UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

Text Classification for Major Depressive Disorder (MDD) Symptoms and Treatments Using Convolutional Neural Networks (CNN)

Felicia Chin Hui Fen per 1st Affiliation
Faculty of Computing
Universiti Teknologi Malaysia
Skudai, Johor, Malaysia
feliciahui@graduate.utm.my

***Abstract*—This study aims to develop a classification model for Major Depressive Disorder (MDD) symptoms and treatments using Convolutional Neural Networks (CNN) using 5000 abstract medical journals from NCBI. The data was retrieved from the NCBI website and underwent text preprocessing. Three CNN models were built: benchmark CNN model, a proposed model, and a proposed model using Word2vec embedding matrix. Each model was tested using three different dataset ratios. Performance was evaluated using accuracy, precision, recall, F1-Score, and confusion matrix metrics. Based on the results, 70.00% training, 15.00% testing, and 15.00% validation splitting were the most suitable dataset splitting ratio. The proposed CNN model outperformed other models with an accuracy of 92.00% without model overfitting. The use of Word2vec did not significantly improve the results. This research contributes to the limited exploration of text classification for MDD symptoms and treatments in medical journals using CNNs.**

Keywords — Text Classification, Major Depressive Disorder (MDD), Symptoms and Treatments, Convolutional Neural Network (CNN).

I. INTRODUCTION

Major Depressive Disorder (MDD), a prevalent mental disorder, is challenging to diagnose and treat due to its complex origins and lack of biological indicators [1]. While the root cause of the depression is still unknown, it is generally believed that Major Depressive Disorder (MDD) is a disorder that results from the psychological, interaction of social, and biological factors of an individual. As a result, there is no specific explanation that can conclusively explain the pathogenesis of this disorder from genetics, neurobiology or neuroimaging perspective. Consequently, earlier diagnosis and treatment for Major Depressive Disorder (MDD) can be challenging.

One of the techniques for analysing text and extracting essential patterns from natural or human languages is natural language processing (NLP) [2]. It includes numerous jobs that rely on different statistics and data-driven computation techniques. Text classification is a type of Natural language processing (NLP) where it helps to assign tags or labels to the textual contents [3]. This includes queries, sentences, paragraphs, and documents. Convolutional Neural Networks (CNN) which is a type of deep learning method that shows significant results in the text classification.

Major Depressive Disorder (MDD) does not have any biological indicators [4]. Thus, symptomatology is the only method available for pre-diagnosis, which further complicates the task of recognizing its symptoms. Additionally, Major Depressive Disorder (MDD) treatments are ineffective because they require different types of medication, time, planning, and are specific to each patient. Recent studies have shown the potential of text classification applying natural language processing methods to identify depressive symptoms. For example, in [5], the author used Health Questionnaire-9 and various types of deep learning algorithms, which also included Convolutional Neural Networks (CNN), to identify text data fromon social media. The findings demonstrated that pre-diagnosis of depression using text categorization is possible, with CNN demonstrating good accuracy in recognizing depressive symptoms.

Major Depressive Disorder (MDD) is a critical mental disorder that is difficult to diagnose and treat due to its multifactorial nature. Early detection of MDD symptoms is essential for successful treatment; however identifying these symptoms is difficult due to the lack of biological markers. Furthermore, treatment is also unique for each individual, and it takes time and preparation. In order to pre-diagnose Major Depressive Disorder (MDD) based on symptoms, an effective and efficient method must be developed.

This study aims to address these challenges by developing a Convolutional Neural Networks (CNN) based text classification model to identify MDD symptoms and treatments from medical journals. The objectives include identifying features related to MDD symptoms and treatments, implementing text classification using CNN, and evaluating the performance of this approach. The research questions focus on the features of MDD symptoms and treatments in medical journals, the implementation of the CNN-based text classification method, and the evaluation of this machine learning methodology. The literature review will present in Section II, methodology of this research will present in Section III, result and discussion will present in Section 1V and V, and conclusion will be made in Section VI.

II. Literature Review

A. *Text Classification*

Text classification is the process of automatically categorising a collection of documents into one or more labelled or predefined groups based on their contents [6]. Text classification is important for organising large numbers of documents. It can be used in information management applications to help group documents into one or more labelled or predefined groups automatically based on the given category. Text classification can help solve natural language challenges, which include topic modelling, news classification, sentiment classification, language translation, and question answering [7]. Data collection, text preprocessing, feature extraction, dimensionality reduction, classification techniques, and performance evaluation are the main stages of the text classification process [6].

B. *Word Embedding*

Feature extraction in deep learning models is mainly performed using word embedding methods [8]. Word embedding techniques play an important role in deep learning models by providing appropriate input features in downstream tasks, especially in text classification [9]. Word embedding represents a word in fixed-length vectors that consist of continuous real numbers. This aids in the transfer of a vocabulary word to a latent vector space where terms with similar contexts are grouped together. Word embedding techniques include Word2vec, GloVe, and FastText [8]. Google developed the pre-trained Word2vec model for sentiment analysis. This effectively combines the continuous bag-of-words and skip-gram architectures for building word vector representations. The GloVe model uses unsupervised learning to identify meaningful vector representations of words. The comparable task of natural language processing can be resolved using pre-trained word embeddings, which have already been trained on a larger corpus. FastText is another method of learning word representation created by Facebook's AI. The FastText method is essentially a library that is used to acquire word vector representations and learn word embeddings.

1) *Keras Embedding*

Keras provides a Tokenizer function that helps to vectorise the corpus. This will help to create the embedding layer by converting the text data into a sequence of integers. Each integers the index of a token in a dictionary [10]. A typical neural network can accept these input values that contain the encoded integer values. This is because the Keras library will provide the pre-building word embedding layer from the training dataset using the Tokenizer API. The layer will be flexible to embed all the words with the random weight initialised from the training dataset [11].

2) *Word2vec*

Word2Vec is a word embedding method that converts each word in a text into a vector that can represent the semantic meaning of the word [12]. The input, projection, and output layers make up its three layers. One of the Word2Vec methods' hidden layers is the projection layer. The projection layer and output layer will be connected by the weights, as well as the input layer and projection layer. The word in an n-gram context will be connected into continuous vectors by the projection layer. Words tend to be triggered by the same weight when they are together or appear frequently in an N-gram context, so there is a correlation between them. The V x N size W matrix can be used to represent weights between the input layer and the projection layer, where V is the dimension of the input layer and N is the dimension of the projection layer. A matrix of size N x D is used to represent matrix W between the projection layer and the output layer, where N is the projection layer's dimension and D is the output layer's dimension.

There are two components in Word2Vec which are CBOW and skip-gram. Word2vec in the CBOW model employs words that precede and follow the target whereas skip-gram employs words that come before and after the word. The window places a limit on both. The kernel for gathering input and target words is a window. The window is shifted from the beginning to the end of the text.

C. *Deep Learning Approaches*

Deep learning is a type of machine learning and an artificial intelligence frontier [13]. It consists of many classifiers that work together with linear regression and some activation functions. Activation functions imitate the structure of the human brain by combining multiple neural nodes and layers, resulting in a nonlinear relationship between the input and output layers. These are known as neural nodes or neural networks. A neural unit, or perception, is what the classifier node is called. Deep learning differs from traditional statistical learning in that it uses many neural nodes rather than just one, which is known as linear regression. Deep learning algorithms have many layers between input and output, and each layer may contain thousands of neural units. These layers are also referred to as hidden layers, and the node is referred to as a hidden node. One feature that distinguishes deep learning is its ability to create the network by itself rather than manually writing a complex hypothesis. Therefore, deep learning is a powerful technique for understanding nonlinear relationships. CNN, RNN, LSTM, GRU are the general deep learning algorithms that can be used to perform the text classification.

D. *Multi-label Text Classification*

Multi-label classification (MLC) allows for assigning multiple labels to an input instance [14]. It is a type of classification that is not only of theoretical interest but also involves practical applications in different real-world situations. These include music classification, gene function classification, image classification, document classification, and disease classification. Multi-label text classification (MLTC) is one of the key subfields of multi-label learning [15]. It is mostly used in question answering, dialog behaviour classification, topic labelling, and sentiment analysis.

In the recent studies found, [11] utilised several deep learning algorithms, NN, CNN, RNN, LSTM, Bidirectional LSTM, GRU, and Bidirectional GRU to classify the toxic comment levels using the dataset from Wikipedia comment. The results show that CNN model ranked second in classifying toxic comment levels and was able to achieve more than 80.00% precision, recall and F1-score.

Furthermore, author in [16] had proposed two new large corpora to perform the text classification using several deep learning models. The deep learning models used by the author included CNN, BIGRU, BILSTM, CGRU, CLSTM, GRU, HANGRU, HANLSTM, and LSTM. These nine deep learning models have been tested for single-label and multi-label datasets. Besides that, the Word2Vec method is also used to improve the text classification. From the result, CNN models performed well, achieving 87.23%, when the average performance for single-label datasets was 85.91%. For a maximum subset of eight categories from the multi-label dataset (SkyNewsArabia), CNN has the highest accuracy with confidence greater than 50% out of 70.34% of the overall performance among the deep learning models. Moreover, when the Word2Vec feature was applied to the experiment, the top accuracy results for the CNN model were improved overall by 3.00%.

Besides that, the research done by [15] on the multi-label text classification based on tALERT-CNN, which uses the Latent Dirichlet Allocation (LDA) topic model and A Lite BERT for Self-supervised Learning of Language Representations (ALBERT) model in TextCNN, a CNN model that used for text data. The result shows that the proposed model is able to outperform with precision of 89.32%, recall of 85.59%, and F1 score 87.92%, which shows its effectiveness in multi-label text classification.

Lastly, a comparative study on word embeddings in deep learning for text classification was done by [9]. The author had used the CNN algorithm and BiLSTM model in his study. From the research result, it showed that the performance of the CNN algorithm is more than 70.00% accuracy. The result increased to more than 80.00% when the Word2Vec model was applied for single-labelled datasets. Other than that, the CNN model is able to perform better than the BiLSTM model when multi-label datasets are used. The result was able to achieve more than 50.00% accuracy and 70.00% macro-F1 score compared to single-labelled datasets. Moreover, when the author applied the Word2Vec, the accuracy of the result increased by 5.32% and the micro-F1 score increased by 3.30% on average.

The study of identifying the symptoms of MDD has been done a lot using text classification, either using machine learning or deep learning approaches. However, the datasets mostly used were mostly from social media, websites, and clinical datasets of the patients; research based on medical journal articles was rarely found. Furthermore, the text classification mostly focused on identifying symptoms of MDD, and most of the treatments were only predictions. This is mainly because MDD is a mental disorder, and the treatments for it will need a professional's aid. However, if there is a way for an individual to self-check their symptoms and find a possible treatment, it will be very helpful for them.

Furthermore, the datasets chosen in this research are from medical journals where the information can be trusted and reliable. Moreover, extracting textual data from medical journal datasets can be done, as previous research has used it before. Therefore, features of symptoms and treatments of MDD are able to be extracted in this study. Besides that, deep learning approaches such as CNN algorithms are able to perform well in text classification, as shown in recent research, where they generally obtained a range of accuracy from 70.00% to 80.00% from all the recent studies found. Deep learning has the ability to create the network by itself rather than manually writing a complex hypothesis, which really reduces the workload of the study. In this study, the CNN algorithm is chosen to be the classifying technique because of its better accuracy and performance compared with other deep learning algorithms. This can be seen in previous research, such as the author in [16], where CNN took second place with an accuracy of 70.34% in classifying multi-label text data with maximum eight categories. For the feature extraction steps, the word embedding method, Word2Vec, is chosen since it is able to improve the accuracy of the CNN model based on the recent study.

Generally, the study of text classification for MDD symptoms and treatments using CNN algorithms is feasible as it provides a way to identify the related features of MDD symptoms and treatments in medical journals. Besides that, this study will determine whether text classification is able to be performed in order to classify the MDD symptoms and treatments using CNN algorithms in the medical journal datasets, as previous research has preferred social media and clinical datasets. Lastly, this study will also evaluate the accuracy of CNN and whether it is able to maintain its high accuracy when using the medical journal datasets.

III. METHODOLOGY

Figure 1 shows the research framework of this study. The research framework for this project consists of four phases. The first phase is the literature review of the techniques that had been used in the previous research and the identification of problems and solutions. This had been discussed in chapter 2. The next phase is data collection and data preprocessing to provide clean datasets for this project. The next phase will mainly focus on building the CNN model. The last phase is to analyse and compare the results and perform evaluations on the models to test their performance and accuracy. The objective 1 of the research will achieve via phase 2; objective 2 will be achieved via phase 3 and objective 3 of this research will be achieved via phase 4.

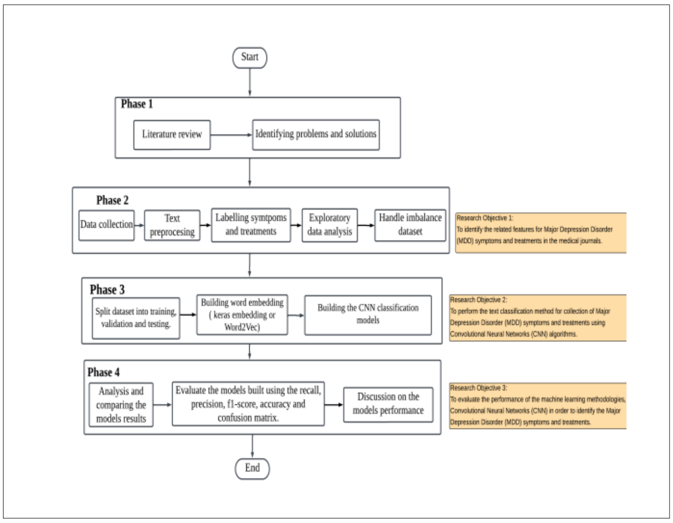
Start
Phase 1
Literature review
Identifying problems and solutions
Phase 2
Data collection
Text preprocesing
Labelling symtpoms and treatments
Exploratory data analysis
Handle imbalance dataset
Research Objective 1:
To identify the related features for Major Depression Disorder (MDD) symptoms and treatments in the medical journals.
Phase 3
Split dataset into training, validation and testing.
Building word embedding (keras embedding or Word2Vec)
Building the CNN classification models
Research Objective 2:
To perform the text classification method for collection of Major Depression Disorder (MDD) symptoms and treatments using Convolutional Neural Networks (CNN) algorithms.
Phase 4
Analysis and comparing the models results
Evaluate the models built using the recall, precision, f1-score, accuracy and confusion matrix.
Discussion on the models performance
Research Objective 3:
To evaluate the performance of the machine learning methodologies, Convolutional Neural Networks (CNN) in order to identify the Major Depression Disorder (MDD) symptoms and treatments.
End

Fig. 1. The flow of research framework

A. *Dataset Preparation*

There were around 5000 medical journals retrieved from the NCBI website by using Entrez, Python and Biopython to access the NCBI's databases. The abstract of the articles was extracted and split into sentences by using the sent_tokenize function in Python. The result is then saved into a CSV file to be used as research's datasets. After that, text preprocessing will be performed to clean the text data. The text preprocessing process includes the removal of stop words, digits, punctuation, and empty rows; converting text to lowercase; tokenization, and lemmatization. Empty rows in the datasets after text preprocessing are also checked again to ensure there will be no null values after the text preprocessing process

Before labelling the dataset, the label of MDD symptoms and treatments need to be validated first. The keywords derived in Table 3.1 are first listed into only two columns, symptoms and treatments. Later, these keywords are reviewed and validated. Firstly, the keywords that consist of using the stop words will be revised into two or more sentences. For example, 'Feelings of guilt, worthlessness, or helplessness' found in VeryWell Health website for symptoms keywords, it will be separate into three sentences, which are 'Feelings of guilt', 'Feelings of worthlessness', 'Feelings of helplessness' .This process is done manually to ensure there is no loss of semantic meaning of keywords. Next, the keywords that consist of duplicate words will be removed. For example, insomnia can be found in both the DSM-5 diagnostic manual and Mayo Clinic website for the symptoms, one of it will be removed. The keywords that consist of the same semantic meaning but using different word representation will still be used in the keyword list. For example, decreased concentration is found in the World Health Organization (WHO) and poor concentration is found in the DSM-5 diagnostic manual, where both have the same semantic meaning, the MDD patient's concentration level reduced. Besides that, the treatments keyword that have abbreviation will be separated into two words, such as selective serotonin reuptake inhibitors (SSRIs) will be separated into two words, 'S\selective serotonin reuptake inhibitors ' and 'SSRIs'. This is because in the medical journals, abbreviation will also be used to represent the word for MDD treatments. There are 48 keywords defined for symptoms and 19 keywords for treatments.

The keywords will undergo text preprocessing, which includes, conversion to lower cases, tokenization, removal of stop words, and lemmatization. The sentences are labelled based on matching the keywords defined. The ngram is imported from NLTK to assist in searching the keywords in the sentences. This is because certain keywords may have more than 1 gram of words. Checking the words in the sentences grams by grams will help increase the accuracy of the labelling process. There will be 4 labels in this dataset. If there are no any keywords found in the sentences, it will be labelled as '0', which represents 'None'. Else if there are any symptom keywords found in the sentences, it will be labelled as '1', which represents 'Symptoms'. Else if there are any treatments keywords found in the sentences, it will be labelled as '2', which represents 'Treatments'. Finally, if the symptoms and treatments keyword are found in the sentences, it will be labelled as '3', which represents 'Both'. The dictionary will also be defined for symptoms and treatments keywords to store the matched word found to assist in plotting the word cloud.

Next, exploratory data analysis will be done to gain a better understanding of it and to handle imbalanced data. A histogram is also plotted to visualise the number of word counts for each sentence of the abstracts. This is to understand the word length of the sentences and to determine a suitable length that can be used to pad the input sequences that need to feed the model. Value of 100 was chosen for the value to pad the sequences as the length of words in this dataset is mostly less than 100 words.

Issue of data imbalance also occurred, where the data for 'None' and 'Both' labels are unbalanced in the dataset. The 'None' category value is using an undersampling technique, randomly deleting the data until it has the number of 5000. The 'None' category is modified to avoid the model more focus on it. Meanwhile, the 'Both' category will be removed as it has the smallest values and it does not provide significant impact to the models to achieve the research's objectives.

B. *Model Building*

Firstly, the labelling dataset is loaded from a CSV file into a Pandas DataFrame. The dataset is then divided into 'X' and 'y' variables, where 'X' contains the text sentences while 'y' represents the classification labels. Then, the 'X' dataset is tokenized using the 'Tokenizer' object which will convert the text into sequences of integers. The sequence length is padded into 100 tokens to ensure the uniformity of the sequence length. The dataset, 'X' and 'y' is then split into 3 sets of training, validation and testing sets. The first set: 80.00% training, 10.00% testing, 10.00% validation. The second set: 70.00% training, 15.00% testing, 15.00% validation. The third set: 60.00% training, 20.00% testing, 20.00% validation.

Besides that, before building the model, there are some parameters and functions which will be used by the three models built later. These include the number of batch size is set to 32 ,number of epochs is set to 15, the optimizer chosen is Adam. The EarlyStopping function will also be used to stop the training process when the model performance does not improve in the validation dataset. The function is set to stop the training process if there is no improvement after 4 consecutive epochs.

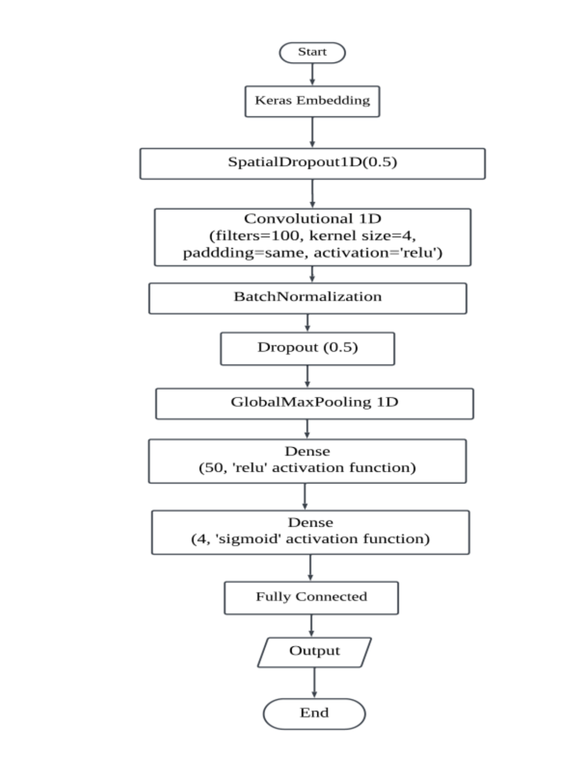

Fig. 2. The flow building benchmark CNN model

Figure 2 shows the flow of the building benchmark CNN model. The benchmark model is proposed by Mohammed *et al.* (2020). The model is built with the start of a sequential model. The sequence layer will accept a stack of layers of the model. Next, the embedding layer is created using Keras. The embedding layer is the input layer and it accepts three parameters: input dimension, input length, and output dimension. The value for input and output dimension is set to 100, which is the maximum features found in the dataset and embedding dimension. The value for input length is the max length of the sentences, which is also 100. After that, SpatialDroupout1D layer is added with a value of 0.5, Later, Convolutional1D layer is added with the filter is set to 100, kernel size with value 4, padding is set to “same”, and ‘ReLU’ is chosen for the activation function. The layers are then followed by BatchNormalization, Dropout layer with value of 0.5, GlobalMaxPool1D, two Dense layers with 50 and 4 number of neurons respectively. After model layers are added, the model is compiled. The loss function chosen is Sparse Categorical Cross Entropy to suit the multilabel text classification. The optimizer chosen is ‘Adam with the learning rate of 0.01, and metrics chosen is accuracy. The model is then fit to train it with the training data and validate with the validation data. The process of building the benchmark model is repeated using the three sets of different splitting ratio datasets.

The flow of building the proposed CNN model is similar to the benchmark model. It is modified based on the benchmark model. All the layers and parameters are the same except there is L2 Regularization applied to the last two Dense layers. Furthermore, the learning rate is set to 0.001 rather than 0.01, same as the benchmark paper. The process of building the proposed model is repeated using the three sets of different splitting ratio datasets.

The third model of the CNN is built using the proposed method but using the Word2vec embedding layer, instead of Keras embedding layer. Firstly, the Word2Vec model is initially defined by tokenizing the sequences back to text and splits them into individual words in the training dataset. Later, the Word2Vec model is trained on these sentences. The vector size is set to 100, window is set to 5, workers is set to 4, and the minimum count is set 1. The vocabulary size of the model built is 99. After that, the embedding matrix was built based on the Word2Vec model. The number of rows for the embedding matrix is the vocabulary size plus one, which is 100. The number of columns of the embedding matrix is set to be 100.

After the embedding matrix is ready, it is fed to the proposed CNN model. In the embedding layer, the input and output dimension are set to the embedding matrix size, which is 100 respectively. The weight of the embedding layer is initialised with the pre-trained Word2vec model built using the training dataset. The weights are allowed to be updated to allow the backpropagation to adjust the weight to fit the task. The process of building the proposed model with Word2Vec embedding is repeated using the three sets of different splitting ratio datasets.

C. *Performance Evaluation*

In phase 4, the results for each CNN model will be recorded and analysed. The comparison will be made based on the evaluation metrics such as accuracy, precision, f1-score, and recall. These metrics will be used to test the performance of the CNN models. Confusion matrix will be plotted to visualise the classification result of the models. A discussion and comparison will be made based on the results for each model. The confusion matrix consists of true positive (TP), false positive (FP), true negative (TN), and false negative (FN). Equation 1, equation 2, equation 3 and equation 4 show the accuracy formula, precision formula, recall formula, and F1-Score formula respectively.

$$\text{Accuracy} = \frac{(\text{TP+TN})}{(\text{TP+FP+FN+TN})} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP+FP})} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP+FN})} \tag{3}$$

$$\text{F1} - \text{score} = 2 \times \frac{\text{precision} \times \text{recall}}{(\text{precision+recall})} \tag{4}$$

IV. Results

TABLE I. Set 1 Dataset Ratio Results

Models	**Accuracy**	**None (Precision, Recall, F1-Score)**	**Symptoms (Precision, Recall, F1-Score)**	**Treatments (Precision, Recall, F1-Score)**
Benchmark CNN Model	0.91	0.87 0.98 0.92	0.97 0.56 0.71	0.97 0.96 0.97
Proposed CNN Model	0.92	0.88 0.97 0.92	0.91 0.63 0.74	0.98 0.96 0.97
Proposed CNN Model + Word2vec	0.92	0.88 0.98 0.92	0.93 0.61 0.74	0.98 0.96 0.97

TABLE II. Set 2 Dataset Ratio Results

Models	**Accuracy**	**None (Precision, Recall, F1-Score)**	**Symptoms (Precision, Recall, F1-Score)**	**Treatments (Precision, Recall, F1-Score)**
Benchmark CNN Model	0.93	0.88 0.99 0.93	0.98 0.65 0.78	1.00 0.95 0.97
Proposed CNN Model	0.92	0.89 0.96 0.92	0.93 0.67 0.78	0.96 0.97 0.97
Proposed CNN Model + Word2vec	0.92	0.88 0.97 0.93	0.94 0.66 0.77	0.98 0.97 0.97

TABLE III. Set 3 Dataset Ratio Results

Models	**Accuracy**	**None (Precision, Recall, F1-Score)**	**Symptoms (Precision, Recall, F1-Score)**	**Treatments (Precision, Recall, F1-Score)**
Benchmark CNN Model	0.88	0.87 0.91 0.89	0.74 0.65 0.69	0.97 0.96 0.96
Proposed CNN Model	0.91	0.88 0.95 0.92	0.89 0.68 0.77	0.97 0.96 0.96
Proposed CNN Model + Word2vec	0.92	0.88 0.97 0.92	0.95 0.66 0.78	0.97 0.96 0.96

The results for each model for the three sets of models were recorded. The accuracy, precision, recall, and F1-score of three models for each dataset's splitting ratio are recorded and tabulated. The results for set 1 dataset splitting ratio is shown in Table I, the result for set 2 is shown in Table II and result for set 3 is shown in Table III.

Based on the result tabulated for each set, it shows that the overall accuracy result of each model for set 1 and set 2 are higher than set 3. The range of the accuracy for set 1 is from 0.91 to 0.92 , set 2 is from 0.92 to 0.93, whereas set 3 is from 0.88 to 0.92. Recall in set 2 for symptoms has an average 6.00% higher values across three models compared to set 1. The F1-score in set 2 for symptoms has an average 5.00% higher values across three models compared to set 1. For the none and treatments labels, both set 1 and set 2 have the values of metrics higher than 80.00%.

From Table II, the benchmark CNN model is able to achieve 93.00% of accuracy while the accuracy of the proposed model is reduced by 1.00% but the precision value is decreased to 96.00%. Furthermore, the proposed model with and without Word2vec achieves the accuracy of 92.00%. The differences of precision, recall, and F1-Score between these two models are around 1.00% across the none and symptoms labels. For the treatments label, the precision of the proposed model with Word2Vec (98.00%) is 2.00% higher than the proposed model (96.00%). The recall and F1-Score for both models are the same, 97.00%.

V. Discussion

From the result of the three different sets of dataset splitting ratio, it shows that the overall accuracy result of each model for set 1 and set 2 are higher than set 3. The range of the accuracy for set 1 is from 0.91 to 0.92 , set 2 is from 0.92 to 0.93, whereas set 3 is from 0.88 to 0.92. In between set 1 and set 2, set 2 is chosen. This is because the recall and F1-score of the symptoms label are higher than set 1 for all three models. Recall in set 2 for symptoms has an average 6.00% higher values across three models compared to set 1. The F1-score in set 2 for symptoms has an average 5.00% higher values across three models compared to set 1. For the none and treatments labels, both set 1 and set 2 have the values of metrics higher than 80.00%. Generally, set 2 is chosen to be used to compare the results between models in further discussion. This means that the models can make more accurate predictions in true or negative instances when using set 2 dataset splitting ratio.

The benchmark CNN model shows model overfitting occurs as it has 100.00% precision value for the treatments label. This is because the dataset used in the benchmark paper is significantly larger than the dataset in this research, which is around 20,000. The model built in the benchmark paper will suit for its dataset but not in this research.

The use of L2 Regularization in the last two dense layers helps to reduce the overfitting. Furthermore, the learning rate for the Adam optimizer is decreased to 0.001 to assist in model generalization and reduce overfitting. Even though the accuracy of the proposed model dropped by 1.00%, it is able to generalize better than the benchmark model.

Besides that, to improve the model performance, Word2Vec embedding matrix is used in the proposed model. However, there is no significant improvement after changing the embedding layer. The accuracy of the model using Word2Vec is the same as the proposed model, 92.00%. The overall difference values for the recall, precision, F1-Score for the none and symptoms labels between these two models is 1.00%. The precision for the treatments label increases by 2.00% but the recall and F1-Score are the same with the proposed model, 97.00%. This is mainly because the vocabulary size of the dataset used by this research is relatively small, which is only 99. This causes the word2vec embedding matrix generated is unable to capture meaningful words. When the proposed CNN model is trained with this embedding layer, the model performance is unable to improve well.

Previous studies done by Adewumi *et al.* (2022) also show that the use of the smallest dataset, such as Wiki Abstract also show poorer results in the Word2vec in downstream tasks, such as sentiment analysis [17]. In this research, our dataset is only 1.35 MB, which is significantly smaller than the dataset used by Adewumi *et al.* (2022), which is 15 MB. As a result, it can be said that the proposed CNN model is the best model that classifies the MDD symptoms and treatments using the medical journal articles.

VI. Conclusion

In conclusion, the result from using three different dataset split ratios shows that all models outperformed when using the set 2 dataset. The comparison between the three CNN models shows that the proposed CNN model with set 2 dataset split ratio has the best overall performance, with an accuracy of 92.00%, without overfitting. The benchmark model proposed by the author, Mohammed *et al.* (2020), is not suitable to be used in this research as model overfitting occurred. This is because the size and context of the dataset used in benchmark paper is different from this research. The benchmark paper utilised a significantly large dataset, approximately 20,000 size and its content are toxic comments from Wikipedia talk pages. These differences may affect the model performance as the CNN model built in benchmark paper may only suit for its own dataset only and is not suitable to use in this research.

Furthermore, the use of Word2vec does not show any significant impact on the model performance. This is because the dataset is relatively small and produces a relatively small vocabulary size which builds a less meaningful embedding matrix to train the model. This concludes that the proposed CNN model is the best model to classify MDD symptoms and treatments in this research.

Besides that, the research objectives are achieved, including identifying the related features for MDD symptoms and treatments in the medical journals, performing text classification method for collection of Major Depression Disorder (MDD) symptoms and treatments using Convolutional Neural Networks (CNN) algorithms, and evaluating the performance of the machine learning methodologies, CNN in order to identify the Major Depression Disorder (MDD) symptoms and treatments. Other than that, there are also future work can be done which include: identify solutions to increase the corpus of keywords of MDD symptoms and treatments to provide a more precise labelling process; implement other methods to handle imbalanced datasets rather than undersampling techniques; Lastly, experiment with different embedding layers other than Keras and Word2vec to enhance the performance of the models such as Glove or FastText.

References

[1] Li, Z., Ruan, M., Chen, J., and Fang, Y. (2021). Major Depressive Disorder: Advances in Neuroscience Research and Translational Applications. *Neuroscience Bulletin,* 37(6), 863-880.

[2] Soni, S., Chouhan, S. S., and Rathore, S. S. (2022). TextConvoNet: a convolutional neural network based architecture for text classification. *Appl Intell (Dordr),* 1-20.

[3] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J. (2021). Deep Learning--based Text Classification: A Comprehensive Review. *ACM Comput. Surv., 54*(3), Article 62.

[4] Cousins, A., Nakano, L., Schofield, E., and Kabaila, R. (2022). A neural network approach to optimising treatments for depression using data from specialist and community psychiatric services in Australia, New Zealand and Japan. *Neural Comput Appl,* pp. 1-20.

[5] Kim, N. H., Kim, J. M., Park, D. M., Ji, S. R., and Kim, J. W. (2022). Analysis of depression in social media texts through the Patient Health Questionnaire-9 and natural language processing. *Digit Health,* p. 8.

[6] Kadhim, A. I. (2019). Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review,* 52(1), pp. 273-292.

[7] Dogra, V., Verma, S., Kavita, Chatterjee, P., Shafi, J., Choi, J. and Ijaz, M. F. (2022). A Complete Process of Text Classification System Using State-of-the-Art NLP Models. *Comput Intell Neurosci,* p. 1883698.

[8] Nadeem, A., Naveed, M., Islam Satti, M., Afzal, H., Ahmad, T. and Kim, K. I. (2022). Depression Detection Based on Hybrid Deep Learning SSCL Framework Using Self-Attention Mechanism: An Application to Social Networking Data. *Sensors (Basel),* 22(24).

[9] Wang, C., Nulty, P. and Lillis, D. (2021). A Comparative Study on Word Embeddings in Deep Learning for Text Classification. *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval,* Seoul, Republic of Korea: Association for Computing Machinery.

[10] Li, S. and Gong, B. (2021). Word embedding and text classification based on deep learning methods. *MATEC Web Conf.,* 336**,** p. 06022.

[11] Mohammed, H. H., Dogdu, E., Gorur, A. K. and Choupani, R. (2020). Multi-Label Classification of Text Documents Using Deep Learning. *2020 IEEE International Conference on Big Data (Big Data*, pp. 4681-4689.

[12] Dharma, E. M., Gaol, F. L., Warnars, H. and Soewito, B. (2022). The accuracy comparison among Word2vec, Glove, and Fasttext towards convolution neural network (CNN) text classification. *Journal of Theoretical and Applied Information Technology,* 100(2)**,** p. 31.

[13] Dong, S., Wang, P. and Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review,* 40, p. 00379.

[14] Yang, Z. and Emmert-Streib, F. (2024). Optimal performance of Binary Relevance CNN in targeted multi-label text classification. *Knowledge-Based Systems,* 284**,** p. 111286.

[15] Liu, W., Pang, J., Li, N., Zhou, X. and Yue, F. (2021). Research on Multi-label text Classification Method Based on taLBERT-CNN. *International Journal of Computational Intelligence Systems,* 14, p. 201.

[16] Elnagar, A., Al-debsi, R. and Einea, O. (2020). Arabic text classification using deep learning models. *Information Processing & Management,* 57(1)**,** p. 102121.

[17] Adewumi, T., Liwicki, F. and Liwicki, M. (2022). Word2Vec: Optimal hyperparameters and their impact on natural language processing downstream tasks. *Open Computer Science,* 12**,** pp. 134-141.