

Emotion Detection and Analysis using Textual Data through Trainable and Pre-trained Word Embedding Methods

Majdah Alvi¹, Adnan Akhter¹, Muhammad Bux Alvi^{1*}, Noor Fatima¹

¹Department of Computer Systems Engineering, Faculty of Engineering, The Islamia University of Bahawalpur, Punjab, Pakistan

Keywords: Artificial Neural Networks, Emotion Analysis, Emotion Detection, Pre-Trained Embeddings, Trainable Word Embeddings

Journal Info:

Submitted:
April 05, 2025
Accepted:
April 28, 2025
Published:
May 03, 2025

Abstract

Emotion expression modes play a significant role in human communication. Humans use emotions to convey their state of mind to each other on platforms such as X (formerly Twitter), Facebook, and other online social networks. People often express their emotions using free text, which triggers a vast research area of emotion detection and analysis. This work aims to detect and analyze emotions from unstructured text data. For this purpose, this research study proposes a solution to the problem by building a deep artificial neural network model using trainable and pre-trained word embedding methods. Afterward, the performance of the models developed with different word embeddings is evaluated using the performance metrics. Experimental works demonstrate that the deep artificial neural network with trainable word embedding surpassed all other models by achieving 67.36% accuracy, 53.27% recall, 82.62% precision, and 64.50% F-measure.

*Correspondence author email address: mbalvi@iub.edu.pk

DOI: [10.21015/vtse.v13i2.2115](https://doi.org/10.21015/vtse.v13i2.2115)

1 Introduction

Emotions are an inevitable part of human communication. Humans use emotions to convey their state of mind in various ways. People use text, emojis, and emoticons to express their emotions on social media and the Internet [1]. Human emotions in communication through text trigger a vast research area, i.e., emotion detection and analysis. Emotion analysis is derived from the sentiment analysis domain, where human orientation can be judged from a given piece of text data [2]. Paul Ekman initially suggested six

emotional states: "joy", "sadness", "disgust", "anger", "enjoyment", and "surprise" [3]. Robert Plutchik in [4] then extended Paul's list by adding two more emotion states: "trust" and "anticipation". Ortony et al. disagreed with the emotional state analogy of Ekman and Plutchik [5]. They proposed 22 emotion categories out of which 16 emotions were added to the proposed Ekman's emotions as "hope", "like", "dislike", "self-reproach", "shame", "envy", "relief", "appreciation", "reproach", "pity", "gratification", "fears-confirmed", "gloating", "admiration", "disappointment", and "grief".



This work is licensed under a Creative Commons Attribution 3.0 License.

Parrot in [6] further restructured Ekman's six emotions into a hierarchical model, organizing them into primary, secondary, and tertiary levels, consisting of 100 different emotions. These emotions are represented in Figure 1.

Russell in [7] represented the emotional states in a two-dimensional circular model as shown in Figure 2. The author differentiated emotion states in the arousal and valence fields. The arousal domain distinguishes emotion states by activation and deactivation, whereas the valence domain distinguishes them by pleasantness and unpleasantness. Plutchik in [4] represented the emotional states in a two-dimensional wheel as given in Figure 3. The wheel exhibits emotional states in concentric circles. The innermost part of the wheel represents the eight fundamental emotions first, then the eight basic emotions, and finally a combination of primary emotional states in the outermost part. Russell and Mehrabian in [8] represented three-dimensional emotion states comprising arousal, valence, and dominance fields. Arousal and valence work the same as in [7], while the third dimension describes the degree to which examiners control their emotions. This work's motivation stems from the fact that people often express their point of view and show their emotions using free text. Therefore, there is a need to develop a model that can detect emotions expressed within a piece of text and analyze them for actionable insights.

This study proposes a solution by developing deep artificial neural network models with trainable and pre-trained word embedding methods, including embedding layer, Word2Vec, and GloVe. These models can detect and analyze emotions from the text dataset, and their performance is evaluated through metrics such as accuracy, recall, precision, and F-measure [9]. The obtained results from the experimental work show that the deep artificial neural network with trainable word embeddings, such as the embedding layer, surpassed all other developed models with pre-trained word embeddings, such as Word2Vec and GloVe.

This section introduced the research study. Section II encompasses the related literature review.

Afterward material and methods will be discussed in section III, followed by results presentation in section IV along with their elaboration. Section V concludes the study and provides suggestions. At the end cited work is enlisted.

2 Literature Study

Many researchers have presented their studies on sentiment and emotional analysis in recent years. This section summarizes recent research on emotion analysis using text data.

Adil Majeed et al. [10] used supervised learning approaches, including k-nearest neighbor (KNN), decision tree (DT), random forest (RF), and support vector machine (SVM), for emotion classification in Roman Urdu text. The authors collected a corpus of text data from selected sites, such as YouTube, Hamariweb, Twitter, What-Mobile, and Drama Online, using 'selenium' and 'twint' scraping tools. The dataset consisted of 18,000 labeled sentences with six classes: happy, anger, sad, fear, love, and neutral. The authors achieved 69% accuracy with SVM compared to KNN, random forest, and decision tree using Word2Vec. Matla Suhasini et al. [11] developed the system using Naive Bayes (NB) and KNN for emotion classification. They considered tweets and label features from the dataset collected from the sentiment140. The dataset consisted of 5,000 samples and four emotion classes such as c1(happy-active), c2(happy-inactive), c3(unhappy-active), and c4(unhappy-inactive). The authors reported achieving performance accuracy of 72.60% using the Naive Bayes (NB) algorithm and 55.50% with KNN. Ahmad Fakhri Ab. Nasir et al. in [12] applied SVM, Multinomial Naive Bayes (MNB), KNN, and decision tree developed models by constructing them first for text-based emotion classification. The authors used the ISEAR (International Survey on Emotion Antecedents and Reactions) dataset, which consisted of 6,570 corpus texts and six emotion states (anger, disgust, sadness, fear, joy, and guilt), with each state containing 1,096 records. They used the Bag-of-Words (BOW) technique for feature representations. The results produced from the experiment showed that Multinomial Naive Bayes based classification

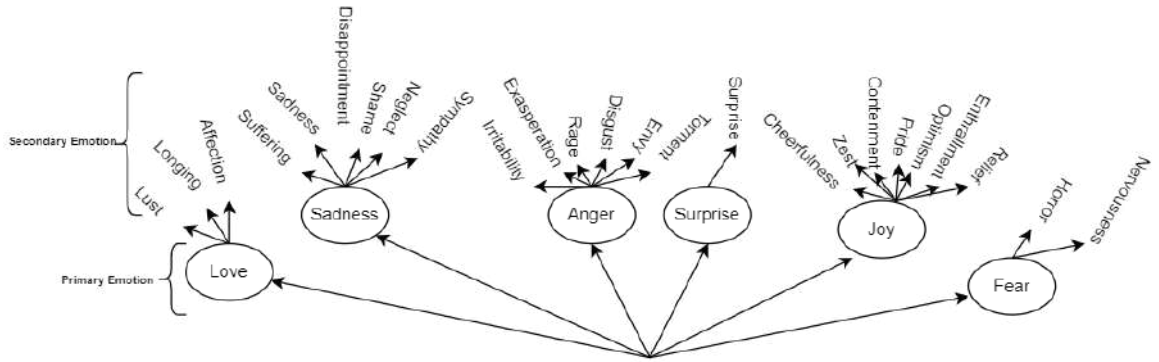


Figure 1. Parrot's Hierarchical Model of Emotion States [6]

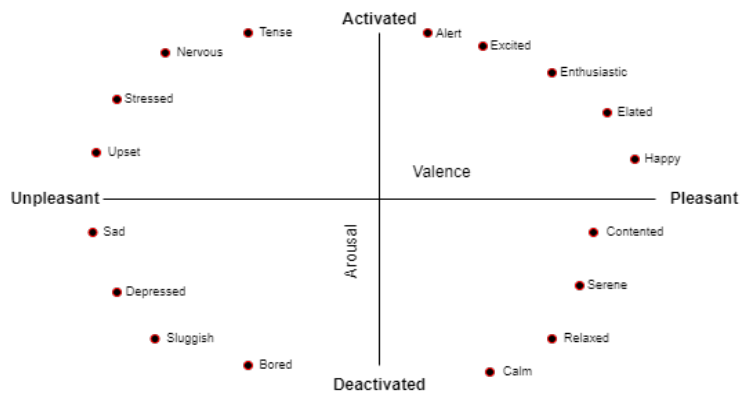


Figure 2. Russell's Two-Dimensional Emotion States Model [7]

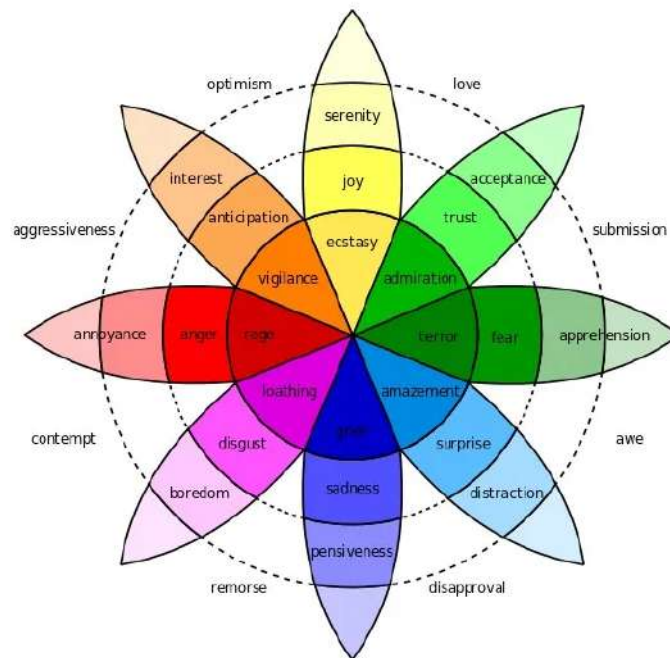


Figure 3. Plutchik's Emotion Wheel [4]

model achieved 64.08% of accuracy, surpassing DT (52.36%), KNN (47.95%), and SVM (15.37%) based models.

The authors in [13] developed J48, NB, KNN, and SVM-based models for the emotion classification in Indonesian text. The authors used a text corpus of 1,000 Indonesian sentences and six emotion classes (disgust, anger, joy, fear, surprise, and sadness), collected from the Dongeng website. They used the BOW technique for feature representation. The experimental results showed that the SVM-SMO model achieved 85.5% accuracy, outperforming NB (80.2%), J48 (80.8%), and KNN (68.1%) models. Xueying Zhang et al. [14] applied support vector machine, ELM with kernel ¹, and developed models by constructing them first for the sentiment analysis in Chinese text. They acquired the Chinese text corpus from Hotel BB, which consisted of user comments classified into two emotion classes: positive and negative emotions. They used the Bag-of-Words and TF-IDF techniques for text feature vectorization. The authors reported that an extreme learning machine with kernels achieved 88.74% accuracy, slightly better than SVM based model that obtained 88.54% accuracy. Jia Zheng [15] employed the deep learning bidirectional long short-term memory (LSTM) algorithm to build a text-based emotion classification model. The author used 37,167 text samples and six emotional states (sadness, happiness, surprise, fear, jealousy, and anger). The author converted the text into word vectors, representing features at the word level and character level, using word embedding. The author achieved 64% accuracy using a bi-LSTM model.

Jacek Filipczuk et al. [16] applied deep learning techniques, including Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN), to develop models for text-based emotion classification. The authors used word features as feature columns. They used ISEAR and IMDB datasets. The ISEAR dataset consists of 7,666 statements and seven emotional states: joy, fear, anger, sadness, disgust, shame, and guilt, whereas IMDB contains 25,000 samples with no predefined

emotion labels (self-built). The authors reported that they achieved accuracies of 24.13% (MLP, ISEAR), 96.25% (MLP, IMDB), 96.65% (CNN, IMDB), 36.21% (CNN, ISEAR), 93.20% (LSTM, IMDB), and 41.00% accuracy with LSTM while using the ISEAR dataset. Mansur Alp Tocoglu et al. [17] developed deep learning models using CNN, Recurrent Neural Network (RNN), LSTM, and Artificial Neural Network (ANN) to classify Turkish tweets. The authors used two datasets: Turkish tweet emotion (Turted), which contained 195,000 tweets and six emotion classes (happiness, sadness, fear, anger, disgust, and surprise), and the other was 'Tremo'. The authors reported that on the 'Turted' dataset, CNN achieved 74% accuracy, whereas LSTM obtained 73% accuracy, and ANN produced 71% accuracy. Ekasari Nugraheni [18] applied Multi-Layer Perceptron (MLP) and LSTM-based models by constructing them first for the emotion classification in the presidential election. The author used an Indonesian Twitter dataset consisting of 100,145 tweets and six emotional states (disgusted, happy, angry, sad, afraid, and surprised). The dataset was divided into imbalance-standard, imbalance-extra, balance-standard, and balance-extra. The author reported accuracies of 42.34% on standard and 43.21% on extra using MLP with an imbalanced dataset, 71.41% on standard, and 71.74% on extra using MLP with a balanced dataset. It also reported achieving accuracies of 43.31% on standard and 40.59% on extra using LSTM with an imbalanced dataset, 69.09% on standard, and 70.18% on extra using LSTM with the balanced dataset.

Dongliang Xu et al. [19] developed CNN-text-Word2Vec, LSTM, RNN, SVM, logistic regression, random forest, NB, and decision tree models to classify emotions in micro-blog text. The authors used a Chinese micro-blog dataset from Sina Weibo, consisting of 80,000 micro-blog messages, of which 40,000 expressed positive emotions and 40,000 negative emotions. They used the Word2Vec embedding technique for feature representation at both the Chinese character level and word level. The authors reported that CNN-text-Word2Vec achieved 97.60% accuracy with character-level Word2Vec and 94.51% accuracy with word-level Word2Vec. Mounika Karna

¹Extreme Learning Machine with kernels

et al. [20] used Long Short-Term Memory (LSTM), nested Long Short-Term Memory (CNN kernels are applied to an embedding matrix, and the resultant embedding is fed into LSTM), and support vector machine (SVM)-based developed models by fabricating them for the classification of emotions in text. They used an emotion classification dataset containing six emotion states. They employed a word embedding technique for word feature representation. They reported after the experimental work that LSTM achieved 94.15% accuracy, surpassing nested LSTM and SVM models, which achieved 92.34% and 93.1% accuracies, respectively.

3 Materials And Methods

This section describes the method adopted for emotion detection and analysis using unstructured (text) data. Figure 4 illustrates the proposed method for emotion detection and analysis.

3.1 Dataset

Data collection is the process of gathering information from different sources. The dataset named ArtEmis has been collected from the Art Emotion website ². The ArtEmis dataset was developed collaboratively by researchers from Stanford University, King Abdullah University, and École Polytechnique. It is a large-scale dataset that narrates an affective language for visual art (painting) that derives emotional responses from viewers in textual form. Annotators (viewers) were asked to explain how they felt about a given image and specify the dominant emotions, such as Contentment, Awe, Amusement, Sadness, Excitement, Fear, Anger, Disgust, and Something else. The ArtEmis dataset consists of approximately 455,000 text samples and five attributes: Art Style, Painting, Emotion, Utterance, and Repetition. The ArtEmis dataset has used 80,031 unique artworks (paintings/images) for effective annotations. The independent attribute, utterance and dependent attribute emotion are used for the emotion analysis (classification). It has all non-null values. The dataset includes one numerical attribute (integer type) and four categorical attributes (string

type). The attributes description is given below in Table 1.

Table 1. ArtEmis Dataset Attributes Description

Attribute	Description	Data Type
Art_Style	Art_Style attribute gives the style of the image given for responses, and annotations	Non-Numerical
Painting	Painting attribute is the image/painting on which annotators provide responses and emotion	Non-Numerical
Utterance	Utterance attribute is that in which annotators give an explanation about the image/painting in support of the dominant emotions	Non-Numerical
Emotion	Emotion attribute specify the dominant emotions which consist of Contentment, Awe, Excitement, Amusement, Sadness, Anger, Fear, Disgust, and Something else that are provided by the annotators on a given image	Non-Numerical
Repetition	Repetition attribute is an integer noting how many annotators have provided explanations for each artwork	Numerical

3.2 Text Preprocessing

Text data is highly prone to noise. Text preprocessing cleans and prepares the text data before feeding it to an algorithm to build a model and evaluate its performance [21]. The following preprocessing steps were implemented to prepare the dataset for this study.

3.2.1 Case Normalization

Lowercasing is the process of converting mixed-case (uppercase and lowercase) text data to lowercase. The objective of case normalization is to bring all the tokens in a uniform shape to curb the vector space and make it more interpretable by the machine [22, 23].

²www.artemisdataset.org/

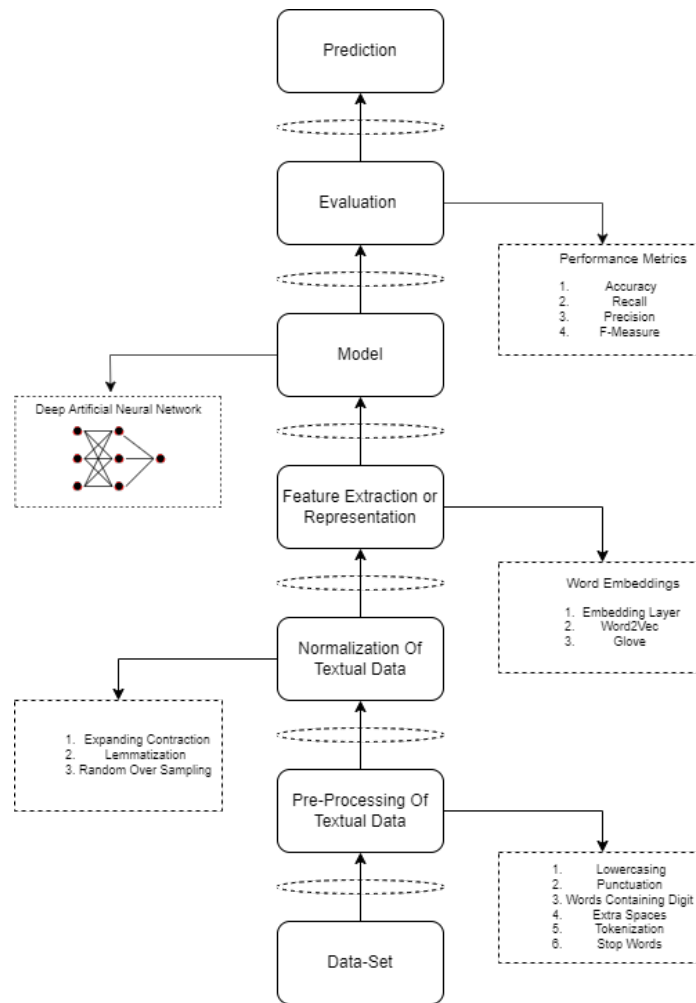


Figure 4. Emotion Detection and Analysis Model

3.2.2 Punctuation Handling

Punctuation handling is a natural language processing (NLP) process that removes symbols such as commas, semicolons, colons, delimiters, etc., from text data. Punctuation does not provide meaningful information. Their presence in the text data makes it noisy (increasing the vector space) and decreases the model performance. To enhance performance, punctuation symbols are therefore removed from text data [22, 23].

3.2.3 Word Containing Digits

It is the process in which alphanumeric tokens (words = characters + digits) are cleaned to obtain tokens consisting only of characters. The presence of alphanu-

meric tokens creates difficulty for the machine to interpret. Such tokens are removed from the text data.

3.2.4 Extra Spaces

Extra spaces often occur in text data that contains more than one space, adding unnecessary noise. These extra spaces with no semantic information are removed from the text for text analysis, leaving only one space behind.

3.2.5 Tokenization

Tokenization is the process of breaking text, sentences, or phrases into smaller chunks or separate words, called tokens. The primary purpose of tokenization is to identify and isolate valid words within a sentence [24].

3.2.6 Stop Words

Stop words are the linguistic terms frequently present in the text, having inherently no helpful information. The stop words add noise to the corpus when examining it. These stop words may be removed from the corpus to make the model more efficient [?].

3.3 Text Normalization

Text Normalization is the process of natural language processing in which the tokenized words or inflected words in the text are converted into the standard or base form.

3.3.1 Expanding Contraction

Contraction is the method in which the word(s) are shortened by dropping the letter(s), which are replaced with the help of the apostrophe(s). These contractions often occur in the text data. For the text normalization, the contractions are either expanded or removed [25].

3.3.2 Lemmatization

Lemmatization is a natural language processing (NLP) method that reduces inflected forms of words to their root or base form (dictionary form), known as the "lemma". It is a dictionary-based method. It is grammatically more accurate than the stemming method for text normalization [26].

3.3.3 Random Oversampling (ROS)

Random oversampling is a method that increases the number of minority classes by selecting observations randomly from the dataset to balance the skewed class distribution. This method is simple and easy to implement [27–29].

3.4 Feature Representation

Feature representation methods transform document features or words into numerical vectors, which are subsequently fed into algorithms for model development.

3.4.1 Word Embedding

Word embedding is a feature representation method that represents semantically similar features with similar real-valued vectors. Word embedding reduces di-

mensionality by mapping words with similar meanings to similar real-valued vectors in a lower-dimensional space. These words are grouped closer together in the lower-dimensional space [30]. There are two kinds of word embedding methods used: trainable word embeddings and pre-trained word embeddings.

3.4.2 Trainable Word Embedding

The embedding Layer is a type of word embedding used in neural networks to learn and create embeddings of words from text data. Each unique word in the text data is encoded with a unique integer by the tokenizer in the Embedding Layer. The embedding layer consists of three parameters: 1) Input Dimension, 2) Output Dimension, and 3) Input Length.

3.4.3 Pre-Trained Word Embedding

Word2Vec is an efficient method for generating embeddings from words. Word2Vec generates embeddings using two layers within a shallow feed-forward neural network. This network predicts the similarity and semantic relationships between words. The input of Word2Vec is the text data, and the output of Word2Vec is the feature vectors (feature vector represents the word in the text data). Word2Vec turns the words into a neural network interpretable form for processing. The pre-trained word vectors, developed by Google, are used to create embeddings from text data before feeding them into an algorithm [31, 32].

Global Vectors for Word Representation (GloVe) is an extended approach to the Word2Vec method for creating word embeddings. GloVe produces word embeddings by aggregating the global word co-occurrence matrix of text data. The co-occurrence matrix produces the relations between the words. The co-occurrence matrix identifies word pairs that frequently co-occur. GloVe processes the entire text data, which considers the words from the whole text, not local, and builds the large matrix that can find co-occurrences of words in text data. The pre-trained GloVe word vectors, developed by Stanford University, create embeddings of text data before forwarding them to an algorithm to train the model [33].

3.5 Deep Artificial Neural Networks

Artificial Neural Networks (ANNs) are employed to solve complex problems. Biological neurons (Human Neurons) inspire Artificial Neural Networks, mimicking the behavior of the human brain. ANNs consist of neurons that are arranged in interconnected layers [34, 35]. Artificial Neural Networks are classified into forward propagation and backward propagation. Feed-forward and feed-backward mechanisms allow signals to flow from input to the output through hidden layers and vice versa [36]. It is bidirectional in backward but unidirectional for feed-forward—no feedback loop in feed-forward and vice versa. The functional principle of feed-forward and feed-backward propagation consists of an input data layer, passing such input to the hidden layers for computations. After computation, the hidden layers pass the resulting values to the output layer. Feed-backward propagation propagates errors concerning weights backward from the output layer to the input layer using the optimizer. Figure 5 demonstrates their working.

The feed-forward neural network is mathematically represented as:

$$y = \sum_{i=1}^n w^i x^i + b \quad (1)$$

$$Hidden_layer(H_1) = activation_function(y) \quad (2)$$

$$Hidden_layer(H_2) = activation_function\left(\sum_{k=1}^n H_1^i w^i + b\right) \quad (3)$$

...

$$H_i(curr_layer) = activation_function\left(\sum_{k=1}^n H_{prev}^i w^i + b\right) \quad (4)$$

$$output(z) = activation_function\left(\sum_{k=1}^n H_{curr_layer}^i w^i + b\right) \quad (5)$$

In Equation 1, w represents the weights, x is the input nodes (data), and b is the bias. The Equation represents the summation of the input nodes associated

with weights (dot product of input and weights) and the bias. Equations 2 and 3 represent the computations in the hidden layers. The number of hidden layers may vary, as all computations are performed in the hidden layers. Equation 4 represents the generalized form for the hidden layers. The two hidden layers are used. Equation 5 represents the output produced by applying activation functions to Equation 4.

3.5.1 SoftMax function

The SoftMax function predicts relative probabilities and returns the probability of each class. It is often described as a generalized form of multiple sigmoid functions. The objective of the SoftMax function is to take a multi-dimensional vector containing real values and form another multi-dimensional vector of real values in the output layer in the range of 0 to 1. The SoftMax functions are used for handling multi-class classification problems, sometimes called soft argmax or multi-class logistic regression [37]. Mathematically, the SoftMax function is represented as shown in Equation 6:

$$Softmax_function(\sigma(z)_k) = \frac{\exp(z_k)}{\sum_{i=1}^i \exp(z_i)} \quad (6)$$

Here, in Equation 6, $\exp(z_k)$ is the exponential function for the input vector, while $\exp(z_i)$ belongs to the summation of the output vector exponential function.

3.5.2 Categorical cross-entropy

The categorical cross-entropy cost function is applied in classification problems involving two or more classes. The one-hot encoded representation of classes/labels is required to use this loss function. Categorical cross-entropy is measured as the sum of losses for each class per observation [38–40]. The mathematical expression for categorical cross-entropy [41] is represented in Equation 7.

$$Categorical_Cross_Entropy = - \sum_{i=1}^N x_{i,c} \cdot \log(P_{i,c}) \quad (7)$$

Equation 7 represents a categorical cross-entropy loss function for multi-class classification. N belongs

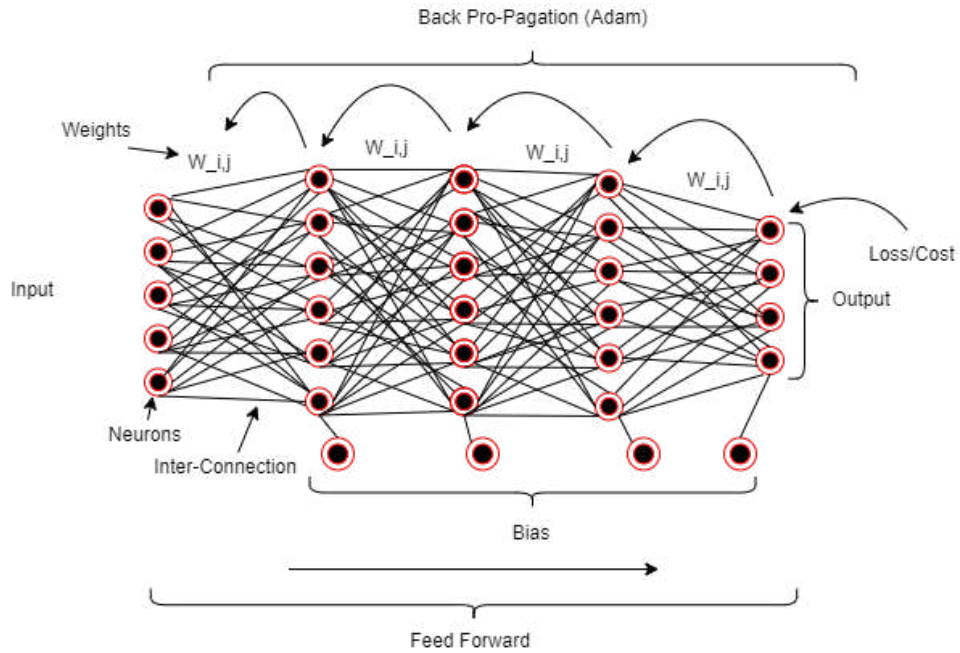


Figure 5. Architecture of feed-forward and feed-backward propagation

to the summation of the number of classes, whereas p represents the predicted probability. x are the target, i belongs to data numbers, and c belongs to number of the class.

3.5.3 Adaptive Moment Estimation (Adam)

Adam is used to adjust the learning rate for each hyperparameter adaptively. For faster convergence, Adam employs adaptive learning rates and momentum. The major objective is to minimize the cost function with respect to the weight. Adam combines stochastic gradient descent with momentum and RMSProp (Root Mean Squared Propagation) optimizers. Adam takes the average of exponentially decaying, which is identical to the momentum of past gradients (momentum and RMSProp). It is a more effective and efficient optimizer compared to others [42]. The mathematical expressions of these adaptive moment estimations from the stochastic with momentum and RMSProp are given below:

$$a_t = \beta_1 * a_{t-1} - (1 - \beta_1) * g_t \quad (8)$$

$$b_t = \beta_2 * b_{t-1} - (1 - \beta_2) * g_t^2 \quad (9)$$

$$g_t = \frac{\partial C}{\partial W_t} \quad (10)$$

$$a'_t = \frac{a_t}{1 - \beta_1^t} \quad (11)$$

$$b'_t = \frac{b_t}{1 - \beta_2^t} \quad (12)$$

$$\Delta W_t = -\alpha * \left(\frac{1}{\sqrt{b'_t + \epsilon}} * a'_t \right) \quad (13)$$

The equation of Adam for the weight updation becomes:

$$Weight_{t+1} = Weight_t + \Delta Weight_t \quad (14)$$

Equations 8 and 9 represent the average exponential, the average exponential of the squares of past gradients for each and all parameters. The equations 11 and 12 represent a bias correction. In equation 13, the learning rate is multiplied by the average exponential of the gradient and divided by the square root of the average exponential of the squares of the gradients. Equation 14 represents the weight update using an exponentially weighted average.

3.6 Evaluation Metrics

Performance metrics evaluate the developed models [43, 44].

3.6.1 Accuracy

Accuracy is the most commonly used performance metric for evaluating the performance of classification models. Accuracy is the ratio of the correctly predicted observations to the total observations. High accuracy indicates good model performance. Accuracy is a reliable metric when the dataset is symmetrical. The mathematical expression of the accuracy is given as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (15)$$

3.6.2 Precision

Precision is the ratio of predicted results that are correctly judged by the model to the total predicted positive results. It is used when false negatives occur in such a way that it is unacceptable. Mathematically, it is represented as:

$$Precision = \frac{TP}{TP + FP} \quad (16)$$

3.6.3 Recall

Recall is the ratio of predicted results that are correctly judged by the model to the total actual positive results. It is used when false negatives occur in such a way that it is unacceptable. The mathematical formula is described below:

$$Recall = \frac{TP}{TP + FN} \quad (17)$$

3.6.4 F-Measure

F-Measure is the harmonic mean of precision and recall. It is broadly used for retrieving information. F-Measure takes all false positives and all false negatives into account. It gives the same weight to recall and precision. It solves the problem of accuracy by performing well when classes in the dataset are unevenly distributed. The expression of F-measure is described below:

$$F - Measure = \frac{2 * (recall * precision)}{(recall + precision)} \quad (18)$$

4 Results

This section presents the results obtained from the experimental evaluation. The results were obtained by validating test data using the Deep Artificial Neural Network (Deep ANN) with an embedding layer for emotion classification on the ArtEmis dataset. Table 2 shows the results of the Deep Artificial Neural Network with an embedding layer.

As shown in Table 2, the Deep ANN with an embedding layer achieved the best results on testing data at epoch 10, beginning from epoch 1. At epoch 10, the best results were observed with a categorical cross-entropy loss of 1.0793, an accuracy of 0.6741, an F-measure of 0.6479, a precision of 0.8271, and a recall of 0.5326.

Table 3 presents the results of emotion classification on the ArtEmis dataset using the Deep ANN with Word2Vec. As shown in Table 3, the Deep ANN with Word2Vec embeddings achieved its best performance at epoch 10, with a categorical cross-entropy loss of 1.1915, an accuracy of 0.6060, an F-measure of 0.5638, a precision of 0.7639, and a recall of 0.4470.

Table 4 shows the results produced using the Deep ANN with GloVe for emotion classification of the ArtEmis data. Table 4 indicates that the Deep ANN using GloVe embeddings achieved its best results on the test dataset at epoch 10, with a loss of 1.3851, an accuracy of 0.5345, an F-measure of 0.4625, a precision of 0.7102, and a recall of 0.3431.

The results of the Deep ANN with Embedding Layer, Word2Vec, and GloVe were evaluated using the testing dataset. The evaluated results in terms of accuracy, F-measure, precision, and recall have been compared in Table 5.

As shown in Table 5, the Deep ANN with the Embedding Layer outperformed Word2Vec and GloVe for emotion analysis on the ArtEmis dataset, achieving the highest scores for accuracy (0.6736), F-measure (0.6450), precision (0.8262), and recall (0.5327). The Deep ANN with GloVe has produced less adequate results: 0.5345 (accuracy), 0.3429 (recall), 0.7102 (precision), and 0.4591 (F-measure). The Deep ANN with Word2Vec produced moderate results. The quantitative comparison of results with related research has

Table 2. Experimental results of Deep ANN with Embedding Layer

Epoch	Loss	Accuracy	F-measure	Precision	Recall
1	1.1326	0.6605	0.6222	0.8339	0.4963
2	1.1233	0.6637	0.6281	0.8328	0.5044
3	1.1152	0.6646	0.6307	0.8346	0.5070
4	1.1085	0.6665	0.6336	0.8355	0.5104
5	1.1017	0.6686	0.6359	0.8340	0.5140
6	1.0950	0.6701	0.6392	0.8324	0.5189
7	1.0908	0.6708	0.6414	0.8295	0.5229
8	1.0863	0.6732	0.6443	0.8308	0.5264
9	1.0826	0.6729	0.6451	0.8289	0.5282
10	1.0793	0.6741	0.6479	0.8271	0.5326

Table 3. Experimental results of Deep ANN with Word2Vec

Epoch	Loss	Accuracy	F-measure	Precision	Recall
1	1.2832	0.5712	0.4715	0.7900	0.3362
2	1.2305	0.5883	0.5196	0.7779	0.3902
3	1.2115	0.5954	0.5354	0.7747	0.4092
4	1.2024	0.5985	0.5468	0.7698	0.4241
5	1.1975	0.6014	0.5524	0.7686	0.4313
6	1.1943	0.6034	0.5550	0.7665	0.4351
7	1.1926	0.6038	0.5589	0.7656	0.4402
8	1.1921	0.6049	0.5631	0.7625	0.4464
9	1.1915	0.6053	0.5635	0.7629	0.4468
10	1.1915	0.6060	0.5638	0.7639	0.4470

Table 4. Experimental results of Deep ANN with GloVe

Epoch	Loss	Accuracy	F-measure	Precision	Recall
1	1.4136	0.5203	0.4031	0.7429	0.2767
2	1.3900	0.5288	0.4382	0.7257	0.3140
3	1.3850	0.5310	0.4512	0.7162	0.3295
4	1.3837	0.5328	0.4526	0.7199	0.3302
5	1.3823	0.5335	0.4552	0.7176	0.3334
6	1.3832	0.5339	0.4592	0.7112	0.3392
7	1.3831	0.5340	0.4592	0.7136	0.3387
8	1.3836	0.5335	0.4614	0.7111	0.3417
9	1.3851	0.5343	0.4608	0.7122	0.3408
10	1.3851	0.5345	0.4625	0.7102	0.3431

been shown in Table 6.

Table 5. Evaluation Results of Deep ANN with Trainable and Pre-trained Word Embeddings

Models with Word Embeddings	Loss	Accuracy	F-measure	Precision	Recall
Deep ANN with Embedding Layer	1.0803	0.6736	0.6450	0.8262	0.5327
Deep ANN with Word2Vec	1.1915	0.6060	0.5609	0.7633	0.4469
Deep ANN with GloVe	1.3851	0.5345	0.4591	0.7102	0.3429

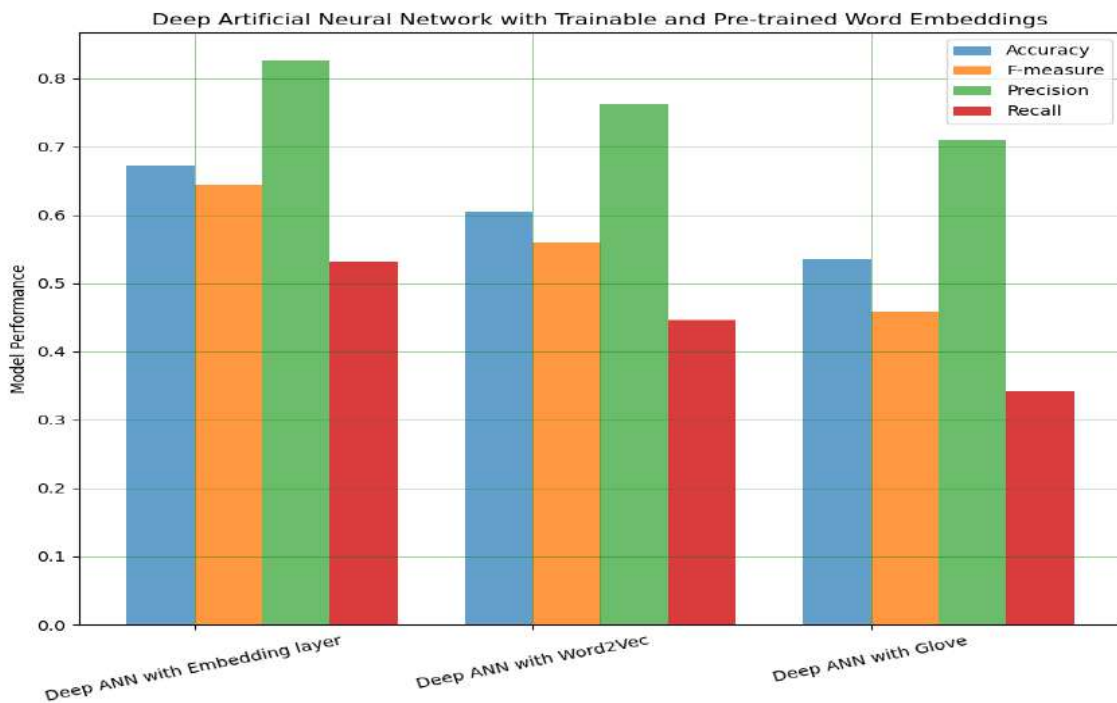


Figure 6. Model Performance Comparison

5 Conclusion

The field of Emotion Analysis is derived from the sentiment analysis domain, where human emotions can be judged from a given piece of text data. People often express their emotions in text on various platforms, such as Twitter (now X) and Facebook. However, this research study analyzes the human emotions from the text dataset. For the analysis, the ArtEmis text dataset containing emotions was first collected from the Art-Emotion website. After preprocessing the

unstructured data, the text was transformed into a structured form using feature representation and transformation methods. Subsequently, Deep Artificial Neural Network models were built. The results were produced by evaluating a deep artificial neural network model with different word embeddings, such as embedding layer, Word2Vec, and GloVe, that were compared using performance metrics such as accuracy, F-measure, recall, and precision. From the comparative results, it was found that Deep ANN with

Table 6. Comparison with Other Related Research studies

Ref.	Algorithms		Techniques		Dataset		Accuracy		Precision		Recall		F-Measure	
	Other Work	This Work	Other Work	This Work	Other Work	This Work	Other Work	This Work	Other Work	This Work	Other Work	This Work	Other Work	This Work
[12]	NB	Deep-ANN	BOW	Emb. layer, Word-2vec, GloVe	ISEAR	Art-Emis	52.- 36%	67.- 36%	-	82.- 62%	-	53.- 27%	-	64.- 50%
[15]	Bi-LSTM	Deep-ANN	Emb. layer	Emb. layer, Word-2Vec, GloVe	Noval text	Art-Emis	64%	67.- 36%	-	82.- 62%	-	53.- 27%	-	64.- 50%
[16]	LSTM	Deep-ANN	Emb. layer	Emb. layer, Word-2Vec, GloVe	ISEAR	Art-Emis	41%	67.- 36%	-	82.- 62%	-	53.- 27%	-	64.- 50%
[45]	SVM	Deep-ANN	infer-sent-GloVe	Emb. layer, Word-2Vec, GloVe	Emo-Twet	Art-Emis	-	67.- 36%	-	82.- 62%	-	53.- 27%	53.- 7%	64.- 50%

an Embedding Layer performed better in terms of accuracy, recall, precision, and F-measure. The resulting developed model obtained 0.6736, 0.5327, 0.8262, and 0.6450 scores for precision, recall, precision, and F-measure, respectively. In contrast, the Deep ANN with GloVe performance was inadequate, resulting in scores of 0.5345 for accuracy, 0.3429 for recall, 0.7102 for precision, and 0.4591 for F-measure.

Author Contributions

Majdah Alvi: Conceptualization, methodology, software implementation, and model development. **Adnan Akhter:** Data collection, data curation, and interpretation of data. **Muhammad Bux Alvi:** Model testing, original draft preparation, investigation, and supervision. **Noor Fatima:** Revising the draft, visualization, and model validation.

Compliance with Ethical Standards

It is declare that all authors don't have any conflict of interest. It is also declare that this article does not contain any studies with human participants or animals performed by any of the authors. Furthermore, informed consent was obtained from all individual participants included in the study.

References

- [1] S. Harguem, Z. Chabani, S. Noaman, M. Amjad, M. B. Alvi, M. Asif, M. H. Mehmood, and A. H. Al-kassem, "Machine learning based prediction of stock exchange on nasdaq 100: a twitter mining approach," in *2022 International Conference on Cyber Resilience (ICCR)*. IEEE, 2022, pp. 01–10.
- [2] K. Jawad, M. Ahmad, M. Alvi, and M. B. Alvi, "Rusas: Roman urdu sentiment analysis system." *Computers, Materials & Continua*, vol. 79, no. 1, 2024.

- [3] P. Ekman, "An argument for basic emotions," *Cognition & emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.
- [4] R. Plutchik, "A general psychoevolutionary theory of emotion," in *Theories of emotion*. Elsevier, 1980, pp. 3–33.
- [5] A. Ortony, G. L. Clore, and A. Collins, *The cognitive structure of emotions*. Cambridge university press, 1990.
- [6] W. G. Parrott, *Emotions in social psychology: Essential readings*. psychology press, 2001.
- [7] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.
- [8] J. A. Russell and A. Mehrabian, "Evidence for a three-factor theory of emotions," *Journal of research in Personality*, vol. 11, no. 3, pp. 273–294, 1977.
- [9] N. Fatima, M. Alvi, and M. Alvi, "Empowering sentiment analysis with deep learning model: Evaluating social media's benefits and drawbacks," *VAWKUM Transactions on Computer Sciences*, vol. 12, pp. 285–297, 12 2024.
- [10] A. Majeed, H. Mujtaba, and M. O. Beg, "Emotion detection in roman urdu text using machine learning," in *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering Workshops*, 2020, pp. 125–130.
- [11] M. Suhasini and B. Srinivasu, "Emotion detection framework for twitter data using supervised classifiers," in *Data Engineering and Communication Technology*. Springer, 2020, pp. 565–576.
- [12] A. F. A. Nasir, E. S. Nee, C. S. Choong, A. S. A. Ghani, A. P. A. Majeed, A. Adam, and M. Furqan, "Text-based emotion prediction system using machine learning approach," in *IOP Conference Series: Materials Science and Engineering*, vol. 769, no. 1. IOP Publishing, 2020, p. 012022.
- [13] N. A. S. Winarsih, C. Supriyanto *et al.*, "Evaluation of classification methods for indonesian text emotion detection," in *2016 International seminar on application for technology of information and communication (ISemantic)*. IEEE, 2016, pp. 130–133.
- [14] X. Zhang and X. Zheng, "Comparison of text sentiment analysis based on machine learning," in *2016 15th international symposium on parallel and distributed computing (ISPDC)*. IEEE, 2016, pp. 230–233.
- [15] J. Zheng, "A novel computer-aided emotion recognition of text method based on wordembedding and bi-ilstm," in *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*. IEEE, 2019, pp. 176–180.
- [16] J. Filipczuk, N. F. Capece, S. Senatore, and U. Erra, "A preliminary investigation of deep emotion-based classification from natural language text," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 3832–3839.
- [17] M. A. Tocoglu, O. Ozturkmenoglu, and A. Alpkocak, "Emotion analysis from turkish tweets using deep neural networks," *IEEE Access*, vol. 7, pp. 183 061–183 069, 2019.
- [18] E. Nugraheni, "Indonesian twitter data pre-processing for the emotion recognition," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE, 2019, pp. 58–63.
- [19] D. Xu, Z. Tian, R. Lai, X. Kong, Z. Tan, and W. Shi, "Deep learning based emotion analysis of microblog texts," *Information Fusion*, vol. 64, pp. 1–11, 2020.
- [20] M. Karna, D. S. Juliet, and R. C. Joy, "Deep learning based text emotion recognition for chatbot applications," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*. IEEE, 2020, pp. 988–993.
- [21] M. B. Alvi, N. A. Mahoto, M. A. Unar, and M. A. Shaikh, "An effective framework for tweet level sentiment classification using recursive text pre-processing approach," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, 2019.
- [22] M. S. Imrona, Widyawan, and L. E. Nugroho, "Pre-processing task for classifying satire in indonesian news headline," in *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, 2020, pp. 176–179.
- [23] J. Camacho-Collados and M. T. Pilehvar, "On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis," *arXiv preprint arXiv:1707.01780*, 2017.
- [24] J. Joseph and D. Jeba, "Information extraction using tokenization and clustering methods," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, pp. 3690–3692, 11 2019.

- [25] M. B. Alvi, N. A. Mahoto, M. S. A. Reshan, M. Unar, M. Elmagzoub, and A. Shaikh, "Count me too: Sentiment analysis of roman sindhi script," *SAGE Open*, vol. 13, no. 3, p. 21582440231197452, 2023.
- [26] D. Khyani, S. B S, N. Niveditha, D. M., and D. Y M, "An interpretation of lemmatization and stemming in natural language processing," *Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technology*, vol. 22, pp. 350–357, 01 2021.
- [27] L. A. Sevastyanov and E. Y. Shchetinin, "On methods for improving the accuracy of multi-class classification on imbalanced data." in *ITMM*, 2020, pp. 70–82.
- [28] K. Fujiwara, Y. Huang, K. Hori, K. Nishioji, M. Kobayashi, M. Kamaguchi, and M. Kano, "Over-and under-sampling approach for extremely imbalanced and small minority data problem in health record analysis," *Frontiers in Public Health*, vol. 8, p. 178, 2020.
- [29] R. Alejo, V. García, and J. H. Pacheco-Sánchez, "An efficient over-sampling approach based on mean square error back-propagation for dealing with the multi-class imbalance problem," *Neural Processing Letters*, vol. 42, no. 3, pp. 603–617, 2015.
- [30] M.-E. Brunet, C. Alkalay-Houlihan, A. Anderson, and R. Zemel, "Understanding the origins of bias in word embeddings," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 09-15 Jun 2019, pp. 803–811. [Online]. Available: <https://proceedings.mlr.press/v97/brunet19a.html>
- [31] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [32] C. Allen and T. Hospedales, "Analogies explained: Towards understanding word embeddings," in *International Conference on Machine Learning*. PMLR, 2019, pp. 223–231.
- [33] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. [Online]. Available: <https://aclanthology.org/D14-1162>
- [34] A. Jain, J. Mao, and K. Mohiuddin, "Artificial neural networks: a tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.
- [35] S. Walczak and N. Cerpa, "Artificial neural networks," in *Encyclopedia of Physical Science and Technology*, third edition ed., R. A. Meyers, Ed. New York: Academic Press, 2003, pp. 631–645. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B0122274105008371>
- [36] R. Qamar and B. Zardari, "Artificial neural networks: An overview," *Mesopotamian Journal of Computer Science*, vol. 2023, pp. 130–139, 08 2023.
- [37] M. Roshan, M. Rawat, K. Aryan, E. Lyakso, A. M. Mekala, and N. Ruban, "Linguistic based emotion analysis using softmax over time attention mechanism," *Plos one*, vol. 19, no. 4, p. e0301336, 2024.
- [38] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2019.
- [39] L. Feng, S. Shu, Z. Lin, F. Lv, L. Li, and B. An, "Can cross entropy loss be robust to label noise?" in *IJCAI*, 2020, pp. 2206–2212.
- [40] A. Rusiecki, "Trimmed categorical cross-entropy for deep learning with label noise," *Electronics Letters*, vol. 55, no. 6, pp. 319–320, 2019.
- [41] A. Jain, A. Fandango, and A. Kapoor, *TensorFlow Machine Learning Projects: Build 13 real-world projects with advanced numerical computations using the Python ecosystem*. Packt Publishing Ltd, 2018.
- [42] C. Zhang, M. Yao, W. Chen, S. Zhang, D. Chen, and Y. Wu, "Gradient descent optimization in deep learning model training based on multistage and method combination strategy," *Security and Communication Networks*, vol. 2021, 2021.
- [43] J. Hernández-Orallo, P. Flach, and C. Ferri Ramírez, "A unified view of performance metrics: Translating threshold choice into expected classification loss," *Journal of Machine Learning Research*, vol. 13, pp. 2813–2869, 2012.
- [44] Y. Liu, Y. Zhou, S. Wen, and C. Tang, "A strategy on selecting performance metrics for classifier evaluation," *International Journal of Mobile Computing and Multimedia Communications (IJCMCMC)*, vol. 6, no. 4, pp. 20–35, 2014.

- [45] Y. K. Shyang and J. L. S. Yan, "A text augmentation approach using similarity measures based on neural sentence embeddings for emotion classification on microblogs," in *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*. IEEE, 2020, pp. 1–6.