# Alzheimer's Disease Detection from Spontaneous Speech using Attention Mechanism with Deep Neural Network and Transfer Learning Approach

Md Sadman Hasan, K M Tahzeem Zaman, Akib Ahmed, T I M Sakir Safkat, Md Motahar Mahtab, Md Humaion Kabir Mehedi and Annajiat Alim Rasel
*Dept of Computer Science and Engineering*
*Brac University*
Dhaka, Bangladesh
Email:{md.sadman.hasan@g.bracu.ac.bd, k.m.tahzeem.zaman@g.bracu.ac.bd, akib.ahmed@g.bracu.ac.bd, t.i.m.sakir.safkat@g.bracu.ac.bd, md.motahar.mahtab@g.bracu.ac.bd, humaion.kabir.mehedi@g.bracu.ac.bd, annajiat@gmail.com}

***Abstract*—Alzheimer's disease (AD) is a brain disorder that causes cognitive and functional abilities to deteriorate as the disease advances. It's one of the most common reasons for dementia. Considering its widespread existence, it has far-reaching consequences for more than just patients and caregivers, as it has a significant effect around the world. Although loss of memory is oftentimes considered as the hallmark cause of Alzheimer's disease, language impairment can also appear in the early stages of the disease. Early prognosis is critical because therapeutics can postpone advancement and give those diagnosed valuable time. Models that analyze spontaneous speech could render an efficient and accurate prognosis modality for earlier Alzheimer's disease diagnosis. We employed NLP approaches to classify dementia patients' spontaneous speech dataset from DementiaBank to forecast dementia in this work. Additionally, we used CNN-LSTM neural network models to distinguish between dementia and control patients. We discovered that using pre-trained embedding GloVE rather than random embedding culminated in a considerable betterment in accuracy when applying mentioned models. In comparison to training the entire dataset on a large number of neural network model parameters, the transfer learning technique to text categorization yielded more promising results. Precision, recall, F1 score, and specificity were all evaluated as performance indicators. In addition to combining the attention mechanism with the model, we performed hyper-parameter optimization of the CNN-LSTM model, which yielded excellent accuracy.**

***Index Terms*—AD (Alzheimer's Disease). NLP (Natural Language Processing), CNN (Convolutional Neural Network), LSTM (Long Short Term Memory)**

## I. Introduction

ALZHEIMER'S disease is a kind of dementia that causes memory, cognitive, and behavior problems. The symptoms eventually grow severe enough to interfere with daily activities. As the world's population ages, its impact on many areas of society grows [1]. While memory and cognitive decline are two of the most common symptoms of Alzheimer's disease, literature suggests that language impairment is also a common sign that can be used to support diagnosis and assessment of disease severity, given that speech and language production can provide information about a person's cognitive status and other aspects related to brain damage. Although human assessment of speech and language can be used to diagnose and assess patients in the clinical setting, it does not allow for objective quantitative analysis or repeatability. In this approach, the use of speech recognition and Natural Language Processing (NLP) techniques may result in the development of novel precision medicine tools such as objective measures and biomarkers [2]. Recurrent neural networks with long short-term memory (LSTM) have emerged as a viable and scalable paradigm for a variety of learning problems involving sequential input. Earlier approaches to these difficulties were either limited to a specific case or did not scale over long time periods. Because they are both wide and effective, LSTMs are valuable for capturing long-term temporal relationships. They don't have the optimization concerns that conventional recurrent networks have, and they've been used to advance the state of the art for a range of difficult applications. Handwriting recognition and creation, language modeling and translation, voice acoustic modeling, speech synthesis, protein secondary structure prediction, and audio and video data analysis are all part of this [3].

For building vector representations of words and texts, word embedding is among the most useful deep learning algorithms. Due to their capacity to capture the syntactic and semantic relationships between words, these techniques have sparked a promising future in text and sentiment analysis. Word2Vec and Global Vectors are two excellent deep learning techniques for word embeddings (GloVe). We present a novel strategy for enhancing the precision of pre-trained Word2Vec/Glove vectors in sentiment analysis tasks in this paper. Numerous sentiment datasets and numerous deep learning models from scientific papers were used to test the proposed approach. The technique, according to the findings, improves the sentiment analysis accuracy of pre-trained word embeddings vectors. These pre-trained embeddings can capture semantic definitions of words, but they are context-free and do not catch higher-level notions in contexts like polysemous disambiguation, syntactic structures, semantic roles, and anaphora [2]

Attention networks have recently acquired popularity. Attending the model not only enhances its efficiency, but it also produces understandable interim manifestations. As a result, attention can act as a suitable interface for delving into the inner workings of neural networks. Because analyzing raw attention values directly can be difficult for humans, visual representations that emphasize word alignments across phrases, like the bipartite graph and the attention matrix heatmap have been developed [4].

## II. Literature Review

### A. Natural Language Processing

NLP techniques based on words and concepts are only the first step toward natural language understanding. The word sensemaking was invented to describe the near-future of NLP, which is built on computational frameworks that have been biologically and semantically motivated to aid story understanding. Computational intelligence has a bright future ahead of the opportunity to play a significant influence in The study of NLP. For instance, fuzzy logic has a direct link to NLP as shown by [5] and also demonstrated by [6] as well as sentiment analysis linguistic summary shown respectively by [3], and inference of word meaning [7]. Artificial neural networks can be of assistance in completing tasks in Natural Language Processing. The research paper also sheds light on different methodologies of NLP namely Statistical NLP, Lexical NLP. Taxonomic, Endogenous, and Noetic NLP. The applied methodology differs in the case of intended application and results with Keyword Spotting being the go-to straightforward approach for most basic NLP functions. In the last 15 years, NLP research has not progressed at the same rate as other technologies. Despite the fact that NLP research has crafted meaningful strides in generating advanced ai behaviour, including such Google's, IBM's Watson, and Apple's Siri, neither of these NLP methodologies comprehend what they're on about, making them no different than a parrot looking to learn to repeat the same thing without recognizing what one is trying to say. Even the world's most common NLP technologies consider text analysis to be a problem of matching words or patterns. Trying to grasp a bit of text by looking at it at the pixel level is similar to trying to comprehend an image by looking at the word level [8].

### B. Attention Mechanism

Attention models are already ubiquitous in Nlp tasks. Attention can always be directed to distinct input portions, multiple renderings of the same data, or different functions to build a concise representation of data as well as to underline the important information. Attention was suggested by [7] for machine translation, while [9] explored its derivatives (2015). Both of these researches are already widely used in NLP. A distribution function is often used to make a choice, which considers locality in various dimensions, encompassing space, time, and semantics. Attention, which would be a key component of this paper's technique, was initially created and used in conjunction with LSTMs by [10],[11], and also to a lesser degree alongside CNNs by [12].

Attention can be employed to compare inputs to a query component using resemblance or relevance metrics. It can also acquire what really is relevant on its own without constructing a depiction that resembles the crucial facts. In paper [13], [14] it is point out the Sparse attention mechanisms and linearized attention as indicated by [15] are two interesting directions in this field [9]. These alternative attention mechanisms necessitate model training and are utilized as stand-ins for the original attention mechanism in order to speed up training or decrease computation to find relevance. As a result, incorporating attention into framework aims could result in massive efficiency improvements. Furthermore, attention can be considered as a tool to investigate network dynamics. For example, in order to identify the importance of context components, Paper [16] presents dynamic convolutions to anticipate separate convolution kernels only depending on the current time-step [8]. Within that effort, the research study provided a classification of attention models, allowing anyone to carefully chart and analyse a significant portion of the methodologies published in the literature. The study report also discussed how attention may be used to address fundamental AI difficulties, including how it could be used to pump information into a neural network in order to reflect particular aspects or to exploit previously learned knowledge gained in transferring learning contexts. The papers [11], [17] and [16] have all conducted gaze tracking procedures to examine personal interactions throughout time in cognitivism research. Attention and neural impulse motions are tightly related, according to papers [18] and [19].

According to the article, this might open up new and challenging research areas wherein attention might be employed to force the usage of subsymbolic models in combination with symbolic knowledge representations to perform reasoning tasks or manage natural language interpretation. Recent findings also suggest that attention could play a role in autonomous learning systems, guiding and focussing the supervised learning in the lack of antecedent supervision [10].

### C. Convolutional Neural Network

CNN or Convolutional Neural Networks are series and classes of artificial networks deployed to analyze and process visual imagery in high impedance deep learning. In the paper reviewed below, we look at how CNN processes and analyses said visual imagery and the results it accumulates while doing so, its accuracy and how its application enhances and aids in multivarious fields of computational performance. The advent of deep convolutional neural networks (CNN), suggested by [14], and the presence of huge annotated datasets, as described by [12]), have facilitated enormous advancements in image processing. As mentioned by in [17], enormous scope very much commented on datasets with delegate information dissemination highlights are fundamental for building more exact or generalizable models in information driven learning as expressed at [14] [17].Not at all like prior picture datasets utilized in PC vision, ImageNet [5] gives an immense library of over 1.2 million arranged regular photographs separated into 1000+ classes The CNN models that were trained on this database are the foundation for far better object tracking and segmentation techniques [5].There have been currently three major ways for using CNNs to properly classify medical images: using ”off-the-shelf CNN” features as complementary information channels to existing hand-crafted image features for chest X-rays coined in [18] and CT lung nodule identification by [8] and [9] respectively; performing unsupervised pre-training on natural or medical images and fine-tuning on medical target images using CNN or other types of deep learning models [20] [21]. To circumvent the ”curse of dimensionality,” [6], yet to be published, used a region-based classification 2.5D image resampling and an aggregation of random view categorization scores to collect a suitable series of training picture samples.Regardless of the different among natural and medical images, image classifications established for object identification in image features, (SIFT) introduced in [21] and the histogram of arranged slopes (HOG) introduced in [22], have been widely used in clinical image analysis for object identification and division. The ability of ImageNet pre-trained CNNs to detect and identify lung disease in X-ray and CT modalities has recently been reviewed. No medical image datasets, on the other hand, have yet to be used to fine-tune an ImageNet pre-trained CNN model.

### *D. Long Short Time Memory*

Several versions of the Long Short-term Memory (LSTM) architecture for RNN were published since its inception in 1995. In latest days, these networks have now become state-of-the-art methods for a variety of machine learning problems. This offers a chance to explore the role and value of certain computational elements in typical LSTM permutations. This research presents the first large-scale study of eight LSTM modifications on three main tasks: speech recognition, handwriting recognition, and polyphonic music modelling. This is the major study on LSTM networks to date, with the findings of 5400 experimental runs (about 15 years of CPU time). The results indicate that neither of the options can significantly outperform the standard LSTM design, with the forget gate and output activation function being the most essential components. They also remark that the hyperparameters under consideration are essentially independent, and they proposed ways to make them more successful. They examined the most common LSTM architecture (vanilla LSTM) and eight other variations of it on three benchmark problems: acoustic modelling, handwriting recognition, and polyphonic music modelling. The only variation between each variant and the vanilla LSTM is the flavour. This allowed them to examine how each of these changes affects the architecture's performance. [20] introduced the most extensively used LSTM configuration in this literature. It's known as vanilla LSTM, and it serves as a benchmark against which all other variants are compared. The vanilla LSTM employs full gradient training and incorporates modifications to the original by [23] and [19].

### *E. Potential Challenges*

*1) Hyperparameters:* In the NLP and machine learning fields, hyper-parameter optimization has gotten a lot of attention at [22]. To perform successfully, learning approaches such as logistic regression and support vector machines, as well as the more advanced model types such as boosted regression trees and neural networks, rely on the correct instantiation of its hyperparameters. There will only be a certain quantity of training data that can be adjusted. In large-scale problems in which the volume of data is so large that even a quadratic running time is too long, hyper-parameter choices can make all the difference between poor and state-of-the-art execution [24].

*2) Overfitting:* Overfitting is defined as the application of modelling or procedures that deviate from parsimony, i.e., add additional words or utilise more complex techniques than are needed. It's critical to differentiate between the two types of overfitting: To begin with, the implemented approach is more flexible than it should be. Because it can accommodate some bending relationships, a neural net is more flexible than a conventional linear regression. When implemented to a data set with a linear model, however, it adds complexity without expanding the number of features. Despite the fact that it would have a negligible impact on system performance and productivity [12]. Secondly, while the deployed model carries unnecessary components. For instance, a polynomial of excessive degree or multiple linear regression with both irrelevant and necessary factors. Overfitting can be bothersome for a variety of reasons. A couple of them are stated as: Adding predictors that serve no purpose means that you'll have to measure and record these predictors to replace their values in the model if you want to utilise the regression to make predictions in the future. This not only costs resources, but also increases the chances of undetected database faults leading to forecast errors [17]. In a feature selection challenge, models with unnecessary predictors produce worse results. For example, in drug discovery, a blunder in using the amount of NH2 groups in a QSAR model when this number is meaningless will result in compounds being incorrectly ignored based on their irrelevant number of NH2 groups. This can result in the loss of valuable leads [23]. Training Time The total time consumed by the approach proposed in this paper is larger than the FCN algorithm due to the high degree of complexity in the ROIAlign layer computation. The time it takes to train datasets is determined by a variety of factors, including hyperparameters and optimizer selection. As a result, the training and prediction times on the networks are not the ultimate benchmarks as shown by [16]. The batch normalisation layer minimises the amount of time required for training. It enables the Convolutional Neural Network to handle the weightage issue by utilising high learning rates.

## III. Data Collection and Preprocessing

### *A. Dataset Collection*

The data used for this project is Dementia Bank Dataset. The dataset contains speech of 3272 patients of which 1676 are non-dementia patients and 1596 are dementia patients.

### *B. Data Preprocessing*

We imported a.tsv folder includes original data from Dementia sufferers' continuous speech. Stopwords are frequently used words that should be deleted from data since they bring no value to the research. These terms have little or no meaning. We used the NLTK library, which contains a set of words that are deemed stopwords in the English language. We used our custom stopwords as for example : [’more’, ’most’, ’must’, ”mustn’t”, ’my’, ’myself’, ’need’, ”needn’t”, ’no’, ’nor’, ’not’, ’now’, ’o’, ’of’, ’off’, ’on’, ’once’, ’only’, ’or’, ’other’, ’our’, ’ours’, ’ourselves’, ’out’, ’over’, ’own’, ’re’, ’s’, ’same’]. Furthermore, we removed all the characters except for alphabets (e.g. ‘’, ‘*’, “”). Also converted all the sentences to lower case and tokenized each word and performed lemmatization. In the lemmatization step, it is made sure that the words don’t lose their meaning. Lemmatization has a predefined dictionary that stores the context of words and checks with the word in the dictionary while diminishing. It works better than the stemming approach as in the root form of the words loses the meaning or it is not diminished to a proper English word. After this, we dropped all the nan values from our dataset. Finally, we saved our clean pre-processed dataset into pickle format for the implementation of our models. We transformed the transcript of the data to list and using train-test-split from sklearn we split the train and test set into 0.2 ratios. That is 80 percent of the data was used for training and 20 percent for testing.

### C. Text Preprocessing

We have done the most common text preprocessing using Keras Api which includes Tokenization, sequencing and padding. Deep learning models do not understand text, so tokenization is used to convert sentences into words and encode those into integers. To have the same size of inputs, padding is done. In our case, we used the post padding sequence.

### D. Word Embedding

Word Embedding is a learnt text representation in which words with related meanings are represented similarly. Individual document is represented as real-valued vectors in a predetermined vector space in this method. Each word is allocated to a single vector, as well as the vector values are learned in a neural network-like manner. The most common application of word embedding is to assess similarities in context or usage. The calculation of similarity between the two vectors is generally done with the use of a metric such as with the Cosine Similarity (i.e., a normalised dot product).

## IV. Pre-Trained Embedding and Transfer Learning

For transfer learning purposes we have used pre-trained embedding which is usually done to use the already learned embedding for solving our model task which is similar to the task on which this embedding was trained. These word embeddings capture the semantic and syntactic meaning of a word as they are trained on a larger dataset and help in boosting the performance of the NLP models. For our purpose, we have used Stanford’s GloVe word embedding. This will aid in deriving the relationship between the words with the application of simple statistical calculation using the co-occurrence matrix or count matrix. Count matrix works by increasing the counter whenever it finds a word in the context of the other. GloVe learns to encode the information of the probability ratio in the form of word vectors. The most general form of the model is given by:

$$F(\mathrm{w_i}, \mathrm{w_j}, \tilde{\mathrm{w}}_\mathrm{k}) = \frac{\mathrm{P_{ik}}}{\mathrm{P_{jk}}}$$

### A. Optimizer

RMSprop (Root Mean Square Propagation) attempts to attenuate oscillations more effectively than momentum optimization. It also dynamically modifies the learning rate and selects a various learning rate for every variable. The RMSprop optimization parametric equation is as follows:

For each Parameter $\mathrm{w}^j$

*j subscript dropped for clarity*

$$\nu_\mathrm{t} = \rho\nu_\mathrm{t-1} + (1-\rho) * \mathrm{g_t^2}$$

$$\Delta\omega_\mathrm{t} = -\frac{\eta}{\sqrt{\nu_\mathrm{t} + \epsilon}} * \mathrm{g_t}$$

$$\omega_\mathrm{t+1} = \omega_\mathrm{t} + \Delta\omega_\mathrm{t}$$

$\eta$: *Initial Learning Rate*

$\nu_t$: *Exponential Average of squares of gradients*

$\mathrm{g}_t$: *Gradient at time t along* $\omega^\mathrm{j}$

Adaptive Moment Optimization is also known as Adam Optimizer algorithms works better by combining the heuristics of both the momentum and RMSprop optimizers. For our CNN model, we preferred Adam optimizer as it performs better in this scenario. The parametric equation of adam optimizer is shown below:

For each Parameter $\mathrm{w}^j$

*j subscript dropped for clarity*

$$\nu_\mathrm{t} = \beta_1 * \nu_\mathrm{t-1} - (1-\beta_1) * \mathrm{g_t}$$

$$\mathrm{s_t} = \beta_2 * \mathrm{s_{t-1}} - (1-\beta_2) * \mathrm{g_t}^2$$

$$\Delta\omega_\mathrm{t} = -\eta\frac{\nu_\mathrm{t}}{\sqrt{\mathrm{s_t} + \epsilon}} * \mathrm{g_t}$$

$$\omega_\mathrm{t+1} = \omega_\mathrm{t} + \Delta\omega_\mathrm{t}$$

$\eta$: *Initial Learning Rate*

$\mathrm{g}_t$: *Gradient at time t along* $\omega^\mathrm{j}$

$\nu_t$: *Exponential Average of gradients along* $\omega_\mathrm{j}$

$\mathrm{s}_t$: *Exponential Average of squares of gradients along* $\omega_\mathrm{j}$

$\beta_1, \beta_2$: *Hyperparameters*

## V. Classification Methods

### A. Attention Mechanism

We imported the sequential model from Keras and used an embedding layer of vocab length to 847 and input length of 100. We designed our attention layer with attention weight and bias with initializers of normal and zeros respectively. We also applied the tan hyperbolic function followed by a softmax as an activation function to get normalized alignment scores for our output. We used a convoluted 1D layer followed by a max-pooling layer, dropout layer and a bidirectional lstm layer on top of the attention model. We defined the dense layer with activation "relu" and kernel-initializer of "he-uniform" and the final output layer was defined with activation "sigmoid" and kernel=initializer of "glorot-unifrom". All the best-fit parameters were obtained after hyperparameter tuning of the models shown in Fig. 1

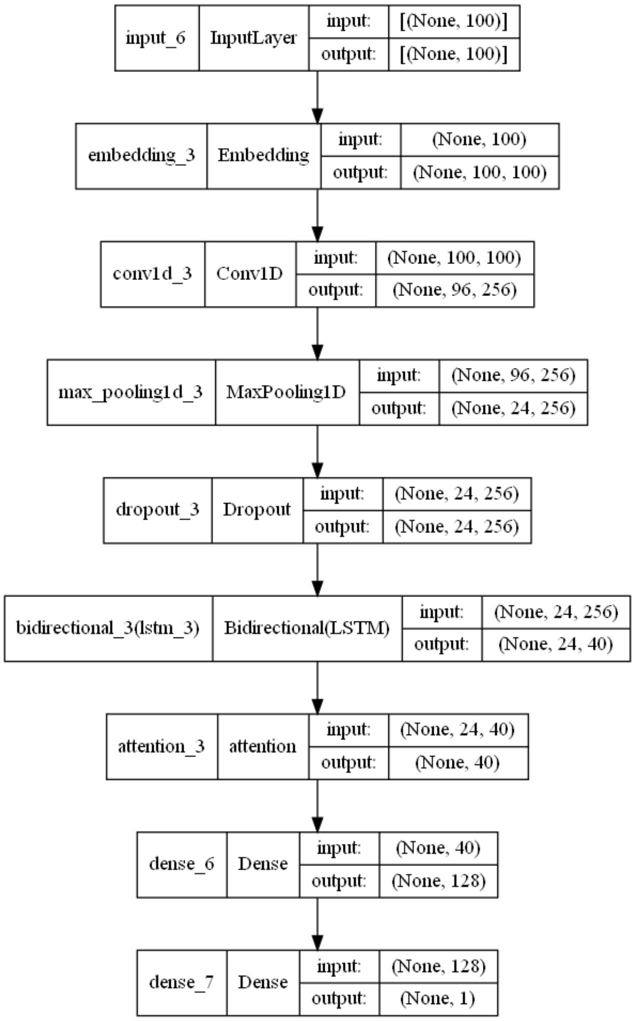
input_6 | InputLayer | input: [(None, 100)] | output: [(None, 100)]
embedding_3 | Embedding | input: (None, 100) | output: (None, 100, 100)
conv1d_3 | Conv1D | input: (None, 100, 100) | output: (None, 96, 256)
max_pooling1d_3 | MaxPooling1D | input: (None, 96, 256) | output: (None, 24, 256)
dropout_3 | Dropout | input: (None, 24, 256) | output: (None, 24, 256)
bidirectional_3(lstm_3) | Bidirectional(LSTM) | input: (None, 24, 256) | output: (None, 24, 40)
attention_3 | attention | input: (None, 24, 40) | output: (None, 40)
dense_6 | Dense | input: (None, 40) | output: (None, 128)
dense_7 | Dense | input: (None, 128) | output: (None, 1)

Fig. 1: Model Summary for Attention Mechanism

### B. CNN Model

We used convolutional neural networks for recognizing patterns in the continuous speech of our textual dataset. In our model, we used 1D convo filters instead of 2D filters for text classification. For overcoming the overfitting problem we used a dropout ratio of 0.2 along with a kernel initializer. For our hidden layers, we used the activation function as "relu" and "he-uniform" as our kernel initializer while for the classification layer "sigmoid" was used as the activation function. We also compiled the model with the "Adam" optimizer and loss function as "binary-crossentropy" for the binary classification of our dataset. After appropriate hyperparameter tuning of our CNN model we used the following optimized parameters with our layer adjustments shown in Fig. 2

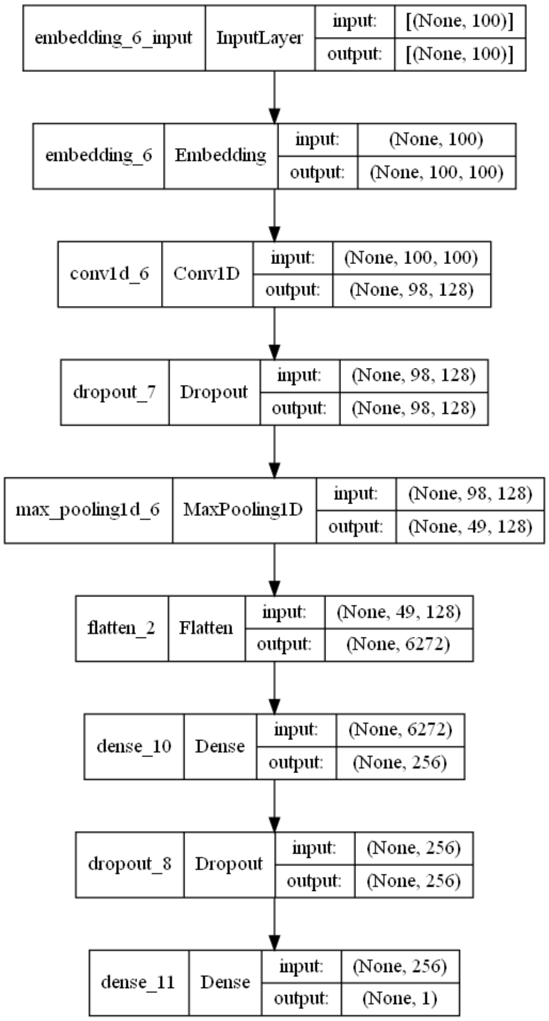
embedding_6_input | InputLayer | input: [(None, 100)] | output: [(None, 100)]
embedding_6 | Embedding | input: (None, 100) | output: (None, 100, 100)
conv1d_6 | Conv1D | input: (None, 100, 100) | output: (None, 98, 128)
dropout_7 | Dropout | input: (None, 98, 128) | output: (None, 98, 128)
max_pooling1d_6 | MaxPooling1D | input: (None, 98, 128) | output: (None, 49, 128)
flatten_2 | Flatten | input: (None, 49, 128) | output: (None, 6272)
dense_10 | Dense | input: (None, 6272) | output: (None, 256)
dropout_8 | Dropout | input: (None, 256) | output: (None, 256)
dense_11 | Dense | input: (None, 256) | output: (None, 1)

Fig. 2: Model summary for our CNN model

### C. CNN-LSTM

Traditional neural network models face short-term memory problems which leads to the vanishing gradient problem. To overcome this problem implementation of the LSTM network plays an important role by memorizing certain patterns and performs better than traditional recurrent neural networks. In the LSTM network, the relevant information is kept and irrelevant information is discarded in each cell. According to our dataset, we defined the LSTM layer with 20 neurons to best fit our model and all the parameters were obtained from hyper tuning and most desirables were used. The layer was added following the max-pooling layer after feature extractions using the convoluted 1D layers. For our classification layer, we used the activation function “sigmoid” and kernel-inilializer as “glorot-uniform” and the model was compiled with “RMSprop” as optimizer and “binary-crossentropy” as the loss function. A summary of the model is shown in Fig. 3

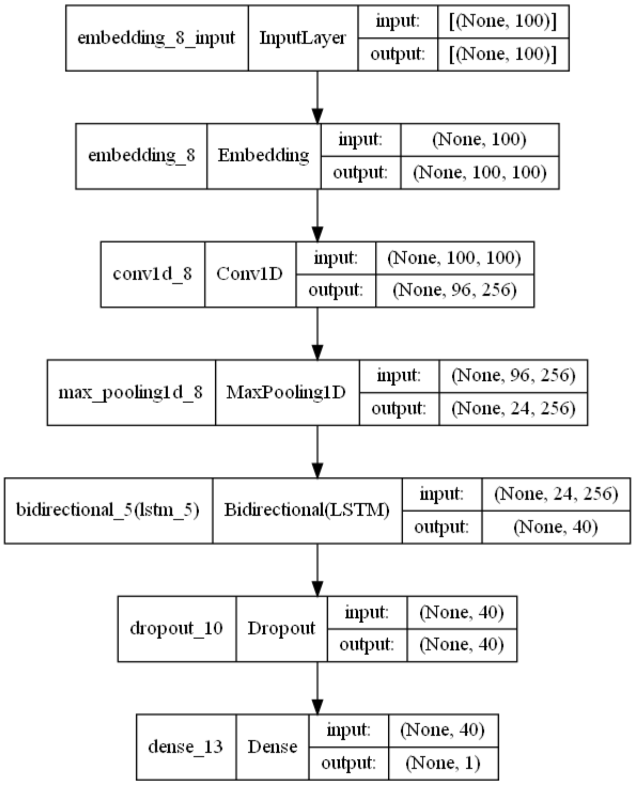

Fig. 3: Model summary for our CNN-LSTM model

## VI. Result Analysis

### A. *CNN model without Attention mechanism*

We implemented our CNN model with optimized parameters and best-fit activation function, dropout ratio and optimizers to test how our model runs without the attention mechanism. The Classification Report of CNN model without Attention Mechanism shown in TABLE I

| Parameters | Score |
|---|---|
| Precision | 0.8758 |
| Recall | 0.8273 |
| F1 | 0.8509 |
| Specificity | 0.8947 |

TABLE I: CNN model without Attention Mechanism metrics

The AUC curve for the CNN model is shown in Fig. 4

### B. *CNN-LSTM without Attention mechanism*

After testing our dataset with the CNN model we added a bidirectional LSTM layer on top of the max-pooling layer of CNN after 1D convolution and also set the dropout ratio to 0.5 for overcoming the overfitting problem. In this case, we used the RMSprop optimizer which was the best fit for this model. The Classification Report of CNN-LSTM model without Attention Mechanism is shown in TABLE II:

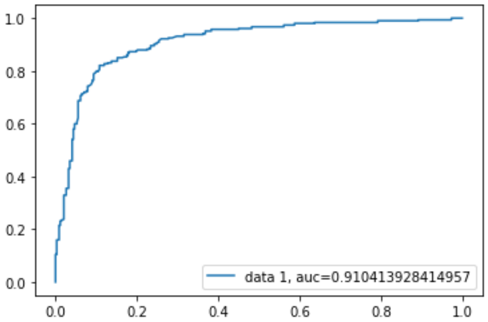

Fig. 4: AUC curve for CNN without Attention Mechanism

| Parameters | Score |
|---|---|
| Precision | 0.8576 |
| Recall | 0.8631 |
| F1 | 0.8603 |
| Specificity | 0.8713 |

TABLE II: CNN-LSTM model without Attention Mechanism metrics

The AUC curve for the CNN-LSTM model is shown in Fig. 5

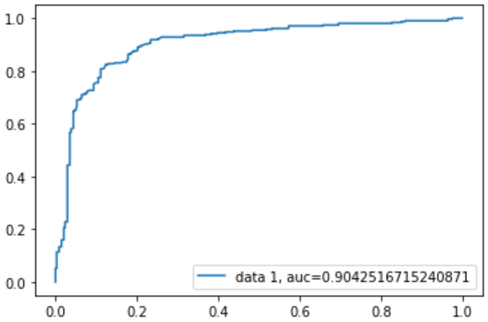

Fig. 5: AUC curve for CNN-LSTM Network without Attention Mechanism

### C. *CNN-LSTM with attention mechanism*

After implementation of attention mechanism along with CNN-LSTM network with optimized parameter and adam as our optimizer we got the Classification Report for our CNN-LSTM shown in TABLE III

| Parameters | Score |
|---|---|
| Precision | 0.8888 |
| Recall | 0.8338 |
| F1 | 0.8605 |
| Specificity | 0.9064 |

TABLE III: CNN-LSTM Network with Attention mechanism metrics

The AUC curve for the model highlighting the how accurately our classification was performed between the classes is shown in Fig. 6

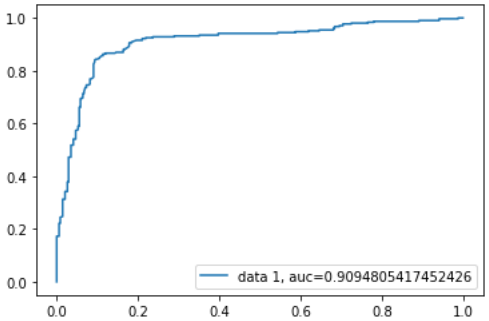

Fig. 6: AUC curve for CNN-LSTM with Attention Mechanism

We used the transfer learning approach using the pre-trained word embedding with appropriate hyperparameter tuning for best-fit parameters for our implemented models and found a significant result in the accuracy. From the 3 models implemented we found using the attention mechanism with the CNN-LSTM network yielded the most accurate score. A comparison of the accuracy of the models are shown in a bar chart shown in Fig. 7

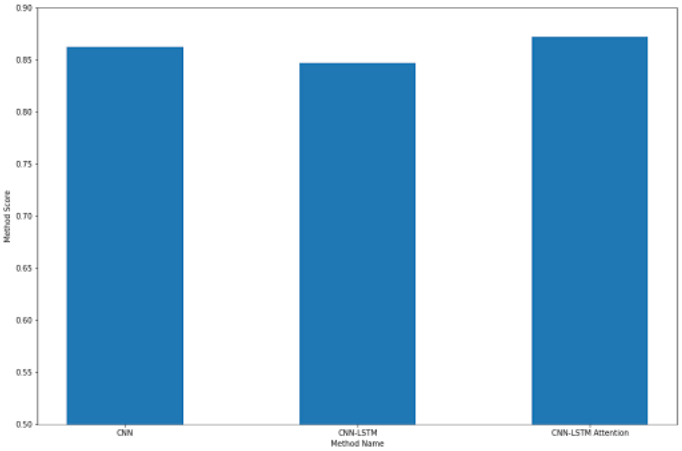

Fig. 7: Accuracy comparison of the models used

## VII. Conclusion

We implemented the Convolutional neural network model with LSTM as a hybrid model for the prediction of dementia patients, we implemented the model with and without attention mechanism and analysed the differences in their accuracy. We found a significant increase in classification accuracy using the global embedding, GloVE as a transfer learning approach. We faced difficulties in the tuning of hyperparameters for the model in order to find the best-fit parameters for optimal performance. After plugging in all the best-fit parameters we found using the attention mechanism with the CNN-LSTM network yielded the most accurate score of 87.21%. In our future work, we wish to implement some of the pre-trained models for text classification like XLNet, BPT while combining them with GloVE embedding which hopefully will improve test accuracy.

## References

[1] S. Saxena, M. Funk, and D. Chisholm, "World health assembly adopts comprehensive mental health action plan 2013–2020," *The Lancet*, vol. 381, no. 9882, pp. 1970–1971, Jun. 2013, ISSN: 01406736. DOI: 10.1016/S0140-6736(13)61139-3. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0140673613611393.

[2] R. Pappagari, J. Cho, L. Moro-Velázquez, and N. Dehak, "Using state of the art speaker recognition and natural language processing technologies to detect alzheimer's disease and assess its severity," in *Interspeech 2020*, ISCA, Oct. 2020, pp. 2177–2181. DOI: 10.21437/Interspeech.2020-2587. [Online]. Available: https://www.isca-speech.org/archive/interspeech_2020/pappagari20_interspeech.html.

[3] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017, ISSN: 2162-237X, 2162-2388. DOI: 10.1109/TNNLS.2016.2582924. [Online]. Available: http://ieeexplore.ieee.org/document/7508408/.

[4] S. Liu, T. Li, Z. Li, V. Srikumar, V. Pascucci, and P.-T. Bremer, "Visual interrogation of attention-based models for natural language inference and machine comprehension," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, 2018, pp. 36–41. DOI: 10.18653/v1/D18-2007. [Online]. Available: http://aclweb.org/anthology/D18-2007.

[5] E. Cambria and B. White, "Jumping nlp curves: A review of natural language processing research [review article]," *IEEE Computational Intelligence Magazine*, vol. 9, no. 2, pp. 48–57, May 2014, ISSN: 1556-603X. DOI: 10.1109/MCI.2014.2307227. [Online]. Available: http://ieeexplore.ieee.org/document/6786458/.

[6] H.-C. Shin, H. R. Roth, M. Gao, *et al.*, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, May 2016, ISSN: 0278-0062, 1558-254X. DOI: 10.1109/TMI.2016.2528162. [Online]. Available: https://ieeexplore.ieee.org/document/7404017/.

[7] A. Alishahi, G. Chrupała, and T. Linzen, "Analyzing and interpreting neural networks for nlp: A report on the first blackboxnlp workshop," *Natural Language Engineering*, vol. 25, no. 4, pp. 543–557, 2019. DOI: 10.1017/S135132491900024X.

[8] E. Sood, S. Tannert, D. Frassinelli, A. Bulling, and N. T. Vu, "Interpreting attention models with human visual attention in machine reading comprehension," in *Proceedings of the 24th Conference on Computational Natural Language Learning*, Association for Computational Linguistics, 2020, pp. 12–25. DOI: 10.18653/v1/2020.conll-1.2. [Online]. Available: https://www.aclweb.org/anthology/2020.conll-1.2.

[9] Y. Dong, C. Bhagavatula, X. Lu, *et al.*, "On-the-fly attention modulation for neural generation," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Association for Computational Linguistics, 2021, pp. 1261–1274. DOI: 10.18653/v1/2021.findings-acl.107. [Online]. Available: https://aclanthology.org/2021.findings-acl.107.

[10] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2016, pp. 2249–2255. DOI: 10.18653/v1/D16-1244. [Online]. Available: http://aclweb.org/anthology/D16-1244.

[11] M. Artetxe, G. Labaka, and E. Agirre, "Learning bilingual word embeddings with (almost) no bilingual data," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 451–462. DOI: 10.18653/v1/P17-1042. [Online]. Available: https://aclanthology.org/P17-1042.

[12] R. Collier, "An historical overview of natural language processing systems that learn," *Artificial Intelligence Review*, vol. 8, no. 1, pp. 17–54, 1994, ISSN: 0269-2821, 1573-7462. DOI: 10.1007/BF00851349. [Online]. Available: http://link.springer.com/10.1007/BF00851349.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[15] L. Wang, M. Feng, B. Zhou, B. Xiang, and S. Mahadevan, "Efficient hyper-parameter optimization for nlp applications," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2015, pp. 2112–2117. DOI: 10.18653/v1/D15-1253. [Online]. Available: http://aclweb.org/anthology/D15-1253.

[16] Y. Wang, B. Zheng, D. Gao, and J. Wang, "Fully convolutional neural networks for prostate cancer detection using multi-parametric magnetic resonance images: An initial investigation," in *2018 24th International Conference on Pattern Recognition (ICPR)*, IEEE, Aug. 2018, pp. 3814–3819, ISBN: 9781538637883. DOI: 10.1109/ICPR.2018.8545754. [Online]. Available: https://ieeexplore.ieee.org/document/8545754/.

[17] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[18] Y. Bar, I. Diamant, L. Wolf, S. Lieberman, E. Konen, and H. Greenspan, "Chest pathology detection using deep learning with non-medical training," in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, IEEE, Apr. 2015, pp. 294–297, ISBN: 9781479923748. DOI: 10.1109/ISBI.2015.7163871. [Online]. Available: http://ieeexplore.ieee.org/document/7163871/.

[19] F. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 3, Jul. 2000, 189–194 vol.3. DOI: 10.1109/IJCNN.2000.861302.

[20] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5–6, pp. 602–610, Jul. 2005, ISSN: 08936080. DOI: 10.1016/j.neunet.2005.06.042. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0893608005001206.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004, ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029664.99615.94. [Online]. Available: http://link.springer.com/10.1023/B:VISI.0000029664.99615.94.

[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, IEEE, 2005, pp. 886–893, ISBN: 9780769523729. DOI: 10.1109/CVPR.2005.177. [Online]. Available: http://ieeexplore.ieee.org/document/1467360/.

[23] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000, ISSN: 0899-7667, 1530-888X. DOI: 10.1162/089976600300015015. [Online]. Available: https://direct.mit.edu/neco/article/12/10/2451-2471/6415.

[24] K. Eggensperger, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Surrogate benchmarks for hyperparameter optimization," in *Proceedings of the 2014 International Conference on Meta-Learning and Algorithm Selection - Volume 1201*, ser. MLAS'14, Prague, Czech Republic: CEUR-WS.org, 2014, pp. 24–31, ISBN: 16130073.