Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.040202

ARTICLE

Text Augmentation-Based Model for Emotion Recognition Using Transformers

Fida Mohammad[1,*], Mukhtaj Khan[1], Safdar Nawaz Khan Marwat[2], Naveed Jan[3], Neelam Gohar[4], Muhammad Bilal[3] and Amal Al-Rasheed[5]

[1]Department of Computer Science, The University of Haripur, Haripur, 22620, Pakistan

[2]Department of Computer Systems Engineering, Faculty of Electrical and Computer Engineering, University of Engineering and Technology Peshawar, Peshawar, 25120, Pakistan

[3]Department of Electronics Engineering Technology, University of Technology, Nowshera, 24100, Pakistan

[4]Department of Computer Science, Shaheed Benazir Bhutto Women University, Peshawar, 25000, Pakistan

[5]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, 11671, Saudi Arabia

*Corresponding Author: Fida Mohammad. Email: fidamuhammad120@gmail.com

Received: 08 March 2023 Accepted: 04 July 2023 Published: 08 October 2023

ABSTRACT

Emotion Recognition in Conversations (ERC) is fundamental in creating emotionally intelligent machines. Graph-Based Network (GBN) models have gained popularity in detecting conversational contexts for ERC tasks. However, their limited ability to collect and acquire contextual information hinders their effectiveness. We propose a Text Augmentation-based computational model for recognizing emotions using transformers (TA-MERT) to address this. The proposed model uses the Multimodal Emotion Lines Dataset (MELD), which ensures a balanced representation for recognizing human emotions. The model used text augmentation techniques to produce more training data, improving the proposed model's accuracy. Transformer encoders train the deep neural network (DNN) model, especially Bidirectional Encoder (BE) representations that capture both forward and backward contextual information. This integration improves the accuracy and robustness of the proposed model. Furthermore, we present a method for balancing the training dataset by creating enhanced samples from the original dataset. By balancing the dataset across all emotion categories, we can lessen the adverse effects of data imbalance on the accuracy of the proposed model. Experimental results on the MELD dataset show that TA-MERT outperforms earlier methods, achieving a weighted F1 score of 62.60% and an accuracy of 64.36%. Overall, the proposed TA-MERT model solves the GBN models' weaknesses in obtaining contextual data for ERC. TA-MERT model recognizes human emotions more accurately by employing text augmentation and transformer-based encoding. The balanced dataset and the additional training samples also enhance its resilience. These findings highlight the significance of transformer-based approaches for special emotion recognition in conversations.

KEYWORDS

Emotion recognition in conversation; graph-based network; text augmentation-based model; multimodal emotion lines dataset; bidirectional encoder representation for transformer

1 Introduction

Emotion Recognition in Conversation (ERC) is a subfield of Emotion Recognition (ER) that focuses on extracting human emotions from dialogues or discussions involving two or more interlocutors. Emotion understanding is crucial to developing humanoid Artificial Intelligence (AI) systems. With the widespread availability of conversational data on platforms like LinkedIn, Twitter, Reddit, YouTube, Facebook, and E-commerce sites, the research focus has shifted towards emotion detection and recognition in conversations using Natural Language Processing (NLP) techniques. Emotion recognition in text-based conversations has gained significant attention due to its potential for sentiment analysis and opinion mining in openly accessible conversational data. While the domains of speech and facial emotion recognition have made notable advancements, text-based emotion identification remains an area that requires further exploration and research [1]. Recognizing and understanding human emotions conveyed through Text is becoming increasingly important in computational linguistics, given its practical implications and applications [2].

Our study addresses explicitly recognizing human emotions in conversations using the MELD dataset (Multimodal et al. [3]). The MELD dataset is a multimodal resource incorporating Text, audio, and visual modalities. However, for our proposed model, we focus exclusively on the textual component of the dataset. By leveraging the rich textual information in conversations, we aim to advance the field of text-based emotion recognition.

A transformer-based approach transforms the text-based input into a numerical representation suitable for machine learning and deep learning models. Specifically, we utilize the Bi-directional Encoder Representations from Transformers (BERT) [4] model, a pre-trained language model capable of encoding the contextual information of a text. BERT converts input utterances into input IDs and attention mask vectors, which work as inputs to our deep neural network (DNN) model for training.

Emotion recognition in textual conversations has gained considerable attention in recent years due to its wide range of applications across various domains, such as opinion mining, healthcare, psychology, robotics, human-computer interaction, and IoT-based systems. The ability to accurately detect and understand human emotions from Text plays a crucial role in enhancing communication, personalization, and user experience. However, existing approaches face several challenges that hinder their accuracy and effectiveness.

One significant challenge in emotion recognition is the presence of imbalanced datasets used for training and evaluation. Imbalanced datasets, where the distribution of emotion classes is uneven, can lead to biased models that perform well on majority classes but struggle with minority classes. This limitation affects the overall performance and generalizability of the models. Additionally, many previous studies in this field have relied on non-contextual word embeddings, which may limit their ability to capture the nuanced meanings and contextual information in textual conversations.

Moreover, despite the advancements in deep learning, deep neural networks (DNNs) have yet to be extensively explored for text-based emotion recognition. DNNs have demonstrated remarkable capabilities in capturing complex patterns and relationships in data, making them suitable for modeling the intricate nature of human emotions. By leveraging the power of DNNs, there is an opportunity to improve the accuracy and performance of emotion recognition models in textual conversations.

To address these limitations, we propose a novel text augmentation-based model for emotion recognition using transformers. Our research aims to enhance the accuracy and effectiveness of emotion recognition models by addressing the challenges of dataset imbalance and leveraging the capabilities of contextualized word embeddings and deep neural networks. By incorporating text augmentation techniques and balancing the dataset, we aim to mitigate the impact of class imbalance and enable more accurate modeling of emotions in textual conversations.

The transformer-based approach, specifically the Bi-directional Encoder Representations from Transformers (BERT), captures the contextual information in textual data. BERT provides a robust framework for representing words in their surrounding context, allowing for a more nuanced and accurate understanding of a text. Furthermore, by employing deep neural networks for emotion classification, we can leverage their ability to capture complex relationships and patterns in the data, leading to improved emotion recognition performance.

This paper presents the details of our proposed text augmentation-based model for emotion recognition. We evaluate the performance of our model using various performance metrics, including accuracy, weighted F1 score, recall, and precision, to assess its effectiveness compared to existing approaches. We demonstrate our model's superior performance and improved accuracy through extensive experiments and analysis, highlighting its novelty and advancement in text-based emotion recognition. Fig. 1 shows the emotion recognition in conversation example from the MELD dataset.

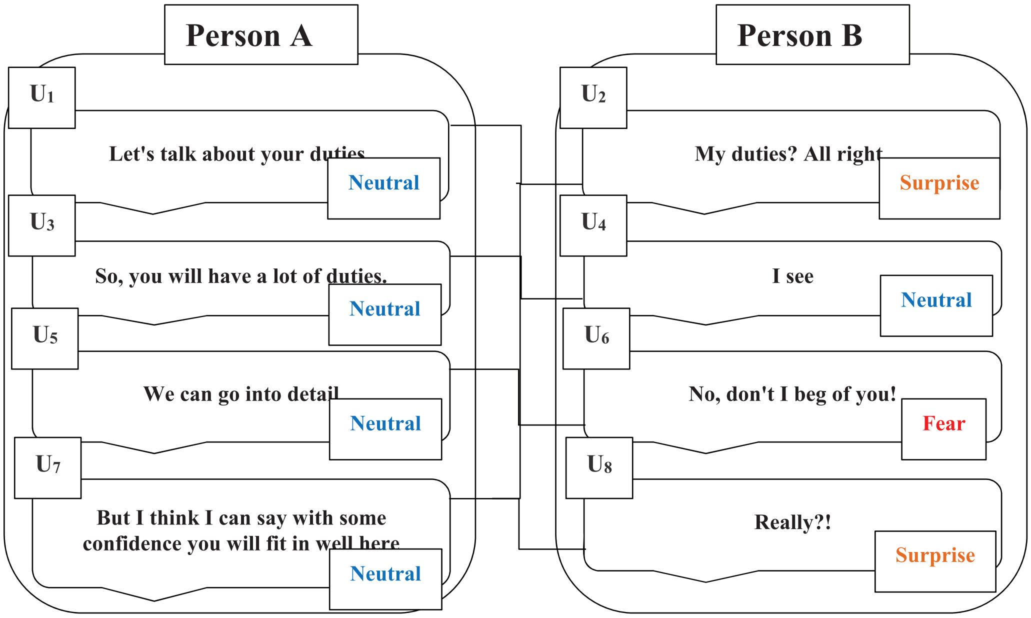


Figure 1: Emotion recognition in conversation example from MELD dataset

2 Literature Review

In this section, we critically examine several methods or models of emotion recognition in conversation, word embeddings, and text data augmentation for emotion classification. In the relevant literature, several models have been proposed to recognize or classify human emotions in conversation, and they achieved different accuracy and weighted F1-score.

Choi et al. [5] proposed a Residual-based Graph Convolutional Network (RGCN) computational model for predicting human emotions in conversion—the proposed model employed intra-utterance feature extraction on ResNet. Inter-utterance feature extraction on Graph Convolutional Network

(GCN) is used to extract features. The author proposed a new loss method. The proposed model applied pre-trained Global Vector (GloVe) word embeddings with 300 dimensions, using an Adam optimizer for optimization. The proposed model is compared with bidirectional contextual long short-term memory (bc-LSTM), Dialogue Graph Convolutional Network (DialogueGCN), Dialogue Recurrent Neural Network (DialogueRNN), and Knowledge Enriched Transformer (KET). The experimental outcome yielded that the RGCN model accomplished a weighted F1 score of 55.98% on the MELD dataset.

Ghosal et al. [6] proposed a DialogueGCN computational model to classify human emotions in a conversation using the benchmark dataset. The proposed model employed two encoders, i.e., sequential context and speaker-level encoders, to encode the input features. To optimize hyperparameters, the proposed model applied pre-trained GloVe word embeddings with 300 dimensions and grid search. The proposed model acquired a weighted F1 score of 58.10% and 59.46% accuracy on the MELD dataset, according to the results of their experiments.

Majumder et al. [7] suggested a DialogueRNN computation model which detects human emotion in conversation. The proposed method employed a CNN to extract the textual features. The Dialogue RNN network gets trained at utterance with emotion labels. The author used Adam optimizer to train the network and grid search for hyperparameters optimization. The proposed model achieved a weighted F1 score of 57.03% and 59.54% accuracy.

Hu et al. [8] proposed a Dialogue CRN computational model to recognize human emotion in conversation. The proposed model applied pre-trained GloVe word embedding with 300 dimensions. Dialogue CRN model included three phases. In the first phase, the model generated the context representation of utterance at the situation and speaker level. In the second phase, it developed multiple turns for reasoning to collect and incorporate emotional information iteratively. The last phase, emotion recognition, predicts or classifies the emotion. A weighted F1 score was utilized on the MELD dataset to evaluate model performance. The suggested DialogueCRN model had a weighted F1 score of 58.39% and 60.73% accuracy.

Zhong et al. [9] proposed the KET model. This model uses external knowledge of common sense based on the attention mechanism. The external knowledge works on the emotion lexicon [10] and ConceptNet [11]. GloVe word embedding with 300 dimensions vectorizes the textual data into a numeric representation. Adam optimizer is helpful for network optimization. The proposed model gains a 58.18% weighted F1 score on the MELD dataset.

Xing et al. [12] proposed Adapted Dynamics Memory Network (ADMN) computational model to predict human emotion in conversation. ADMN model used a global recurrent neural network (RNN) to get inter-speaker impact. The proposed method separately modeled two things, i.e., self and inter-speaker impact, and produced learning for the current utterance. MELD dataset is a multimodality dataset, i.e., it contains textual, audio, and visual data. The A-DMN model trained on textual, visual, and audio data; therefore, they used a convolutional neural network (CNN) for the extraction of textual features, Open Smile (OS) for the extraction of audio features, and a 3D-CNN for the extraction of visual features. The proposed model achieved a 60.45% weighted F1 score on the MELD dataset.

Wang et al. [13] used an LSTM-based encoder to train the model. Features were extracted using a CNN. The proposed model used GloVe word embeddings with 300 dimensions and an Adam optimizer for model optimization. The model achieved a weighted F1 score of 58.36% on the MELD dataset.

Yeh et al. [14] introduced a dialogical emotion decoding (DED) method, which analyses a dialogue as a series and sequentially decodes the emotion states of each phrase over time using a particular recognition engine. Combining emotional effects from intra and inter-speakers in a conversation trains the decoder. For decoding, the proposed method used CNN and RNN. The model used a Distance-Dependent Chinese Restaurant Procedure (DDCRP) for emotion assignment. It is a clustering method used in image segmentation, text modeling, and speaker identification. The proposed method achieved 43.6% weighted accuracy on the MELD dataset and 69.0% on the Interactive Emotional Dyadic Motion capture (IEMOCAP) dataset.

Jiao et al. [15] introduced Attention Gated Hierarchical Memory Network (AGHMN) method for emotion categorization. Conversational real-time emotion recognition is vital to construct an expressively intelligent method. The model was tested, trained, and a detailed analysis on two emotion discussion datasets, MELD and IEMOCAP. The convolutional neural network assists in feature extraction. Firstly, text data was converted into numerical vector form, and word2Vec word embeddings were applied. Secondly, the numerical form of the data is passed to the decoder, a hierarchical memory network with two levels, a lower one and an upper one. The lower level is the utterance reader, and the upper one is the fusion layer. Thirdly, attention weight is calculated, and the output is stored in the memory bank. Finally, a SoftMax activation function is applied for the classification of emotions. Adam optimizer is used for model optimization. On the IEMOCAP dataset, the proposed model received a 63.5% weighted F1 score, unlike the MELD dataset, which received a 58.1% weighted F1 score.

Zhang et al. [16] developed the ConGCN model, focusing on emotion recognition in multi-speaker discussions rather than two regular speaker chats. The author proposed a conversational graph-based CNN. In this method, each utterance and speaker are represented as nodes. Speaker-sensitive dependence is represented by an undirected edge between an utterance node and its speaker node. In contrast, context-sensitive reliance is represented by an undirected edge between two utterance nodes from the same dialogue. The entire corpus of conversational data is portrayed as a sizable composite network, and the emotion detection problem is rephrased as a problem of classifying the nodes in the graph. Firstly, to generate input, the proposed model used GloVe word embeddings with 300 dimensions, and an Adam optimizer was used to train the model. Secondly, the convolutional layer contains three filters, and then a max pooling layer is applied to find the pooled features. Finally, the pooled features are passed to the bidirectional long-short-term-memory layer to classify the emotions. The proposed model achieved a 59.4% weighted F1 score using MELD multimodal (Text and audio) and a 57.4% weighted F1 score on the MELD unimodal.

Sheng et al. [17] developed SumAggGIN, a two-stage network that integrates local dependence reasoning over nearby utterances with topic-related emotional expression inference in a global-to-local manner. The SumAggGIN model used GloVe word embeddings with 300 dimensions and an Adam optimizer for model optimization. SumAggGIN employed three sizes of convolution filters: three, four, and five, each with 50 feature maps. Then, the output of the convolution layer is passed to the max-pooling layer and ReLU activation function. Finally, these activation results are combined and input into a 150-dimensional layer for classification. For local feature extraction, bidirectional LSTM is applied. SumAggGIN achieved a 58.45% weighted F1 score on the MELD dataset.

For emotion classification in conversation, an iterative emotion interaction network was developed by Lu et al. [18]. This network directly represents the emotional interaction between utterances with the help of an iterative improvement technique. The authors used an utterance encoder to collect speech representations and initially predicted all utterance emotions. Then, using an emotion interaction-based context encoder, they combined the initial prediction and the utterances to provide an updated emotion prediction. Finally, they iteratively update the emotions using the iterative improvement method. The proposed model has three main modules; an emotion encoder which encodes the emotions; a bidirectional gated recurrent unit; and an emotion classifier which includes a SoftMax activation function to classify the emotions. The model used GloVe word embeddings with 300 dimensions and obtained a 60.72% weighted F1 score.

Li et al. [19] proposed bidirectional emotion recurrent unit to classify emotions. In the first step, the proposed method converts utterance into a vector, and then the vector is passed to the CNN to extract features. Then the output of CNN is passed to the Max-Pooling layer with the ReLU activation function. Finally, a SoftMax activation function is used to classify the utterance emotion. The proposed method obtained 60.9% accuracy on the MELD dataset (textual modality).

Ishiwatari et al. [20] developed relational position encodings that offer sequential information to Relational Graph Attention Networks (RGAT) that reflect the relational graph structure. As a result, both the speaker dependency and the sequential information may be captured by our RGAT model. The proposed model framework involves three modules of contextual utterance embedding. Fine-tune the BERT model for the speaker dependency module graph-based neural network, and finally, the classifier is used to classify the emotion. The proposed RGAT method obtained a weighted F1 score of 60.91% on the MELD dataset (textual modality).

Hu et al. [21] proposed a Multimodal Dynamic Fusion Network (MM-DFN) to classify emotions in multimodal conversation. The proposed method used multimodalities, i.e., Text, audio, and visual, to predict the emotions label. For textual feature extraction, they applied a bidirectional gated recurrent unit. MM-DFN model achieved 62.49% accuracy and 59.46% weighted F1 score on the MELD dataset using multimodality (Text, audio, and video).

Recognizing human emotion based on textual conversations has several applications in various domains, such as opinion mining, healthcare, psychology, robotics, human-computer interaction, and IoT-based system. Several researchers have proposed different computational models for emotion detection in conversation, achieving good accuracy. However, their datasets were imbalanced, which affected the performance and accuracy of the models. Few researchers have addressed text augmentation in the published literature. Most of the study uses non-contextual word embedding, affecting the previous model's accuracy. To the author's knowledge, the researchers have yet to use deep neural networks for emotion recognition. In this work, we propose a text augmentation-based model for emotion recognition to increase accuracy using a balanced dataset.

The objective of this study is listed below:

- The text augmentation technique is applied to the MELD dataset to improve ERC accuracy.
- The dataset is balanced to overcome the impact of the imbalance dataset.
- Transformer-based BERT model is applied to contextualized word embedding.
- DNN is applied for emotions classification.
- The performance of the suggested model is assessed using various performance metrics.

Based on the objectives mentioned above, the proposed model performance is better than the previously published techniques in terms of accuracy, weighted F1 score, recall, and other evaluation metrics, which define the novelty of the proposed work.

The rest of the paper is organized as follows: Section 3 describes the methodology. Results and Discussion are in Section 4, and Section 5 gives the study's conclusion.

3 The Proposed Model

This section explains the detail of benchmark datasets, text data augmentation methods, word embeddings, transformers, and classification. Fig. 2 illustrates the block diagram of the proposed model. The proposed research begins with a technical assessment of the available computer models for classifying emotions in a conversation. Many researchers have worked to classify human emotions in a conversation by applying machine learning and deep learning techniques to achieve good accuracy. However, there are still possible ways and techniques to achieve better results. Secondly, the proposed study selects a valid MELD dataset focusing on textual data. Thirdly, MELD is an imbalanced dataset; therefore, different text augmentation techniques are applied to resolve the problem. Fourthly, word embedding techniques are applied to convert Text into vectors while keeping the contextual information of a sentence. BERT transformer is used for this purpose. Fifthly, the BERT model for classification is applied to classify emotions. Finally, the evaluation matrices, such as confusion metrics, are calculated to evaluate the proposed model.

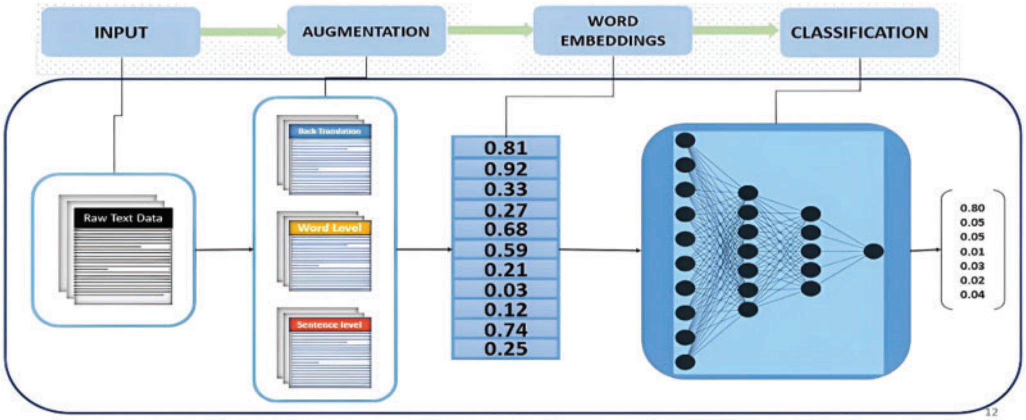


Figure 2: Proposed model block diagram

3.1 Dataset

This study uses MELD [1] as a benchmark dataset, the extended version of the Emotion Line [22] dataset. The MELD dataset is a multimodal dataset that includes Text, video, and auditory data from the friend's television series. The proposed method only considers the textual data. The MELD dataset has 1400 conversations and 13000 utterances. Each utterance in the MELD dataset has been assigned 1 of the seven emotions: fear, surprise, joy, sadness, neutral, disgust, and anger. Fig. 3 shows the distribution of the MELD dataset.

3.2 Text Data Augmentation

In the classification task, the class imbalance problem is one of the key issues which can cause bias, and classifiers are biased toward the majority class [23]. For a long time, the problem of imbalanced data has been a focused area of research, and several solutions have been developed [24]. The MELD dataset class distribution is shown in Fig. 3. The neutral class has 4710 instances, the majority group and the disgusting class has 271 records, a minority group. So, the MELD dataset is imbalanced (contains a class imbalance problem). Data augmentation techniques can be employed to resolve the MELD dataset imbalanced problem. The following data augmentation techniques are used to resolve the imbalanced dataset problem.

- Back Translation Methods,
- Easy Data Augmentation Methods,
- NLPAlbumentation Methods,
- NLPAug Library.

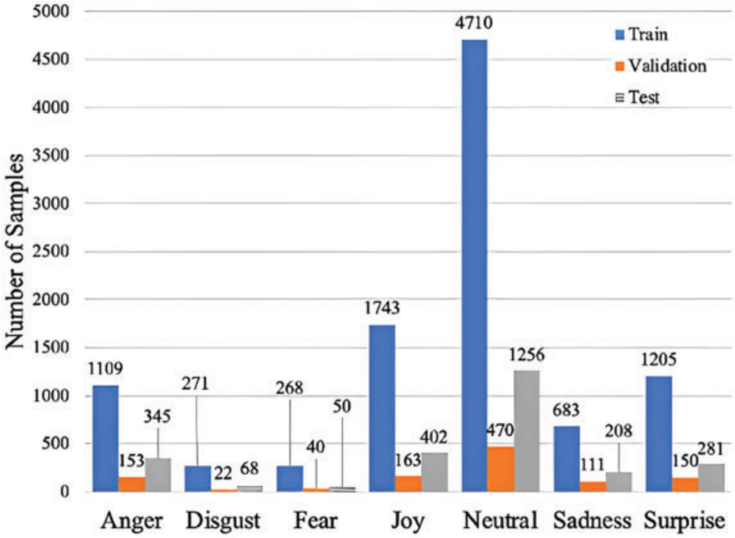


Figure 3: MELD dataset class distribution

To balance the MELD dataset, we substitute a word with its synonym (synonym replacement) through word embeddings to achieve a term with similar sense but dissimilar words in a sentence, replacing words with their synonyms. The synonym substitution is based on the Bert-base-uncased pre-trained contextual word embedding technique to find the synonym for the selected word. The augmented data is defined by Eq. (1). This is done by applying the Data Augmentation function to the original data.

$$x_i' = DataAugmentation(x_i) \tag{1}$$

where the original training data sentences are denoted by the variables x_i, the augmented data x_i' is created by changing synonyms in the original training data sentences. However, the label in the original training data' y_i and the augmented data y_i' label is identical to y_i.

3.3 Transformer and Word Embedding

The transformer technique was introduced by a group of Google researchers [25]. RNNs and CNNs were the most commonly utilized NLP algorithms at that time. The capabilities of the transformer technique are primarily improved due to the application [26] of RNNs and CNNs models because these models do not require data in the sequence's forms. These models have the capabilities to process the data in any order. As the transformer technique processes data in any order, it enables the models to train on larger datasets that were previously inconceivable. Thus, pre-trained models like BERT could train on vast amounts of language data before release.

When classifying text data, the first step is to create a data representation (numerical data) that may be used to train machine learning or deep learning model. Because machines cannot understand natural language or Text in the same way humans can, the text data must be characterized in actual values, with syntactically and semantically equivalent meaning word vectors lying close to each other in vector space and mathematical relationships drawn between them. This method is known as word embeddings. There are two main classes of word embeddings: contextual and non-contextual. Context embedding models may create unique word embeddings for a word that capture its position in a sentence, making them context-dependent. Transformer-based models are commonly used to produce these embeddings. The embeddings are obtained by feeding the entire Text to a model that has already been trained (pre-trained model). It is worth noting that there is a collection of words here, but it does not include contextual embeddings. The word embeddings are created for each word by the other words in a sentence. The context of a sentence refers to those different words. The attention process used by the transformer-based models looks at the relationship between a word's neighbors (surrounding words). Consequently, given a word, embeddings are generated dynamically from a pre-trained or fine-tuned model. Thus, we apply transformer-based contextual embeddings.

Because of the transformer, BERT is better at understanding context and ambiguity in language. Each word in the phrase is examined by all the other words rather than separately by the transformer. Unlike the previous language processing method known as word embedding, earlier models such as word2vec and GloVe would create a vector for each word that only represented one dimension, or a slice, of that word's meaning. BERT uses a masked language modeling technique to stop the word in focus from "seeing itself" or having a fixed meaning independent of its context. The hidden word must then be recognized solely by its context. Instead of having a predetermined identity, their context defines words in BERT. Transformers use the encoder and decoder architecture.

The encoder block employs the self-attention method to provide contextual information from the entire phrase to each token. Any token may have multiple semantics or functions depending on the tokens around it. As a result, the self-attention mechanism uses eight parallel attention heads to allow the model to tap into a variety of embedding subspaces. A linear layer, ReLU, and another linear layer process each embedding vector individually with identical weights in the position-wise feed-forward network (FFN). As a result, the FFN layer transforms each embedding vector, which includes contextual information from multi-head attention. Assuming "x" is an embedding vector, the FFN can be formulated as Eq. (2).

$$\mathrm{f}\left(\mathrm{x}\right) = \mathrm{g}\left(\mathrm{x}\mathrm{w}_1 + \mathrm{b}_1\right)\mathrm{w}_2 + \mathrm{b}_2 \tag{2}$$

where x is an embedding vector, the FFN includes a ReLU activation function g, weights w_1 and w_2, and biased terms b_1 and b_2. w_1 increases the dimensionality of x from 512 to 2048, while w_2 decreases it back to 512. The weights in the FFN are consistent within the same layer. Residual connections, expressed as Eq. (3), are used in the encoder block, employing element-wise additions.

$$\mathrm{x} + \mathrm{sublayer}(\mathrm{x}) \tag{3}$$

The encoder blocks in the transformer utilize multi-head attention and FFN as sublayers. Residual connections pass the prior embeddings to the next layer. The encoder blocks enrich the embedding vectors with information from multi-head self-attention and FFN computations. Each residual connection is followed by layer normalization, as depicted in Eq. (4). Layer normalization operates on each embedding vector individually to mitigate the impact of covariant shift and ensure stable and efficient training.

$$\mathrm{NormalizLayer}(\mathrm{x} + \mathrm{sublayer}(\mathrm{x})) \tag{4}$$

The transformer architecture consists of six stacked encoder blocks. The decoder utilizes the outputs from the final encoder block as input features. These input features are enriched embeddings through multi-head attention, FFN, residual connections, and layer normalization.

3.4 Deep Neural Network

This study applies a deep neural network (DNN) as a classification engine. The deep neural network is a subfield of AI and directly learns features from the data before making decisions. In numerous fields, including speech recognition [27–30], image processing [31–34], natural language processing [35,36], and bioengineering [37], deep learning algorithms have demonstrated that they are the most effective and exceptional machine learning algorithms. Additionally, several studies demonstrated that deep learning algorithms outperform traditional machine learning methods when applied to various complex learning problems [38,39]. Due to its impressive performance in various domains, we used the DNN model as a classifier for emotion recognition in conversation. The DNN model comprises an input, output, and several hidden layers. The input layer is the first layer of the model through which the data is fed to the model. The output layer is the last layer of the model, and it generates the model output. Several hidden layers in the middle are involved in learning the model through the learning procedure. The performance of a DNN model is influenced by the number of hidden layers and other configurations through hyperparameters. In general, a network configured with many hidden layers can produce better performance [40]; however, severe issues like overfitting, computation costs, and model complexity may arise [41].

The DNN model is configured with two input layers (one for kids and another for attention masks), two hidden layers, and one output layer, as illustrated in Fig. 4. Each layer is set up with a different number of neurons. First, the input layer receives the input ids and attention mask vectors produced by the word embeddings. The input from these layers is then passed to the hidden layer, which has 512 neurons and a rectified linear activation unit (ReLU) activation function. The output of the hidden layers is then passed to the output layer, which contains seven neurons with SoftMax activation functions to do multi-class classification. The output layer will generate a probability distribution vector with the probabilities for each emotion class. In the probability distribution vector emotion, a class with the highest probability will be considered a predicted emotion. Adam optimizer is used to optimize the weights of the model. The BERT model is a fine-tuning of the augmented dataset. The DNN can be expressed as:

$$Z_1 = w_1 \times (input\ ids, attention\, mask) \times b_1 \tag{5}$$

$$Z_2 = BERT(w_2 \times Z_1 \times b_2) \tag{6}$$

$$Z_3 = relu(w_3 \times Z_2 \times b_3) \tag{7}$$

$$Z_4 = SoftMax(Z_3) \tag{8}$$

where Z_1 represents the first layer output after applying the weighted sum of the word input_ids and attention_mask (attention mask), using the weight matrix w_1 and bias vector b_1, Z_2 represents the output of the second layer, where we apply the BERT model on the weighted sum of Z_1, using the weight matrix w_2 and bias vector b_2. Z_3 represents the intermediate output after applying the rectified linear activation function (relu) to the weighted sum of Z_2, using the second layer's weight matrix w_3 and bias vector b_3. Z_4 represents the final output after applying the softmax activation function to Z_3, which represents the predicted emotion class probabilities.

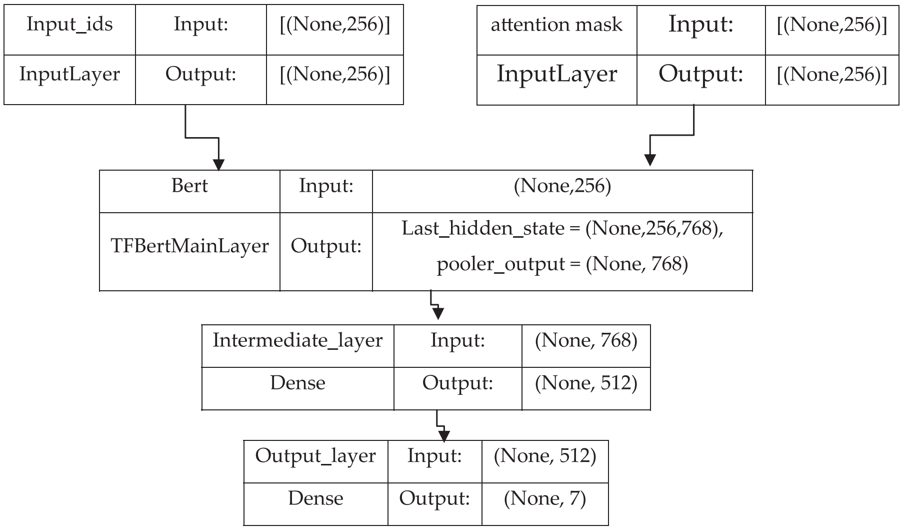


Figure 4: Proposed model architecture

3.5 Performance Evaluation

It is essential to measure how well the classification model predicts the desired outcome when constructing and optimizing it. Before being used in a real-world environment, the performance of a newly developed classifier based on machine learning algorithms can be evaluated using specific processes [42]. We can only know whether a learning model is helpful if it is evaluated. Various types of performance evaluation metrics for evaluating the performance of a machine learning model have been proposed in the literature for evaluating the performance of a machine learning model [43]. In this study, we have considered the following metrics to evaluate the performance of the proposed model.

$$Specificity = \frac{no\ of\ true\ negative}{no\ of\ true\ negative + no\ of\ false\ positive} = \frac{TN}{TN + FP} \tag{9}$$

$$Recall = \frac{no\ of\ true\ positive}{no\ of\ true\ positive + no\ of\ false\ negative} = \frac{TP}{TP + FN} \tag{10}$$

$$Accuracy = \frac{no\ of\ correct\ prediction}{total\ number\ of\ predictions} = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$precision = \frac{True\ Positive}{Total\ predicted\ positive} = \frac{TP}{TP + FP} \tag{12}$$

$$F1\ score = 2 \times \frac{(precision * recall)}{(precision + recall)} \tag{13}$$

In the case of a multi-class problem, the parameters of the confusion matrix can be calculated as True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). The TP value is one in which the actual and predicted values are equal (same) and can be formulated as Eq. (14) where "i" is a reference to each class and "c" is the value in the cell where the number of row and column is equal for example c11.

$$tp_i = c_{ii} \tag{14}$$

The FP value for a class is the total of the values in the corresponding column, except the True Positive value. FP can be formulated as Eq. (15).

$$fp_i = \sum_{k=1}^{n} c_{ki} - tp_i \tag{15}$$

TN is the sum of the column and row values, excluding the class values for which the values are being computed, which makes up the TN value for a given class. TN formulates with the help of Eq. (16).

$$tn_i = \sum_{k=1}^{n} \sum_{j=1}^{n} c_{kj} - tp_i - fp_i - fn_i \tag{16}$$

The FN value for a class is the sum of the values of the related rows, except the true positive value. FN can be formulated as Eq. (17).

$$fn_i = \sum_{k=1}^{n} c_{ik} - tp_i \tag{17}$$

3.6 Experimental Setup

In this section, firstly, the experimental environment is described. Secondly, the methodology to balance the benchmark dataset is presented.

3.6.1 System Specification

Google Colab GPU is used to perform the experiments. We used Keras, TensorFlow, Transformers, Matplotlib, nlpaug, pickle, pandas, NumPy, and sklearn libraries to implement the proposed model.

3.6.2 MELD Dataset and Augmentation

For dataset balancing, one method is to drop some of the samples, called downsampling, and another is to add additional samples to some of the emotion classes that make all the emotion class samples equal in number, called augmentation. Both downsampling and augmentation methods balance the dataset.

1. MELD Training Dataset

After augmentation, the following are the statistics of the training dataset.

- MELD: The original MELD training dataset.
- $\text{MELD}_{\text{N4710}}$: Augmentation concerning the total number of training samples of neutral emotion class which are 4710, shown in Table 1.
- $\text{MELD}_{\text{J1743}}$: Augmentation and downsampling concerning the total no of training samples of joy emotion class which are 1743, shown in Table 1.
- $\text{MELD}_{\text{A1109}}$: Augmentation and downsampling concerning the total no of training samples of anger emotion class which are 1109, shown in Table 1.
- $\text{MELD}_{\text{Avg1427}}$: Augmentation and downsampling concerning the average of all emotion classes training samples, which are 1427, shown in Table 1.

Table 1: Augmentation and downsampling of MELD train dataset

Emotions label	Total number of training samples				
	MELD	$\text{MELD}_{\text{N4710}}$	$\text{MELD}_{\text{J1743}}$	$\text{MELD}_{\text{A1109}}$	$\text{MELD}_{\text{Avg1427}}$
Neutral	4710	4710	1743	1109	1427
Surprise	1205	4710	1743	1109	1427
Fear	268	4710	1743	1109	1427
Sadness	683	4710	1743	1109	1427
Joy	1743	4710	1743	1109	1427
Disgust	271	4710	1743	1109	1427
Anger	1109	4710	1743	1109	1427
Overall	**9989**	**32970**	**12201**	**7763**	**9989**

2. MELD Test and Validation Dataset

After augmentation, the following are the statistics of test and validation datasets:

- MELD_{T}: The original MELD test dataset.
- MELD_{V}: The original MELD validation dataset.
- $\text{MELD}_{\text{TN1256}}$: Augmentation concerning the total number of testing samples of neutral emotion class which are 1256, shown in Table 2.
- $\text{MELD}_{\text{TJ402}}$: Augmentation and downsampling concerning the total no of testing samples of joy emotion class which are 402, shown in Table 2.
- $\text{MELD}_{\text{TS281}}$: Augmentation and downsampling concerning the total no of testing samples of surprise emotion class, which is 281, shown in Table 2.
- $\text{MELD}_{\text{VN470}}$: Augmentation concerning the total number of validation samples of the neutral emotion class, which is 470, shown in Table 3.
- $\text{MELD}_{\text{VJ163}}$: Augmentation and downsampling concerning the total no of validation samples of joy emotion class which are 163, shown in Table 3.
- $\text{MELD}_{\text{VS150}}$: Augmentation and downsampling concerning the total no of validation samples of surprise emotion class, which are 150, shown in Table 3.
- MELD_{V20}: 20 percent of the training dataset.

Table 2: Augmentation and downsampling of the MELD test dataset

Emotions label	Total number of testing samples			
	MELD_{T}	$\text{MELD}_{\text{TN1256}}$	$\text{MELD}_{\text{TJ402}}$	$\text{MELD}_{\text{TS281}}$
Neutral	1256	1256	402	281
Surprise	281	1256	402	281
Fear	50	1256	402	281
Sadness	208	1256	402	281
Joy	402	1256	402	281
Disgust	68	1256	402	281

(Continued)

Table 2 (continued)

Emotions label	Total number of testing samples			
	$\mathrm{MELD}_{\mathrm{T}}$	$\mathrm{MELD}_{\mathrm{TN1256}}$	$\mathrm{MELD}_{\mathrm{TJ402}}$	$\mathrm{MELD}_{\mathrm{TS281}}$
Anger	345	1256	402	281
Overall	**2610**	**8792**	**2814**	**1967**

Table 3: Augmentation and downsampling of the MELD validation dataset

Emotions label	Total number of validation samples			
	$\mathrm{MELD}_{\mathrm{V}}$	$\mathrm{MELD}_{\mathrm{VN470}}$	$\mathrm{MELD}_{\mathrm{VJ163}}$	$\mathrm{MELD}_{\mathrm{VS150}}$
Neutral	470	470	163	150
Surprise	150	470	163	150
Fear	40	470	163	150
Sadness	111	470	163	150
Joy	163	470	163	150
Disgust	22	470	163	150
Anger	153	470	163	150
Overall	**1109**	**3290**	**1141**	**1050**

3.6.3 Configuration Parameters

In Deep learning models, several configuration parameters require configuring. The values of these parameters significantly affect how well a learning model performs. The configuration parameters for the model configuration are listed in Table 4, along with descriptions of each. Unlike the regular parameters, the user specifies configuration parameters during model setup. Deciding what values to specify for a learning model configuration parameter for a given dataset might be challenging. To discover the optimal configuration parameters for the suggested computational model, we tweaked the configuration parameters by altering their values. We created a model for each augmentation, adjusted its configuration parameters, and assessed its output before archiving the data. The set of configuration parameters that produces the best results among all models is chosen as the optimum parameter set for the classifier. We consider the essential parameters during configuration parameter optimization, such as learning rate, batch size, and number of epochs. Table 5 shows the optimal configuration parameters found for the proposed model.

Table 4: Hyperparameters and their explanation

S. No.	Parameter	Explanation
1	Learning rate	The learning rate defines how fast a network updates its weights. If the learning rate is lower, converging will take longer. If the learning rate is high, it will overshoot the minima. Therefore, we train our model on different learning rates and get the optimum one.

(Continued)

Table 4 (continued)

S. No.	Parameter	Explanation
2	No of epochs	One epoch means when all the data is passed through the network, completing both forward and backward passes.
3	Batch size	We divide our complete dataset into batches. Batch size means the total no of training samples appearing in a batch. We choose batch sizes 1, 4, 8, 16, and 32 for our model.
4	Activation function	We add an activation function in our artificial neural network to learn the complex patterns in the data. There are several activation functions. ReLU and SoftMax activation functions are used for the proposed model.
5	Optimizer	Optimizers are used to fine-tune a model's parameters. The primary role of an optimizer is to modify model weights to maximize or minimize the loss function. The loss function is used to measure how well the model performs.
6	Decay	The decay learning rate is a training method for advanced artificial neural networks. It begins by training the network with a high learning rate and gradually reduces it until local minima are obtained. Decay helps in both optimization and generalization.

Table 5: Optimal hyperparameters with a configuration value

S. No.	Parameter	Value
1	Word embedding	BERTbase-cased model
2	Shuffle train dataset	By 2000
3	Shuffle validation dataset	By 100
4	Batch size	16
5	learning rate	1e-5
6	Decay rate	1e-6
7	Loss function	Categorical_Cross_Entropy
8	Optimizer	Adam
9	Epochs	3
10	Activation function	ReLU and SoftMax

4 Results and Discussion

4.1 Experiments

This section discusses and analyzes the experimental results with different text augmentation techniques.

4.1.1 Experiment-1

The proposed model is trained and validated using the original MELD dataset in the first experiment. The MELD dataset contained training instances (9989), the validating instances MELDV (1109), and test instances MELDT (2610). The BERT model transformed the MELD text utterance into a discrete vector. Fig. 5 shows that the validation loss is increasing over each epoch while increasing the number of epochs; the training accuracy of the proposed model is also increasing from 60.64% to 88.32% while the validation accuracy decreases from 60.87% to 56.97%. This behavior of the model shows that the model is overfitting.

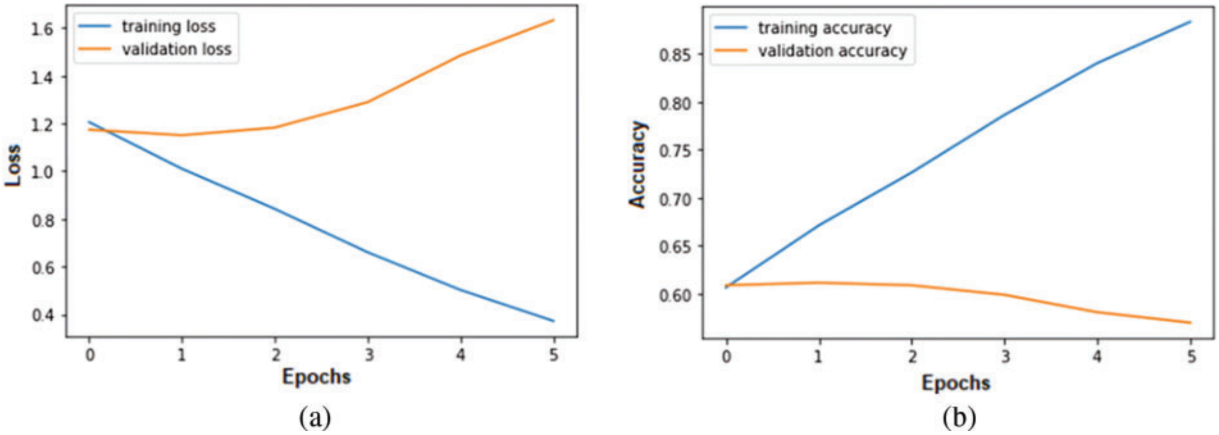


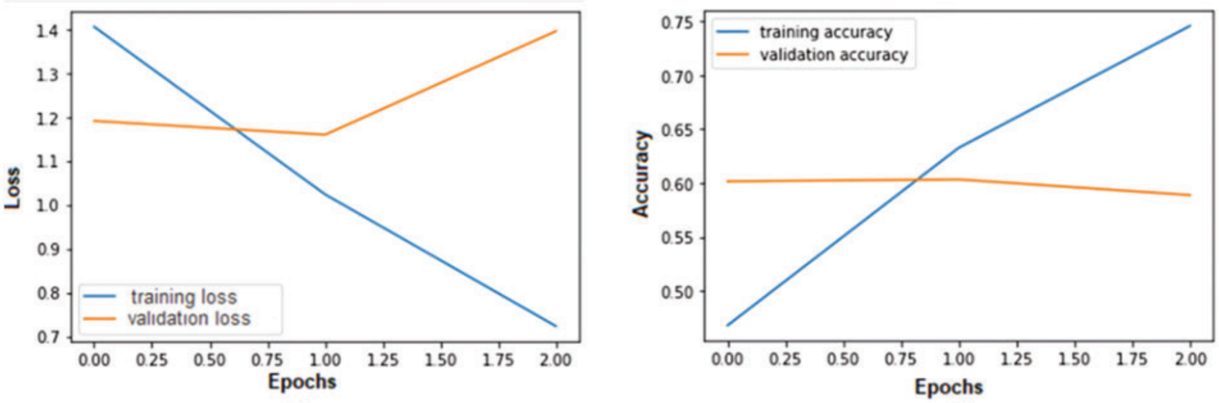
Figure 5: Error loss and accuracy of the proposed model on both training and validation datasets in Experiment-1: (a) error loss of the proposed model; (b) accuracy of the proposed model

4.1.2 Experiment-2

In this experiment, the performance of the proposed model is analyzed by an augmented MLED dataset considering neutral class instances. The neutral class has 4710 instances, which is more than the instances of other classes. Hence, we have increased the number of instances of other classes through augmentation to balance the dataset. Fig. 6 shows that increasing the number of epochs decreases the training error loss while the validation error loss increases. Moreover, the training accuracy of the model is increasing from 52.29% to 83.21%, and validation accuracy is decreasing 57.44% to 54.37% with increasing the number of epochs. The results show that the model is overfitting.

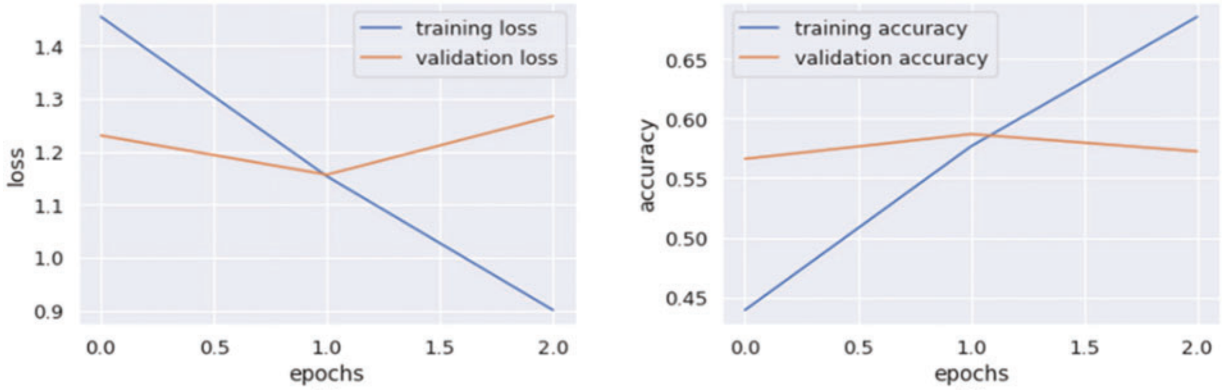
4.1.3 Experiment-3

In this experiment, the performance of the proposed model is analyzed by an augmented MLED dataset considering the joy class instances. The joy class has 1743 instances which are second the biggest in several instances. Hence, we have decreased the number of instances of the neutral class and increased the number of instances of the five remaining classes to balance the dataset. The result of this experiment is illustrated in Fig. 7, which shows that the training error loss decreases by increasing the number of epochs. In contrast, the validation error loss increases, and the training accuracy of the model increases from 43.91% to 68.54%, while the validation accuracy decreases from 58.70% to 57.25%. With increasing the number of epochs, the results show that the model is overfitting.



(a) (b)

Figure 6: Error loss and accuracy of the proposed model on both training and validation datasets in Experiment-2: (a) error loss of the proposed model; (b) accuracy of the proposed model



(a) (b)

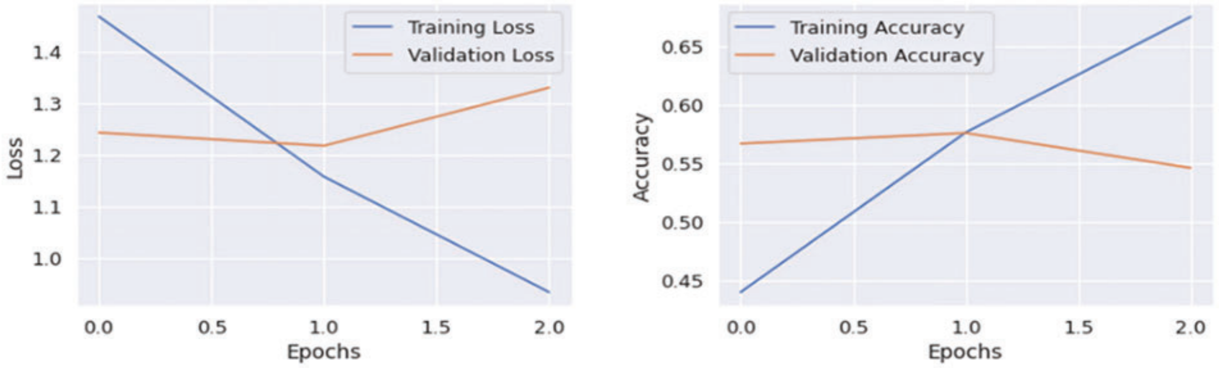
Figure 7: Error loss and accuracy of the proposed model on both training and validation datasets in Experiment-3: (a) error loss of the proposed model; (b) accuracy of the proposed model

4.1.4 Experiment-4

In the fourth experiment, the model is trained by feeding the augmented $\text{MELD}_{\text{A1109}}$ train, testing MELD_{T} (2610), and validating MELD_{V} (1109) dataset. All the other parameters remain the same as in the third experiment. The training and validation loss depicted in Fig. 8 shows that the training accuracy increases over each epoch from 43.99% to 67.55%, while the validation accuracy decreases from 56.70% to 54.62%.

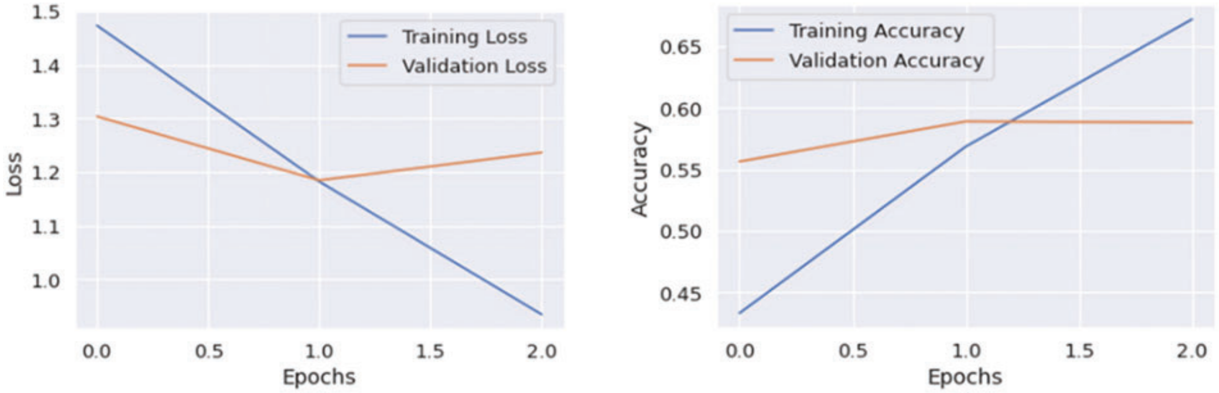
4.1.5 Experiment-5

In the fifth experiment, the model is trained by feeding the augmented $\text{MELD}_{\text{Avg1427}}$ train, testing MELD_{T} (2610), and validating MELD_{V} (1109) datasets. All the other parameters are the same as in the third experiment. The training and validation loss depicted in Fig. 9 shows that the training accuracy increases over each epoch from 43.28% to 67.19%, while the validation accuracy decreases from 58.88% to 57.79%.



(a) (b)

Figure 8: Error loss and accuracy of the proposed model on both training and validation datasets in Experiment-4: (a) error loss of the proposed model; (b) accuracy of the proposed model



(a) (b)

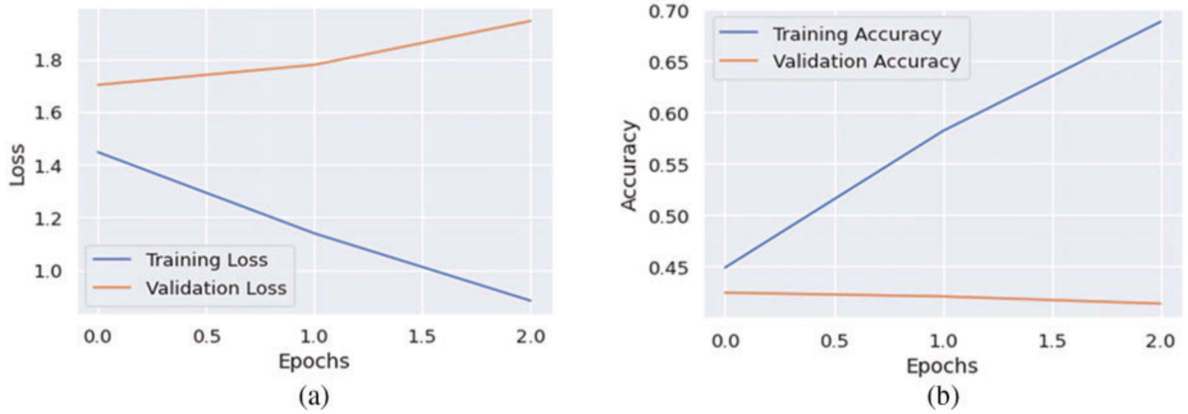
Figure 9: Error loss and accuracy of the proposed model on both training and validation datasets in Experiment-5: (a) error loss of the proposed model; (b) accuracy of the proposed model

4.1.6 Experiment-6

The proposed model is trained in the sixth experiment by feeding the augmented MELD$_{J1743}$ train, testing MELD$_{TJ402}$ (2610), and validating the MELD$_{VN470}$ (1109) datasets. All the other parameters are the same as in the third experiment. Fig. 10 shows that the training accuracy increases over each epoch from 44.90% to 68.84% for three epochs, while the validation accuracy decreases from 42.47% to 41.40%.

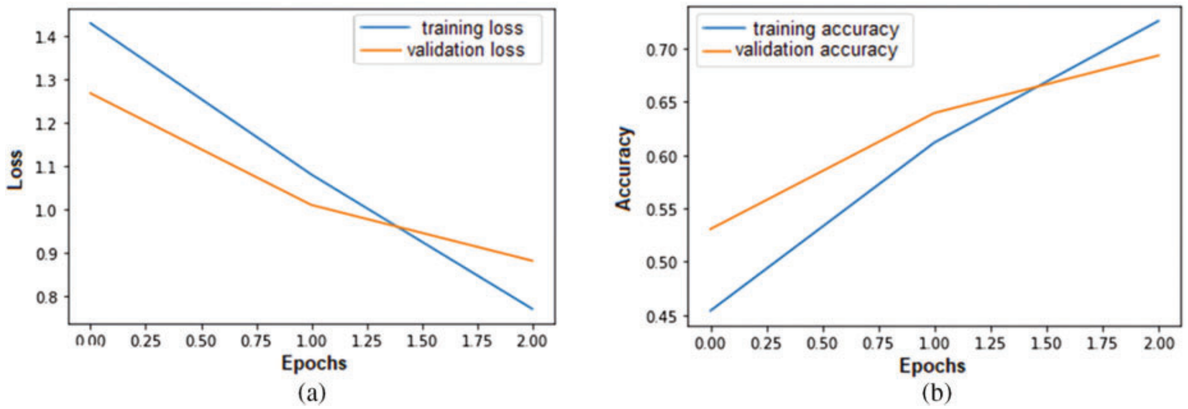
4.1.7 Experiment-7

In the seventh experiment, we train the model by feeding the augmented MELDN4710 train and testing the MELDT datasets. All the other parameters are the same as in the third experiment. Fig. 11 shows that the model training is the best fit because the training accuracy increases from 45.39% to 72.57% over each epoch. In comparison, the validation accuracy also increases over each epoch from 53.05% to 69.33%.



(a) (b)

Figure 10: Error loss and accuracy of the proposed model on both training and validation datasets in Experiment-6: (a) error loss of the proposed model; (b) accuracy of the proposed model



(a) (b)

Figure 11: Error loss and accuracy of the proposed model on both training and validation datasets in Experiment-7: (a) error loss of the proposed model; (b) accuracy of the proposed model

All seven experiments' training, validation, and testing accuracy are shown in Table 6.

Table 6: Impact of data augmentation on the accuracy of the proposed model

Experiment	Dataset			Accuracy			Weighted F1 score
	Train	Val	Test	Training	Validation	Testing	
E1	MELD	MELD$_V$	MELD$_T$	88.32%	56.97%	52.91%	53.78%
E2	MELD$_{N4710}$	MELD$_V$	MELD$_T$	83.21%	54.37%	62.83%	59.45%
E3	MELD$_{J1743}$	MELD$_V$	MELD$_T$	68.54%	57.25%	58.08%	59.00%
E4	MELD$_{A1109}$	MELD$_V$	MELD$_T$	67.55%	54.62%	57.50%	58.93%
E5	MELD$_{Avg1427}$	MELD$_V$	MELD$_T$	67.19%	58.79%	59.84%	60.28%
E6	MELD$_{J1743}$	MELD$_{VN470}$	MELD$_{TJ402}$	68.44%	41.40%	49.73%	49.23%
E7	MELD$_{N4710}$	MELD$_{V20}$	MELD$_T$	72.57%	69.33%	63.10%	61.55%

4.2 *Parameter Tuning*

In this section, the impact of configuration parameters on the performance of the proposed model is analyzed. We have considered two influential parameters for the tuning purpose to improve the performance of the proposed model. These parameters are learning rate and batch size. It is to be noted that for parameter tuning, we have considered Experiment-7.

4.2.1 Learning Rate

The learning rate is a significant factor in machine learning because it controls the size of the model's steps during each iteration. The amount by which weights are updated during the model training phase is known as the step size. The learning rate values range between 0.0 and 1.0. A large value can quickly train the model, but it may overshoot or ignore some of the best characteristics of the features being used during the model training. A small learning rate value may cause overfitting and take longer to train the model. The impact of various learning rate values on the proposed model's performance is presented in Table 7.

Table 7: Impact of learning rate on accuracy and F1 score

Learning rate	Training acc (%)	Validation acc (%)	Testing acc (%)	F1 score (%)
0.1	14.43	14.44	48.12	31.26
0.01	14.07	14.40	13.21	3
0.001	13.85	14.32	7.0	1.0
0.0001	14.42	14.15	1.0	0.0
0.00001	72.57	69.33	63.10	61.55
0.000001	46.75%	48.09%	63.10%	59.74%

Table 7 shows that there is a high variation in both training and testing accuracies when the values of the learning rate are altered. For example, the model generated training and validation accuracies of 14.07% and 14.40%, respectively, on the learning rate 0.1. In contrast, the model yielded 72.57% and 69.33% training and testing accuracies, respectively, using a learning rate 0.0001. These results show that the learning rate substantially impacts the outcome of the proposed model.

4.2.2 Batch Size

Batch size means the total no of training samples appearing in a batch. For the proposed model, we choose the 8, 16, 32, and 64 batch sizes to find an optimal batch size. The impact of batch size on accuracy and F1 score is shown in Table 8.

Table 8: Impact of batch size on accuracy and F1 score

Batch size	Training acc (%)	Validation acc (%)	Testing acc (%)	F1 score (%)
8	76.84	69.85	59.50	59.32
16	72.57	69.33	63.10	61.55
32	66.77	64.58	63.06	62.12

It can be observed from Table 8 that the proposed model generated different outcomes with varying batch sizes. For example, the model produced 76.84% and 69.85% training and validation accuracies using batch size 8, whereas the model obtained 66.77% and 64.58% training and validation accuracies, respectively, using batch size 32. The variations in accuracies show that batch size has a high impact on the performance of the proposed model.

4.2.3 Epochs

The impact of epochs on the performance of the proposed model is presented in Fig. 12. From Fig. 12a, we can observe that the error loss continuously decreases when increasing the number of epochs. Similarly, in Fig. 12b, the proposed model's accuracy increases when the number of epochs increases. These results yielded that the number of epochs impacts the performance of the proposed model. Table 9 shows the performance of the proposed model with two epochs.

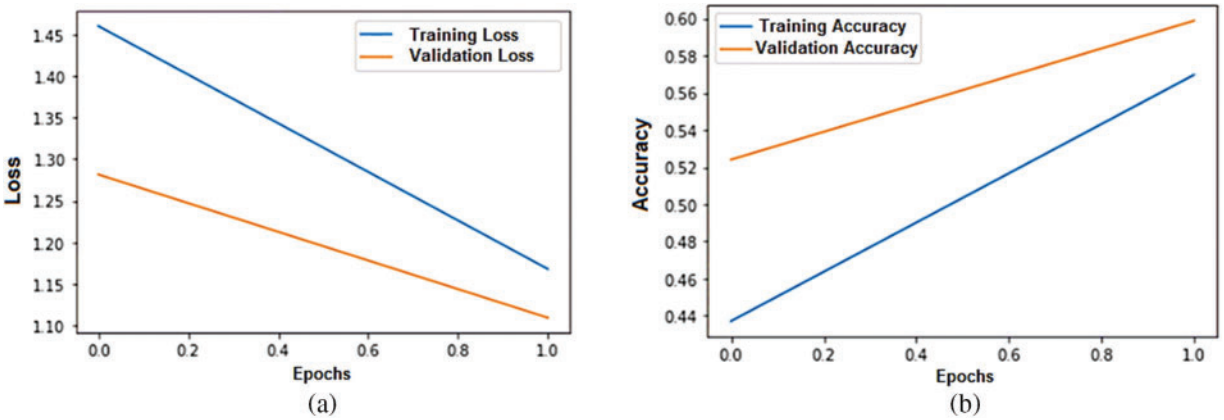


Figure 12: Error loss and accuracy of the proposed model on both training and validation dataset: (a) error loss of the proposed model; (b) accuracy of the proposed model

Table 9: Performance of the proposed model with two epochs

Emotion classes	Precision (%)	Recall (%)	F1 score (%)	Support
Anger	55	37	44	345
Disgust	30	25	27	68
Fear	19	22	20	50
Joy	57	59	58	402
Neutral	73	86	79	1256
Sadness	41	22	28	208
Surprise	61	56	59	281
Accuracy			**64**	**2610**
Macro average	**48**	**44**	**45**	**2610**
Weighted average	**62**	**64**	**63**	**2610**

4.3 Comparison of Proposed Model with Existing Models

This section compares the proposed model's performance with the existing models. The comparison is made with nine currently published models from the literature review, which are: RGCN [5], DialogueGCN [6], DialogueRNN [7], DialogueCRN [8], CESTa [13], AGHMN [15], A-DMN [12], BiERU [19], MM-DFN [21]. The comparison is made regarding the accuracy and weighted F1 score, as shown in Table 10. The proposed model significantly improves on the current state-of-the-art accuracy and F1 score models. For example, considering the accuracy of the proposed model of 63.10%, the second-highest accuracy, 62.49%, was achieved by the MM-DFN [21], and the third-highest accuracy, 60.90%, was achieved by BiERU [19]. Regarding the F1 score, the proposed model scored 61.55%, and the A-DMN [12] achieved 60.45%. These results confirmed that the proposed model improves on the existing models and can recognize emotion with better accuracy and F1 score, illustrated in Table 10.

Table 10: Comparison of accuracy and F1 score of the proposed model and existing models

S. No.	Methods	Accuracy %	Weighted F1 score %
1	RGCN [5]	–	55.98
2	DialogueGCN [6]	59.46%	58.10
3	DialogueRNN [7]	59.54	57.03
4	DialogueCRN [8]	60.73	58.39
5	CESTa [13]	–	58.36
6	AGHMN [15]	–	58.10
7	A-DMN [12]	–	60.45
8	BiERU [19]	60.90	–
9	MM-DFN [21]	62.49	59.46
10	**Proposed model**	**64.36**	**62.60**

Note: ∗∗ The missing values (−) for the attributes show that the model(s) did not consider that attributes in the original paper.

5 Conclusions

This research study presented a computational model for recognizing human emotion in conversation. In this research, we applied text data augmentation on the MELD dataset to balance the dataset. For balancing the dataset, sentences are generated from the MELD dataset utterance without disturbing the meaning of the utterance. In this research, firstly, we employed text augmentation techniques to balance the imbalanced dataset. Secondly, word embedding was employed, and BERT models were used as feature extraction techniques to construct the feature vector. Finally, an emotion recognition system was developed based on the DNN model. The outcome of the proposed emotion recognition system was rigorously assessed using various measurement metrics, including accuracy, weighted F1 score, recall, precision, and confusion matrix. The performance of the proposed model was examined on different learning rates and batch sizes. The results demonstrate that the proposed model achieved the highest accuracy, 64.36%, and the maximum F1 score, 62.60%. Moreover, the proposed model performs significantly better at predicting human emotion in conversation when compared with existing approaches. The proposed model can be used as a helpful tool for identifying human emotion and may have applications in many fields, including psychology in healthcare, student dissatisfaction in education, robotics, automated client support systems, and opinion mining.

Acknowledgement: The authors sincerely appreciate the support from Princess Nourah bint Abdulrahman University Researchers, Riyadh, Saudi Arabia for providing all the necessary technical assistance.

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2023R235), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Fida Mohammad, Mukhtaj khan; data collection: Safdar Nawaz Khan Marwat; analysis and interpretation of results: Neelam Gohar, Naveed Jan, Muhammad Bilal; draft manuscript preparation: Fida Muhammad, Amal Al-Rasheed. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data will be shared up on request.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] N. Sebe, I. Cohen, T. Gevers, T. Huang Nicu Sebe and T. S. Huang, "Multimodal approaches for emotion recognition: A survey," *Internet Imaging*, vol. 5670, no. 17, pp. 56–67, 2005.

[2] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias *et al.,* "Emotion recognition in human-computer interaction," *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 32–80, 2001.

[3] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria *et al.,* "MELD: A multimodal multi-party dataset for emotion recognition in conversations," in *the ACL Conf. on Natural Language Processing*, Florence, Italy, pp. 527–536, 2019.

[4] J. Devlin, M. W. Chang, K. Lee, K. T. Google and A. I. Language, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Computational Linguistics Conf. on Human Language Technologies*, Minneapolis, Minnesota, USA, pp. 4171–4186, 2019.

[5] Y. J. Choi, Y. W. Lee and B. G. Kim, "Residual-based graph convolutional network for emotion recognition in the conversation for smart internet of things," *Big Data*, vol. 9, no. 4, pp. 279–288, 2021.

[6] D. Ghosal, N. Majumder, S. Poria, N. Chhaya and A. Gelbukh, "DialogueGCN: A graph convolutional neural network for emotion recognition in conversation," in *The EMNLP-IJCNLP Conf. on Empirical Methods in Natural Language Processing*, Hong Kong, China, pp. 154–164, 2019.

[7] N. Majumder, S. Poria, D. Hazarika, R. Mihalcea, A. Gelbukh *et al.,* "DialogueRNN: An attentive RNN for emotion detection in conversations," in *The AAAI Conf. on Artificial Intelligence*, Honolulu, Hawaii, USA, pp. 6818–6825, 2019.

[8] D. Hu, L. Wei and X. Huai, "DialogueCRN: Contextual reasoning networks for emotion recognition in conversations," in *The ACL-IJCNLP Conf. on Natural Language Processing*, Bangkok, Thailand, pp. 7042–7052, 2021.

[9] P. Zhong, D. Wang and C. Miao, "DialogueCRN: Contextual reasoning networks for emotion recognition in conversations," in *The EMNLP-IJCNLP Conf. on Empirical Methods in Natural Language Processing*, Hong Kong, China, pp. 154–164, 2019.

[10] S. M. Mohammad, "Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words," in *The ACL Conf. on Economics and Natural Language Processing*, Melbourne, Australia, pp. 174–184, 2018.

[11] R. Speer, J. Chin and C. Havasi, "ConceptNet 5.5: An open multilingual graph of general knowledge," in *The AAAI Conf. on Artificial Intelligence*, San Francisco, California, USA, pp. 4444–4451, 2017.

[12] S. Xing, S. Mai and H. Hu, "Adapted dynamic memory network for emotion recognition in conversation," *IEEE Transactions on Affective Computing*, vol. 13, no. 3, pp. 1426–1439, 2022.

[13] Y. Wang, J. Zhang, J. Ma, S. Wang and J. Xiao, "Contextualized emotion recognition in conversation as sequence tagging," in *The ACL SIGDIAL Conf. on Discourse and Dialogue*, 1st Virtual Meeting, pp. 186–195, 2020.

[14] S. L. Yeh, Y. S. Lin and C. C. Lee, "A dialogical emotion decoder for speech motion recognition in spoken dialog," in *The ICASSP, IEEE Conf. on Acoustics, Speech, and Signal Processing*, Barcelona, Spain, pp. 6479–6483, 2020.

[15] W. Jiao, M. R. Lyu and I. King, "Real-time emotion recognition via attention-gated hierarchical memory network," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 34, no. 5, pp. 8002–8009, 2020.

[16] D. Zhang, L. Wu, C. Sun, S. Li, Q. Zhu *et al.,* "Modeling both context and speaker-sensitive dependence for emotion detection in multi-speaker conversations," in *The IJCAI Conf. on Artificial Intelligence*, Macao, China, pp. 5415–5421, 2019.

[17] D. Sheng, D. Wang, Y. Shen, H. Zheng and H. Liu, "Summarize before aggregate: A global-to-local heterogeneous graph inference network for conversational emotion recognition," in *The ACL Conf. on Computational Linguistics*, Barcelona, Spain, pp. 4153–4163, 2020.

[18] X. Lu, Y. Zhao, Y. Wu, Y. Tian, H. Chen *et al.,* "An iterative emotion interaction network for emotion recognition in conversations," in *The ACL Conf. on Computational Linguistics*, Barcelona, Spain, pp. 4078–4088, 2020.

[19] W. Li, W. Shao, S. Ji and E. Cambria, "BiERU: Bidirectional emotional recurrent unit for conversational sentiment analysis," *Neurocomputing*, vol. 467, no. 5, pp. 73–82, 2022.

[20] T. Ishiwatari, Y. Yasuda, T. Miyazaki and J. Goto, "Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations," in *The ACL Conf. on Empirical Methods in Natural Language Processing*, Barcelona, Spain, pp. 7360–7370, 2020.

[21] D. Hu, X. Hou, L. Wei, L. Jiang and Y. Mo, "Multimodal dynamic fusion network for emotion recognition in conversations," in *The IEEE Conf. on Acoustics, Speech, and Signal Processing*, Sydney, Australia, pp. 7037–7041, 2022.

[22] S. Y. Chen, C. C. Hsu, C. C. Kuo, T. H. K. Huang and L. W. Ku, "EmotionLines: An emotion corpus of multi-party conversations," in *Proc. of the 11th Int. Conf. on Language Resources and Evaluation LREC, 2018*, Miyazaki, Japan, pp. 1597–1601, 2018.

[23] C. Padurariu and M. E. Breaban, "Dealing with data imbalance in text classification," *Procedia Computer Science*, vol. 159, no. 7, pp. 736–745, 2019.

[24] V. Lopez, A. Fernandez, S. Garcia, V. Palade and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, no. 2–3, pp. 113–141, 2013.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," in *Advances in Neural Information Processing Systems 30 NeurIPS*, Long Beach, CA, USA, 2017.

[26] K. Muhammad, A. Ullah, A. S. Imran, M. Sajjad, M. S. Kiran *et al.,* "Human action recognition using attention-based LSTM network with dilated CNN features," *Future Generation Computer Systems*, vol. 125, no. 3, pp. 820–830, 2021.

[27] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed *et al.,* "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[28] G. E. Dahl, T. N. Sainath and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *The IEEE Conf. on Acoustics, Speech and Signal Processing*, Vancouver, Canada, pp. 8609–8613, 2013.

[29] M. B. Kamal, A. A. Khan, F. A. Khan, M. A. Shahid, C. Wechtaisong *et al.,* "An innovative approach utilizing binary-view transformer for speech recognition task," *Computers, Materials & Continua*, vol. 72, no. 3, pp. 5547–5562, 2022.

[30] S. Kumar, M. A. Haq, A. Jain, C. A. Jason, N. R. Moparthi *et al.,* "Multilayer neural network based speech emotion recognition for smart assistance," *Computers, Materials & Continua*, vol. 74, no. 1, pp. 1523–1540, 2023.

[31] P. Sharma and A. Singh, "Era of deep neural networks: A review," in *Proc. of the 8th Int. Conf. on Computing, Communication and Networking Technologies*, Delhi, India, pp. 1–5, 2017.

[32] S. Bosse, D. Maniry, K. R. Muller, T. Wiegand and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 206–219, 2018.

[33] M. Komar, P. Yakobchuk, V. Golovko, V. Dorosh and A. Sachenko, "Deep neural network for image recognition based on the Caffe framework," in *The IEEE Conf. on Data Stream Mining & Processing*, Lviv, Ukraine, pp. 102–106, 2018.

[34] H. Wu, Q. Liu and X. Liu, "A review on deep learning approaches to image classification and object segmentation," *Computers, Materials & Continua*, vol. 60, no. 2, pp. 575–597, 2019.

[35] Y. Zhou, "Natural language processing with improved deep learning neural networks," *Scientific Programming*, vol. 2022, pp. 1–8, 2022.

[36] I. Nirmal, A. Khamis, M. Hassan, W. Hu and X. Zhu, "Deep learning for radio-based human sensing: Recent advances and future directions," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 995–1019, 2021.

[37] A. Bizzego, G. Gabrieli and G. Esposito, "Deep neural networks and transfer learning on a multivariate physiological signal dataset," *Bioengineering*, vol. 8, no. 3, pp. 35, 2021.

[38] I. Nusrat and S. B. Jang, "A comparison of regularization techniques in deep neural networks," *Symmetry*, vol. 10, no. 11, pp. 648, 2018.

[39] C. Xu, D. Chai, J. He, X. Zhang and S. Duan, "InnoHAR: A deep neural network for complex human activity recognition," *IEEE Access*, vol. 7, pp. 9893–9902, 2019.

[40] S. Han, R. F. Zhang, L. Shi, R. Richie, H. Liu *et al.,* "Classifying social determinants of health from unstructured electronic health records using deep learning-based natural language processing," *Journal of Biomedical Informatics*, vol. 127, no. 6, pp. 103984, 2022.

[41] I. Bilbao and J. Bilbao, "Overfitting problem and the over-training in the era of data: Particularly for artificial neural networks," in *The IEEE Conf. on Intelligent Computing and Information Systems*, Krakow, Poland, pp. 173–177, 2017.

[42] V. Silva, "Real-time emotions recognition system," in *Proc. of the Int. Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Lisbon, Portugal, pp. 201–206, 2016.

[43] C. Halimu, A. Kasem and S. H. S. Newaz, "Empirical comparison of area under ROC curve (AUC) and Mathew correlation coefficient (MCC) for evaluating machine learning algorithms on imbalanced datasets for binary classification," in *The ACM Conf. on Machine Learning and Soft Computing*, New York, USA, pp. 1–6, 2019.