. 2007.

[22] J. K. Chambers and N. Schilling, *The Handbook of Language Variation and Change*. Hoboken, NJ, USA: Wiley, 2018.

[23] B. Roark, M. Mitchell, J.-P. Hosom, K. Hollingshead, and J. Kaye, ''Spoken language derived measures for detecting mild cognitive impairment,'' *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 7, pp. 2081–2090, Sep. 2011.

[24] J. Ramirez, J. M. Gorriz, F. Segovia, R. Chaves, D. Salas-Gonzalez, M. Lopez, I. Alvarez, and P. Padilla, ''Early Alzheimer's disease diagnosis using partial least squares and random forests,'' in *Proc. IEEE Int. Symp. Biomed. Imag., From Nano Macro*, Apr. 2010, pp. 81–84.

[25] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, ''Syntactic N-grams as machine learning features for natural language processing,'' *Expert Syst. Appl.*, vol. 41, no. 3, pp. 853–860, Feb. 2014.

[26] E. Prud'hommeaux and B. Roark, ''Graph-based word alignment for clinical language evaluation,'' *Comput. Linguistics*, vol. 41, no. 4, pp. 549–578, Dec. 2015.

[27] S. O. Orimaye, J. S. Wong, K. J. Golden, C. P. Wong, and I. N. Soyiri, ''Predicting probable Alzheimer's disease using linguistic deficits and biomarkers,'' *BMC Bioinf.*, vol. 18, p. 34, Jan. 2017.

[28] S. Karlekar, T. Niu, and M. Bansal, ''Detecting linguistic characteristics of Alzheimer's dementia by interpreting neural models,'' 2018, *arXiv:1804.06440*.

[29] L. JabaSheela, S. Vasudevan, and V. Yazhini, ''A hybrid model for detecting linguistic cues in Alzheimer's disease patients,'' *J. Inf. Comput. Sci.*, vol. 10, no. 1, pp. 85–90, 2020.

[30] Z. Noorian, C. Pou-Prom, and F. Rudzicz, ''On the importance of normative data in speech-based assessment,'' 2017, *arXiv:1712.00069*.

[31] J. Wang, P. Liu, M. F. H. She, S. Nahavandi, and A. Kouzani, ''Bag-of-words representation for biomedical time series classification,'' *Biomed. Signal Process. Control*, vol. 8, no. 6, pp. 634–644, 2013.

[32] P. Wang, B. Xu, C. L. Liu, H. Zhang, F. Wang, and H. Hao, ''Word embedding clustering and convolutional neural network for short text categorization,'' 2015.

[33] R. Sharma, B. Kaushik, N. K. Gondhi, M. Tahir, and M. K. I. Rahmani, ''Quantum particle swarm optimization based convolutional neural network for handwritten script recognition,'' *Comput., Mater. Continua*, vol. 71, no. 3, pp. 5855–5873, 2022.

[34] P. Wang, B. Xu, J. Xu, G. Tian, C. Liu, and H. Hao, ''Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification,'' *Neurocomputing*, vol. 174, pp. 806–814, Jan. 2016.

[35] M. Mahrishi, S. Morwal, A. W. Muzaffar, S. Bhatia, P. Dadheech, and M. K. I. Rahmani, ''Video index point detection and extraction framework using custom YoloV4 darknet object detection model,'' *IEEE Access*, vol. 9, pp. 143378–143391, 2021.

[36] K. Bhalla, D. Koundal, S. Bhatia, M. Khalid Imam Rahmani, and M. Tahir, ''Fusion of infrared and visible images using fuzzy based Siamese convolutional network,'' *Comput., Mater. Continua*, vol. 70, no. 3, pp. 5503–5518, 2022.

[37] M. Adnan, A. Habib, J. Ashraf, S. Mussadiq, and A. Ali Raza, ''Deep neural network based m-learning model for predicting mobile learners' performance,'' *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 28, no. 3, pp. 1422–1441, May 2020.

[38] D. Beaini, S. Achiche, A. Duperré, and M. Raison, ''Deep green function convolution for improving saliency in convolutional neural networks,'' *Vis. Comput.*, vol. 37, pp. 227–244, 2021, doi: 10.1007/s00371-020-01795-8.

[39] D. Scherer, A. Müller, and S. Behnke, ''Evaluation of pooling operations in convolutional architectures for object recognition,'' in *Proc. Int. Conf. Artif. Neural Netw.*, Springer, 2010, pp. 92–101.

[40] M. G. Kim, H. Ko, and S. B. Pan, ''A study on user recognition using 2D ECG based on ensemble of deep convolutional neural networks,'' *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 5, pp. 1859–1867, May 2019.

[41] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' 2014, *arXiv:1409.1556*.

[42] S. O. Orimaye, J. S. Wong, K. J. Golden, C. P. Wong, and I. N. Soyiri, ''Predicting probable Alzheimer's disease using linguistic deficits and biomarkers,'' *BMC Bioinf.*, vol. 18, p. 34, Jan. 2017.

[43] T. Udelhoven and B. Schütt, ''Capability of feed-forward neural networks for a chemical evaluation of sediments with diffuse reflectance spectroscopy,'' *Chemometric Intell. Lab. Syst.*, vol. 51, no. 1, pp. 9–22, May 2000.

[44] Ö. F. Ertuğrul, ''A novel randomized recurrent artificial neural network approach: Recurrentrandom vector functional link network,'' *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 27, no. 6, pp. 4246–4255, Nov. 2019.

[45] M. T. S. Al-Kaltakchi, R. R. O. Al-Nima, and M. A. M. Abdullah, ''Comparisons of extreme learning machine and backpropagation-based i-vector approach for speaker identification,'' *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 28, no. 3, pp. 1236–1245, May 2020.

[46] A. Wanto, A. P. Windarto, D. Hartama, and I. Parlina, ''Use of binary sigmoid function and linear identity in artificial neural networks for forecasting population density,'' *Int. J. Inf. Syst. Technol.*, vol. 1, no. 1, pp. 43–54, 2017.

[47] K. Tran, A. Bisazza, and C. Monz, ''Recurrent memory networks for language modeling,'' 2016, *arXiv:1601.01272*.

[48] M. Sundermeyer, H. Ney, and R. Schlüter, ''From feedforward to recurrent LSTM neural networks for language modeling,'' *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 3, pp. 517–529, Mar. 2015.

[49] T. Carneiro, R. V. M. D. Nóbrega, T. Nepomuceno, G.-B. Bian, V. H. C. D. Albuquerque, and P. P. Reboucas Filho, ''Performance analysis of Google colaboratory as a tool for accelerating deep learning applications,'' *IEEE Access*, vol. 6, pp. 61677–61685, 2018.

[50] T. Carneiro, R. V. M. D. Nóbrega, T. Nepomuceno, G.-B. Bian, V. H. C. D. Albuquerque, and P. P. R. Filho, ''Performance analysis of Google colaboratory as a tool for accelerating deep learning applications,'' *IEEE Access*, vol. 6, pp. 61677–61685, 2018.

[51] O. O. Koyejo, N. Natarajan, P. K. Ravikumar, and I. S. Dhillon, ''Consistent binary classification with generalized performance metrics,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2744–2752.

[52] D. Minnen, T. Westeyn, T. Starner, J. Ward, and P. Lukowicz, ''Performance metrics and evaluation issues for continuous activity recognition,'' in *Performance Metrics for Intelligent Systems*. Gaithersburg, MD, USA: NIST, 2006, pp. 141–148.

[53] A. Sharma, I. Manzoor, and N. Kumar, ''A feature reduced intrusion detection system using ANN classifier,'' *Expert Syst. Appl.*, vol. 88, pp. 249–257, Dec. 2017.

[54] L. Liao, Z. Yang, Q. Liao, K. Li, K. Li, J. Liu, and Q. Tian, ''A half-precision compressive sensing framework for end-to-end person re-identification,'' *Neural Comput. Appl.*, vol. 32, no. 4, pp. 1141–1155, 2020.

[55] D. Yogatama and G. Mann, ''Efficient transfer learning method for automatic hyperparameter tuning,'' in *Proc. Artif. Intell. Statist.*, 2014, pp. 1077–1085.

[56] R. Bardenet, M. Brendel, B. Kégl, and M. Sebag, ''Collaborative hyperparameter tuning,'' in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 199–207.

[57] M. D. Zeiler, ''ADADELTA: An adaptive learning rate method,'' 2012, *arXiv:1212.5701*.

[58] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, ''Activation functions: Comparison of trends in practice and research for deep learning,'' 2018, *arXiv:1811.03378*.

[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, ''Dropout: A simple way to prevent neural networks from overfitting,'' *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[60] Y. Gal and Z. Ghahramani, ''A theoretically grounded application of dropout in recurrent neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1019–1027.

[61] P. Kampstra, ''Beanplot: A boxplot alternative for visual comparison of distributions,'' *J. Stat. Softw.*, vol. 28, no. 1, pp. 1–9, Nov. 2008.

[62] D. Chicco and G. Jurman, ''The advantages of the Matthews correlation coefficient (MCC) over F_1 score and accuracy in binary classification evaluation,'' *BMC Genomics*, vol. 21, no. 1, p. 6, Dec. 2020.

[63] A. Chadha and B. Kaushik, ''A survey on prediction of suicidal ideation using machine and ensemble learning,'' *Comput. J.*, vol. 64, no. 11, pp. 1617–1632, Nov. 2021.

[64] K. M. Ghori, M. Imran, A. Nawaz, R. A. Abbasi, A. Ullah, and L. Szathmary, ''Performance analysis of machine learning classifiers for non-technical loss detection,'' *J. Ambient Intell. Hum. Comput.*, pp. 1–16, Jan. 2020, doi: 10.1007/s12652-019-01649-9.

YUSERA FAROOQ KHAN received the B.Tech. degree in computer science and engineering from BGSB University, Jammu and Kashmir, India, in 2012, and the master's degree in computer science and engineering from NIMS University, Rajasthan, in 2014. She has cleared GATE. From 2014 to 2018, she was an Assistant Professor at BGSB University. She has four years of teaching and research experience. Her research interests include machine learning, deep learning, transfer learning, neuroimage image analysis, and computer vision.

BAIJNATH KAUSHIK received the B.E. degree in computer science and engineering from Nagpur University, in 1997, the Master of Technology degree from the University School of Information Technology, GGSIPU, New Delhi, in 2009, and the Ph.D. degree in computer science and engineering from the IIT Dhanbad, in 2016. He has more than 21 years of teaching and research experience. He is currently an Associate Professor with the School of Computer Science and Engineering, SMVDU, Katra, Jammu and Kashmir. His research interests include machine learning, deep learning, nature inspired algorithms, soft computing, and parallel algorithms.

MOHAMMAD KHALID IMAM RAHMANI (Senior Member, IEEE) was born in Patherghatti, Kishanganj, Bihar, India, in 1975. He received the B.Sc. (Engg.) degree in computer engineering from Aligarh Muslim University, India, in 1998, the M.Tech. degree in computer engineering from Maharshi Dayanand University Rohtak, in 2010, and the Ph.D. degree in computer science engineering from Mewar University, India, in 2015. From 1999 to 2006, he was a Lecturer with the Maulana Azad College of Engineering and Technology, Patna. From 2006 to 2008, he was a Lecturer and a Senior Lecturer with the Galgotias College of Engineering and Technology, Greater Noida. From 2010 to 2011, he was an Assistant Professor with GSMVNIET, Palwal. Since 2017, he has been an Assistant Professor with the Department of Computer Science, College of Computing and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia. He has published more than 37 research papers in journals and conferences of international repute, three book chapters, and holds one patent of innovation. His research interests include algorithms, the IoT, cryptography, image retrieval, pattern recognition, machine learning, and deep learning.

MD EZAZ AHMED received the M.E. (Master of Engineering) degree in computer science & engineering from MDU University Rohtak, Haryana, India, in 2008, and the Ph.D. degree in computer science from Jodhpur National University, India, in 2013. He was with North Cap University (NCU) earlier known as ITM University, Gurgaon, Haryana, as an Assistant Professor, and other rank, from 2002 to 2013. Since 2014, he has been an Assistant Professor with the Department of Computer Science, College of Computing and Informatics, Saudi Electronic University. His research interests include data mining, web technology, machine learning, deep learning, data science, and the IoT.