Comparative Analysis of Lexical Knowledge based Emotion Recognition Using Classifiers

A Project Report

Submitted by:

AMRIT RAJ (1941012187)
DEBASHISH SAMANTARA (1941012411)
ARNAB KUMAR MANDAL (1941012014)
NILAY KUMAR (1941012183)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHONOLOGY

IN

COMPUTER SCIENCE & INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

Faculty of Engineering and Technology, Institute of Technical Education and Research

SIKSHA 'O' ANUSANDHAN (DEEMED TO BE) UNIVERSITY

Bhubaneswar, Odisha, India
(June 2023)

CERTIFICATE

This is to certify that the project report titled "**Comparative Analysis of Lexical Knowledge based Emotion Recognition Using Classifiers**" being submitted by **Amrit Raj (Section - C), Debashish Samantara (Section - C), Arnab Kumar Mandal (Section - C) and Nilay Kumar (Section - C)** to the Institute of Technical Education and Research, Siksha 'O' Anusandhan (Deemed to be) University, Bhubaneswar for the partial fulfillment for the degree of Bachelor of Technology in Computer Science and Engineering is a record of original confide work carried out by them under my/our supervision and guidance. The project work, in my/our opinion, has reached the requisite standard fulfilling the requirements for the degree of Bachelor of Technology.

The results contained in this project work have not been submitted in part or full to any other University or Institute for the award of any degree or diploma.

Dr. Rasmita Rautray

Department of Computer Science and Information Technology

Faculty of Engineering and Technology;
Institute of Technical Education and Research;
Siksha 'O' Anusandhan (Deemed to be) University

ACKNOWLEDGEMENT

We would like to take this opportunity to express our deepest gratitude to our parents because of their constant encouragement and persistent support throughout our life. We would also like to express our sincere gratitude to our supervisor, Dr. Rasmita Rautray, whose guidance, expertise, and mentorship have been invaluable to our personal and professional development. We are immensely grateful for her support and for providing us with constructive feedback throughout our research. We also like to extend our gratitude to our college, Siksha 'O' Anusandhan for providing us with many opportunities and resources throughout our college life.

Amrit Raj

Akmandal

Debashish Samantara

Nilay Kumar.

Place: Bhubaneswar **Signature of students**

Date: 01/06/2023

DECLARATION

We declare that this written submission represents our ideas in our own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/fact/source in our submission. We understand that any violation of the above will cause for disciplinary action by the University and can also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been taken when needed.

Amrit Raj (1941012187)

AKmandal (1941012014)

Debashish Samantara (1941012411)

Nilay Kumar. (1941012183)

Date: 01/06/2023

REPORT APPROVAL

This project report titled **"Comparative Analysis of Lexical Knowledge based Emotion Recognition Using Classifiers**" submitted by **Amrit Raj (1941012187), Debashish Samantara (1941012411), Arnab Kumar Mandal (1941012014) and Nilay Kumar (1941012183)** is approved for the degree of *Bachelor of Technology in Computer Science & Information Technology*.

Examiner(s)

____

____

____

Supervisor

____

Project Coordinator

____

PREFACE

Expressing emotions is an essential aspect of human behavior that plays a significant role in communication between individuals. With the evolution of society, people have expressed themselves in various forms including literature, gestures, facial expressions, and social media engagement. Automatic emotion detection in digital media is crucial for sentiment analysis and affective computing (AC). This involves discovering emotions in raw data using Natural Language Processing (NLP) techniques. This study applied supervised Machine Learning (ML) and Deep Learning (DL) methods to solve text-based emotion prediction problems using various datasets. The study was conducted using traditional classifiers, such as Support Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT), and XGBoost (XGB), as well as DL-classifiers such as Bidirectional Encoder Representations from Transformers (BERT), Gated Recurrent Unit (GRU), Convolutional Neural Network (CNN), and Bidirectional Long Short-Term Memory (BiLSTM). To enhance the feature selection, various preprocessing techniques were examined on the Twitter emotion detection dataset while building the model. The model results demonstrate the efficacy and promise of several DL- and ML-based classifiers for emotion recognition.

INDIVIDUAL CONTRIBUTIONS

Amrit Raj	Proposed Work, Appendix 1: Ethical Aspect Of Emotion Detection, Conclusions and Future Direction
Debashish Samantara	Results and Discussion, Appendix 2: Data Collection Conclusions and Future Direction
Nilay Kumar	Literature Survey, Conclusions and Future Direction
Arnab Mandal	Introduction, Conclusions and Future Direction

TABLE OF CONTENTS

List of Figures

List of Tables

1. INTRODUCTION

Emotions have long been acknowledged as a crucial component of human communication, impacting our ideas, actions, and interpersonal relationships. They cover a wide range of experiences, such as emotions, bodily reactions, mental processes, and behavioural manifestations. The rise of social media and online communication tools has made the study of emotions and their recognition more important than ever in the modern digital era. Due to the widespread of digital communication, automated emotion recognition has attracted a lot of attention, especially in the context of text analysis. Psychology, marketing, and social media analysis are just a few of the industries that have come to rely heavily on the capacity to recognise and interpret emotions from text. Textual data is a valuable source because it not only delivers information but also represents people's emotional states.

When a social media user tweets about something, it expresses their emotions regarding that particular thing. So to analyze their emotions, we need some kind of system to analyze their tweets and get feedback from that tweet.

For an instance,

Tweet : "*This product is very useful and is available on @Amazon #happyshopping*"

Tweets like this provide valuable insights but handling it manually requires tremendous time and efforts. That is why, a system is required to classify all these automatically and precisely.

Lexicon-based, machine learning (ML)-based, and deep learning (DL)-based methods can all be used to recognise emotions in text. Lexicon-based approaches rely on dictionaries or established emotion lexicons that link particular words to particular emotional states. ML-based strategies involve training models on labelled datasets to learn patterns and characteristics indicative of various emotions. Deep neural networks are used in DL-based techniques to automatically develop hierarchical representations from unstructured text data, capturing intricate patterns and semantic linkages.

Emotion recognition from text using machine learning and deep learning is a promising field of research. These methods could increase the reliability, scalability, and accuracy of emotion identification systems. This could have a positive impact on a number of applications, including marketing, education, and healthcare. The application of machine learning and deep learning for emotion recognition from text may expand our understanding of human emotions. We are able to better understand how emotions affect our ideas, behaviours, and physical responses by understanding how emotions are expressed in writing. We can use this information to enhance our mental health, interpersonal connections, and general wellbeing.

1.1 BACKGROUND STUDY

For humans, analysing the emotions in a tweet may seem like a straightforward process, but when we have to study hundreds of tweets at once, it becomes challenging and time-consuming. Numerous cutting-edge strategies can be used to automate this operation. These methods are computer algorithms that naturally get better with use.

It can be difficult to directly implement computer algorithms for tweet emotion analysis. Users typically use their own language while writing tweets, which can have erratic vocabulary. There is no set format for these tweets.

For an instance, "Hi @amritraj.in, proud to be a part of your #SpreadHappiness campaign. You can join also at https://spreadhappiness.com." A human reader could determine right away that the person tweeting this is happy, but a computer might have trouble classifying it because there are tags, mentions, URLs, and punctuation that don't indicate anything, while there is only one word that expresses happiness. Therefore, in this study, we will examine various techniques for analysing emotion in tweets in order to categorise the emotions.

1.1.1 Lexicon- based techniques

One of the earliest methods for identifying emotions from text was to use lexicon-based algorithms. These methods locate the emotions portrayed in a text by using a vocabulary of emotion words, also referred to as lexicons. Lexicons are collections of words connected to specific feelings. As an instance, the term "happy" frequently indicates the emotion of happiness, whereas the word "sad" frequently indicates the emotion of sadness.

Lexicon-based algorithms operate by looking for words that express emotions in a text. It is more likely that a text communicates a specific emotion if there are more emotion terms present. Lexicon-based algorithms can be excellent at identifying emotions in short texts and are very simple to use. But with words that don't directly relate to emotions can give a wrong result. For instance, depending on the situation, the word "cry" can be used to convey happiness or sadness. Still, Lexicon-based algorithms are a useful tool for extracting emotions from text despite their shortcomings. They can be very useful for identifying emotions in short texts and are simple to utilise and are often used with machine-learning and deep-learning to make more accurate predictions.

1.1.2 Machine-learning based techniques

By outperforming conventional lexicon-based methods, machine learning techniques have revolutionised the field of emotion identification from text. These algorithms can learn subtle patterns and contextual signals that help with precise emotion identification by utilising massive datasets that have been labelled with emotional states. The capability of machine learning techniques to extract intricate connections and linkages from text is one of its main benefits. They are able to pick up on verbal inflections, facetious language, and other contextual factors that affect how emotions are expressed. As a result, they can make more precise and sophisticated emotion predictions. The calibre and variety of the training data, however, have a significant impact on the effectiveness of machine learning techniques. To create reliable and generalizable models, the labelled datasets must be representative of many emotional states, cultures, languages, and domains. Machine learning techniques are constantly developing, meaning that this has significant potential for enhancing text-based emotion recognition.

In conclusion, it has been discovered that machine learning techniques are superior to lexicon-based methods for text-based emotion recognition. Despite continued issues with data quality and processing capacity, developments in the field and current research hold promise for even more precise and reliable emotion identification systems in the future.

1.1.3 Deep-learning based techniques

Emotion recognition from text has significantly advanced through the use of deep learning techniques. These deep neural network-based algorithms demonstrate outstanding ability in discovering subtle patterns and representations within textual input, resulting in very powerful emotion identification models. Deep learning techniques' capacity to automatically extract hierarchical features from text is one of its primary benefits. Multi-layered deep neural networks are capable of gradually learning representations at various levels of abstraction. Due to the models' ability to capture intricate linguistic patterns and semantic links through hierarchical learning, emotional expression can be understood in more detail. Deep learning techniques have shown to be more accurate at recognising emotions than traditional methods for machine learning. Deep neural networks' depth and complexity make it easier to model complicated textual features than it could be for shallower architectures. As a result, they are able to accurately identify emotions by capturing subtle nuances, context-dependent clues, and even long-range dependencies.

Furthermore, compared to conventional machine learning algorithms, deep learning techniques frequently require less manual feature engineering. Deep neural networks don't require explicit feature engineering because they can learn pertinent features straight from the data. This quality, along with the availability of big labelled datasets, has helped them perform well in tasks requiring emotion recognition. In contrast with traditional machine learning algorithms, deep learning techniques frequently require additional computer resources during training. Deep neural networks frequently need a lot of computing and memory resources when being trained, especially when working with large-scale datasets or intricate structures.

In conclusion, deep learning techniques for emotion recognition from text have become more accurate and robust than traditional methods for machine learning. Their success has been greatly aided by their capacity to automatically develop hierarchical representations and deal with noisy input. Although computing demands and data accessibility pose difficulties, continuing research in deep learning approaches holds enormous potential for improving emotion recognition capabilities and understanding the complex nature of human emotions through text.

1.2 MOTIVATIONS

Social media sites have emerged as important platforms for user-generated content, providing a tremendous volume of textual information that captures people's feelings, experiences, and opinions. Analysis of these content can reveal important information about users' emotional states, views on particular subjects, and responses to occasions or goods. Understanding emotions expressed on social media is critical for market research, public opinion analysis, mental health research, and the social sciences in general. Researchers can record emotions via social media because of its dynamic and real-time nature which enables them to track and analyse emotional patterns and changes over time. This can be used to spot new trends, shifting opinions, and emotional reactions to particular situations or social problems. Such information can help with many different decision-making processes, including crisis management and many more. Businesses and organisations can use emotion detection from social media data to understand how customers perceive their products or services. Organisations and governments can monitor public opinion, spot potential issues, and proactively address concerns with the help of emotion detection. The use of emotion detection in online spaces can also provide details about the psychological health of individuals and groups of people. It enables mental health professionals and organizations to offer timely support to the needed ones. Additionally, social media has a large and diversified user base, making it possible to analyse emotions across a range of linguistic, cultural, and demographic contexts. As a result, researchers may analyse how emotions change across cultures, find common emotional experiences, and look at how culture affects emotions. Such understandings can help the development of cross-cultural understanding, the creation of inclusive societies, and the adaptation of emotional support systems for various populations. Also, advancements in machine learning and natural language processing (NLP) methods have made it possible to automatically analyse and categorise emotions from substantial amounts of social media texts. The scope and usefulness of research in this area have increased thanks to the development of scalable and effective emotion detection models made possible by these technologies.

2. LITERATURE SURVEY

In recent years, there have been substantial developments in the field of text-based emotion recognition. Several studies explored different approaches and methods to identify and understand the emotions portrayed in textual data. To increase the accuracy and robustness of emotion recognition, researchers have looked into lexicon-based strategies, machine learning algorithms, and more recently, deep learning models. Studies have also worked on overcoming issues with bias, data quality, and computing efficiency. This literature review attempts to present a thorough overview of the most recent studies in the area of emotion recognition from textual data, highlighting major approaches, findings, and future perspectives.

2.1 EXISTING METHODS

Emotion recognition is a significant area of research in NLP research. It is used in many applications, such as sentiment analysis, emotional computing, and human-computer interaction. Over the past few years, lexical knowledge-based approaches to emotion recognition have received increasing attention. These methods rely on lexicons, databases of words, and expressions associated with specific emotions. These techniques search for words connected to emotions in text to ascertain the user's emotional state. Paul Ekman created a popular model to detect emotions in the text [2]. This model identifies six basic emotions that are natural and common to all people. These include happiness, sadness, anger, fear, surprise, disgust, and contempt. Emotions originated from six different neural systems. Each of these is triggered by how a person experiences a situation. Therefore, emotions are independent [3]. This model is commonly used to detect emotions in text. Many emotion-detection systems use these methods. In addition, machine learning for detecting emotions in texts has become a trend among researchers. The authors of [4] found emotions using phrasal verbs and keywords. The ISEAR data [5] were preprocessed using a keyword-based method, and the results were then produced. A database was constructed to identify phrasal verbs connected to emotion categories.

Phrasal verbs and keywords were recognized as interchangeable with distinct emotions in their databases. However, surpassing the researcher's pre-existing problems, such as a preliminary list of emotion keywords and disregard for word semantics in meaning, achieved a significantly greater accuracy of 65%. The author utilized the ISEAR database to find emotions by preprocessing the data and training it using different classifiers such as logistic regression, K-nearest neighbor (KNN), XG-Boost, and support vector machine (SVM). Finally, logistic regression outperformed all other classifiers. In [6], J. Ranganathan and Tsega Tsahai propose a deep learning-based method for sentiment analysis of tweets. According to the authors, conventional approaches to sentiment analysis often rely on lexical resources or randomly created rules, which can be constrained by their inability to adequately account for linguistic nuances and the context in which the tweets are used. The authors have suggested a method that uses a convolutional neural network (CNN) to learn features from the text of tweets before classifying the tweet sentiment as positive, negative, or neutral using a fully connected layer and softmax layer.

The effect of utilizing various preprocessing methods, such as stemming and stop word removal, on the accuracy of the model was also examined by the authors. Another study by Ameeta Agrawal and Aijun An [7] proposed an approach that uses semantic similarity between words and distinct emotion concepts to determine an emotion vector for each possible affect-bearing word. The authors then used the syntactic dependencies found in the sentence structure to fine-tune the ratings. A thorough analysis of the authors' framework on numerous datasets demonstrates that it is an improved and practical solution to the emotion classification problem, and produces significantly more accurate results than the majority of unsupervised methods of emotion detection. Using pre-trained word representations, the authors of [8], E. Batbaatar, M. Li, and K. H. Ryu presented a unique neural network architecture called the Semantic-Emotion Neural Network (SENN) that can use both semantic and syntactic information. According to the author, the SENN model comprises two sub-networks. The first method uses bidirectional long short-term memory (BiLSTM) to capture contextual information. It focuses on semantic relationships, and the second uses a convolutional neural network (CNN) to extract emotional features and focuses on the emotional connections between words in the text. The authors thoroughly assessed the performance of the proposed model using standard real-world datasets. The authors followed Ekman's six fundamental emotions. The experimental results demonstrate that the proposed model, when combined with several cutting-edge techniques, delivers a noticeably higher quality of emotion recognition.

Paper Title	Author	Year	Applied Model	Evaluation Metric
Automatic emotion detection in text streams by analyzing Twitter data [9]	Maryam Hasan, Elke Rundensteiner & Emmanuel Agu	2019	They used EmotexStream to classify live streams of text messages for real-time emotion tracking.	F1-Score
Emotion Detection Framework for Twitter Data Using Supervised Classifiers [10]	Matla Suhasini & Badugu Srinivasu	2020	Applied ML algorithms like Naive Bayes and KNN to detect the emotion of Twitter messages and then classify the text into four emotional categories.	Accuracy
Emotion Detection of Twitter Posts using Multinomial Naive Bayes [11]	Nazia Anjum Sharupa, Minhaz Rahman, Nasif Alvi, M. Raihan, Tanzil Raihan & Afsana Islam	2020	Used Naive Bayes classifier and applied a well-trained data set for emotion detection.	Accuracy
Computational approaches for emotion detection in the text [12]	Haji Binali, Chen Wu& Vidyasagar Potdar	2010	Applied a hybrid-based architecture for emotion detection and verified its efficiency using SVM.	Accuracy
Emotion Detection from Micro-Blogs Using Novel Input Representation [13]	Fahim Anzum & Marina L. Gavrilova	2023	Applied advanced approach based on Genetic Algorithm.	Precision, Accuracy, F1-Score, Recall
French translation of a dialogue dataset and text-based emotion detection [14]	Pierre-Yves Genest, Laurent-Walter Goix, Yasser Khalafaoui, Előd Egyed-Zsigmond and Nistor Grozavu,	2022	Proposes a Translation-based dataset generation method to detect user's emotion from text using the Needleman-Wunsch algorithm and universal sentence encoder and applied BERT for emotion detection	F1-score
Multi-label emotion classification in texts using transfer learning [15]	Iqra Ameer, Necva Bölücü, Muhammad Hammad Fahim Siddiqui, Burcu Can, Grigori Sidorov and Alexander Gelbukh	2022	Investigated using LSTMs and applied fine-tuned Transformer Networks through Transfer Learning along with a single-attention network and a multiple-attention network for multi-label emotion classification.	Accuracy
Multitasking of sentiment detection and emotion recognition in code-mixed Hinglish data [16]	Soumitra Ghosh, Amit Priyankar, Asif Ekbal, and Pushpak Bhattacharyya	2023	XLMR, a pre-trained cross-lingual embedding model, was used for sentiment detection and emotion recognition.	Accuracy

Table 2.1: Summary of a few more papers related to Emotion Recognition

2.2 PROBLEM STATEMENTS

For the Emotion Classification (EC) problem, we introduce a representation$\langle A, B, C\rangle$. Here, A = [$tweetsdoc^{(1)}$, $tweetsdoc^{(2)}$, $tweetsdoc^{(3)}$, ..., $tweetsdoc^{(u)}$] is a training corpus. The letter 'u' signifies the number of documents in the corpus. B = [$emotion^{(1)}$, $emotion^{(2)}$, $emotion^{(3)}$,……., $emotion^{(u)}$] reflects the set of emotions linked with the corpus's documents, i.e., A. C = [cl_1, cl_2, cl_3,

…, cl_v] where 'v' represents the collection of all classes. Following the execution of the preprocessing task on the corpus texts, the FE technique, such as weighting schemes or word-embedding scheme generated a new set of features. Let us say document tweets doc(p), tf-idf weighting scheme was applied mathematically, which can be expressed as (1):

$$\text{tweetsdoc}^{(p)} = \langle Z_1^{(p)}, Z_2^{(p)}, Z_3^{(p)}, \ldots\ldots, Z_n^{(p)} \rangle \quad (1)$$

Where, $Z_1^{(p)}, Z_2^{(p)}, \ldots\ldots, Z_n^{(p)}$ is the new set of features (i.e., the total number n) that has been generated. Let us assume that document $\text{tweetsdoc}^{(p)}$ is associated with a corresponding document, then mathematically label associated with the multiple classes can be written as (2):

$$\text{emotion}^{(p)} = \langle X\{\text{emotion}_{cl_1}^{(p)}\}, X\{\text{emotion}_{cl_2}^{(p)}\}, .., X\{\text{emotion}_{cl_v}^{(p)}\} \rangle \quad (2)$$

Where X{ } can be calculated as follows (3):

$$X\{\text{emotion}_{cl_q}^{(p)} = \begin{cases} 1, \text{if tweetsdoc}^p \text{ belongs to emotion}_q \\ 0, \text{ if tweetsdoc}^p \text{ doesn't belongs to emotion}_q \end{cases} \quad (3)$$

After the preprocessing task, The FE technique was used to produce a new collection of characteristics, which can be described mathematically as (4):

$$\text{tweetsdoc}^{(k)} = \langle Z_1^{(k)}, Z_2^{(k)}, Z_3^{(k)}, \ldots\ldots, Z_n^{(k)} \rangle \quad (4)$$

The FS technique was then applied to this new collection of characteristics, followed by a classification algorithm. The $\text{emotion}^{(k)}$ associated with the document $\text{tweetsdoc}^{(k)}$ was determined.

2.3 RESEARCH OBJECTIVE

The purpose of this research is to provide a comparative analysis of lexical knowledge-based emotion recognition using classifiers and to compare the performance of several classifiers in classifying emotions from textual data. This study examined the precision, recall, accuracy, and F1 score of each classifier to determine the best performance. This approach may help to identify the most successful classifier for emotion detection tasks and provide ideas for ways to improve the automated recognition of emotion systems.

To fill informational gaps regarding various classifiers for emotion recognition, the following research questions (RQs) were introduced:

RQ1. Can a system based on ML or DL techniques be developed to precisely recognize emotions in short, informal, and unstructured text?

RQ2. How does the performance of the classifiers vary across emotions and sentiments?

RQ3. How does classifier performance change when various text preparation methods, such as stemming and stop word removal, are applied?

RQ4. The classifiers performed significantly differently based on the size of the training dataset?

RQ5. How does the classifier performance compare when implementing several evaluation criteria, such as the average, precision, recall, and F1-score?

This study also reviews existing approaches, models, datasets, vectors, and metrics, and thoroughly analyzes our approach. The objective is to provide insights into the effectiveness of different classifiers in lexical knowledge-based emotion recognition and endorse areas for further research. Therefore, the following contributions were made in this study.

- We evaluated the effectiveness of different ML- and DL-based emotion detection classifiers, such as SVM, Nave Bayes, Decision Tree, XGBoost, BERT, CNN, GRU, and BiLSTM.
- This paper highlighted several preprocessing approaches, such as the removal of mentions, tags, URLs, and many more, as well as the use of tokenization and stemming to locate the roots of words.
- We demonstrate our proposed approach for emotion identification systems using a widely accessible multiclass emotion detection dataset.
- Each classifier were evaluated using performance criteria, such as accuracy, precision, recall, and F1-score.
- This study compares various classifiers for lexical knowledge-based emotion recognition in text, including the effects of diverse lexical resources and feature-selection strategies on their performance.
- Guides for the selection of the most effective classifier and lexical resources for emotion recognition

3. PROPOSED WORK

In this study, a model was designed to precisely detect user emotions from a tweet. A tweet can contain spaces, hyperlinks, hashtags, emojis, and other components, and its character limit is 280 characters [17]. Preprocessing was performed to extract most features from the tweets. Preprocessing includes tokenization, lemmatization, lowercasing, and removal of content that does not add any merit to emotion identification before the linguistic features are extracted from the tweet. The data are then split into tokens. Tokens in a document are represented as vectors of numerical values, where each value corresponds to a term or token in the document. Fig. 3.1 shows the proposed framework for implementation.

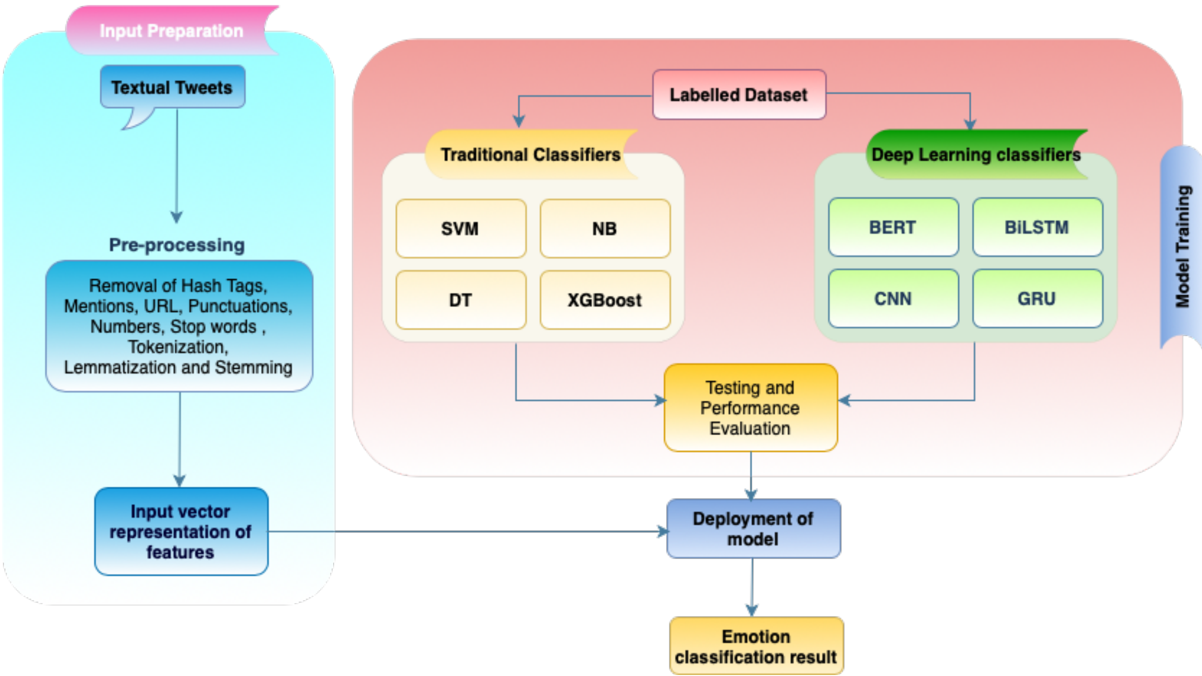

Figure 3.1: Proposed framework for emotion detection

3.1 PROPOSED METHOD

Due to the shortcomings of existing approaches for emotion recognition in text, a fresh approach is presented to improve accuracy and get around problems that traditional machine learning and deep learning techniques run into. This proposed framework also examines the benefits of both strategies while addressing the drawbacks of each. The study also explores different techniques of pre-processing to make data more cleaned and noise-free. The proposed strategy uses a variety of classifiers that include deep learning and machine learning techniques. The system can recognise contextual signals and fine-grained variations in emotional expression by adding domain-specific rules and linguistic patterns. Additionally, complex correlations are captured and generalised from the data using machine learning techniques like Support Vector Machines, Decision Trees, Naive Bayes, and XGBoost or deep learning algorithms like BERT, CNN, GRU, and BiLSTM. To increase accuracy and adaptability, the models are trained on a variety of high-quality labelled datasets. By including clear rules and utilising the data-driven capabilities of machine learning and deep learning, this method improves interpretability as well as adaptability. By addressing the issues of feature engineering, interpretability, and data quality, it provides a more complete and trustworthy solution for emotion identification in text.

3.2 PRE-PROCESSING

Pre-processing is a crucial step in analyzing research data. The raw data were cleaned, filtered, and converted into an appropriate format for analysis during the procedure. The following are some standard preparation procedures followed while working with such data:

1) Data cleaning is necessary once the data has been gathered [18]. This entails eliminating unnecessary data, including URLs, hashtags, mentions, and any unique characters or symbols that can obstruct the analysis.

2) The process of normalizing the text is as follows. All texts must be changed to lowercase, stop words eliminated, and the text stemmed or lemmatized to minimize the number of unique words. Lemmatization involves restoring words to their base form, whereas stemming involves eliminating the suffixes of words.

3) The next step is tokenization, which divides tweets into tokens or single words. This step is crucial because it enables the word-level analysis of tweets.

4) The next step involves generating a matrix of TF-IDF (Term Frequency-Inverse Document Frequency) attributes from a group of tokens. This is a typical pre-processing step in projects involving machine learning, deep learning, and natural language processing. TF-IDF is a statistical technique that assesses the significance of a word in a collection or document. It accounts for the frequency of words within the text (term frequency) and across the entire collection of texts (inverse document frequency).

```python
import nltk
from nltk.stem import WordNetLemmatizer, PorterStemmer
from nltk.corpus import stopwords
import re

def clean_tokenized_lemmatized_stemmed(tweet):
    tweet = tweet.lower()
    tweet = re.sub(r'@\w+', '', tweet)
    tweet = re.sub(r'#\w+', '', tweet)
    tweet = re.sub(r'http\S+|www\S+|https\S+', '', tweet, flags=re.MULTILINE)
    tweet = re.sub(r'[^\w\s]', '', tweet)
    tweet = re.sub(r'\d+', '', tweet)
    tweet = re.sub(r'\s+', ' ', tweet).strip()

    lemmatizer = WordNetLemmatizer()
    stemmer = PorterStemmer()
    stops = stopwords.words('english')

    tokens = nltk.word_tokenize(tweet)
    tokens = [t for t in tokens if not t in stops]
    fintokens = []
    for token in tokens:
        lemmatized_token = lemmatizer.lemmatize(token)
        stemmed_token = stemmer.stem(lemmatized_token)
        fintokens.append(stemmed_token)

    finaltext = " "
    return finaltext.join(fintokens)
```

Figure 3.2: Pre-processing of Raw Tweet Data

Regular Expressions, Lemmatization, Stemming, Stopword Removal and Tokenisation are used in our pre-processing to remove all unwanted things from the raw text sample. For example,

Before Pre-processing: "Hi @debu4master, we like to celebrate your #birthday. Please joined us https://www.meet.com/cse-java-sec"

After Pre-processing: "hi like celebrate please join u"

In the above example, the text sample get lowercased first and then all tags like @debu4master, mentions like #birthday, punctuations, URLs like #https://www.meet.com/cse-java-sec and stopwords like we, to, your are removed from the text sample and all the remaining words are then lemmatized and stemmed to their basic form and then word tokens are generated.

Another example,

Before Pre-processing: "What a joyful day with @amritraj.in #memories !!!"

After Pre-processing: "joy day"

In this tweet, first the sample get lowercased and then @amritraj.in tag and #memories mention as well as !!! punctuations are removed. Stopword like What, a, with are also eliminated and the joyful lemmatized to joy carried out with the remaining tokens of the text sample.

3.2.1 WordNetLemmatizer()

Lemmatization capabilities are offered by the WordNetLemmatizer module of the Natural Language Toolkit (NLTK) library. Lemmatization is the process of reducing words to their lexical or root, or lemma. It seeks to change a word's various inflected forms into a basic form that is used frequently. The WordNetLemmatizer does lemmatization by using WordNet, a lexical database for the English language. WordNet is a massive lexical database that groups words into synsets (sets of synonyms) and offers details on the semantic connections between them.

In order to identify the lemma, the WordNetLemmatizer in NLTK uses a set of predefined criteria based on the part-of-speech (POS) of the word. It automatically thinks that a word's POS tag is a noun (n). Other POS markers, such as verbs (v), adjectives (a), and adverbs (r), can also be handled by this system. The lemmatizer can produce the correct word's base form by providing the POS tag.

By lowering the dimensionality of text input, lemmatization aids in improving the accuracy of subsequent tasks including sentiment analysis, topic modelling, and information retrieval. Because it generates more linguistically sound and meaningful lemmas, it is favoured to stemming.

For an instance, thought $\longrightarrow$ think

When lemmatization happens, “thought” becomes “think”. It can also changes the morphological aspect of the word to get the base or the root form of the word.

3.2.2 Porter Stemming

Martin Porter created the commonly used stemming method known as Porter stemming in 1979. It employs a rule-based algorithm to break down words into their stem or root form. Stemming is the process of stripping suffixes from words to obtain the base or root form, which assists in distilling words down to their basic linguistic components.

To accomplish stemming, the Porter stemming algorithm adheres to a set of guidelines and procedures. Words are subjected to a variety of suffix reduction criteria until an appropriate stem is found. These guidelines are made to deal with the numerous word ends and linguistic patterns found in English. The algorithm uses a measurement known as the "measure of a stem" to ascertain if a rule can be implemented.

Porter stemming is a popular stemming technique for text analysis and information retrieval jobs because it is straightforward and effective. Due to its efficiency and simplicity of use, it is incorporated into well-known natural language processing libraries like NLTK. It is a rule-based algorithm, though, so it might not always create the stems that are the most linguistically precise. Sometimes it can stem words too much or too little, losing information or stemming them incorrectly. Despite that, Porter stemming is useful for jobs like search engines, text mining, and information retrieval systems when reducing words to their simplest forms suffices. In addition to enhancing text processing skills, it offers a rapid and computationally efficient technique to manage word variants.

For example, programming $\longrightarrow$ program

When stemming happens, “programming”, “programmer”, “program” all stemmed down to only “program”. It just removes the inflected endings of words.

3.3 CLASSIFIERS

Different classifiers were used to obtain the best results with better accuracy. A classifier algorithm [19] is a machine learning type used to classify data into different categories or groups. The algorithm is trained on a set of labeled data, where each piece of data is assigned to a category or class, such that the classifier algorithm can use this training data to create a model that can be used to classify new, unlabeled data. Classifier algorithms are used in several fields, including image identification [20], speech recognition [21], natural language processing [22], and financial analysis [23]. They are also used in data mining and predictive analytics to identify patterns and predict outcomes [24]. In our research, two different classification models are used to detect emotions.

3.3.1 TRADITIONAL MACHINE-LEARNING CLASSIFIERS

Traditional classification models are supervised machine learning algorithms that aim to predict the class or category of a given input based on a collection of features. To generate predictions for new and unforeseen data, they first trained a model using a labeled dataset, where each data point had a predefined class label. This approach aims to develop a mapping function that can accurately anticipate the output label from new input data. They have been thoroughly explored and used in various applications ranging from fraud detection and natural language processing to emotion identification and medical diagnosis. In this study, we used four different traditional classifiers for emotion recognition.

A. Support Vector Machine (SVM): A robust machine-learning algorithm is used to classify data into two or more classes [25]. SVM determines the most suitable separating hyperplane between different data classes. A hyperplane is a decision boundary that effectively divides the various data classes. The margin is the interval between the closest data points from each class and hyperplane. The SVM algorithm seeks to minimize the classification error while maximizing the margin between the classes. For the analysis of emotions, the aim was to classify a given text into one of the emotion classes, such as happy, sad, angry, neutral, or many more. SVMs are famous for this task because they can effectively handle high-dimensional feature spaces and nonlinear decision boundaries. The SVM classifier learns a decision boundary that divides the feature vectors denoting various emotions to detect emotions. The emotion category with the highest score was then selected as the predicted emotion based on the output of all SVM classifiers. These are based on findings from the theory of statistical learning, instead of heuristics or comparisons with natural learning systems. The VC dimensions of the classes of the learned functions were used to theoretically guarantee their performance by setting an upper constraint on the generalization error, essentially achieving structural risk minimization. Data that can only be separated using nonlinear rules in the input space are implicitly embedded in SVMs in a high-dimensional feature space. It can be used with geometry and linear algebra to separate data. The learning method is created to take advantage of kernel functions, making it possible to compute inner products in the feature space quickly without the need for explicit embedding. Kernels assume the following form when given a non-linear mapping that embeds input vectors into feature space (5):

$$K(x,z) = \langle \phi(x) \cdot \phi(z) \rangle \tag{5}$$

Where K(x,z) is the kernel function with feature map Ø(x) and Ø(z). According to a hyperplane determined by the type of kernel function utilized, SVM methods divide the training data into a feature space. They identified the hyperplane of the maximal margin, which was determined by adding the hyperplane distances from the closest data points of the two classes. The statistical learning theory states that a hyperplane's generalizability depends only on its margin (which restricts the hyperplane's VC dimension) and not on the dimensionality of the embedding space. SVMs are thus immune to the "curse of dimensionality." The SVM approach trains the nonlinear forms of the following functions (6):

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{l} a_i y_i K(x_i \cdot x) + b \right) \tag{6}$$

where a_i refers to the Lagrange multipliers of the dual-optimization problem. Thus, training this classifier algorithm avoids the issue of local minima and is equivalent to convex quadratic optimization (solving a linearly limited QP problem). It is possible to demonstrate that the optimum solution contains only a subset of I, the support vectors, which originate from the training points closest to the hyperplane. This causes the solution to be sparse and gives rise to effective optimization methods. After obtaining a decision function, one must choose which of the two subspaces described by the separating hyperplane an unclassified example x belongs to and categorize it. Fig. 3.3 illustrates the workings of this classifier.

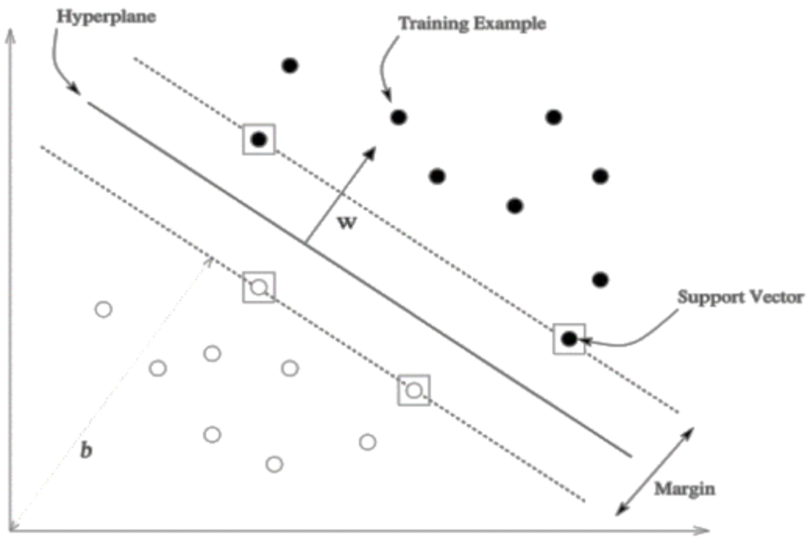

Figure 3.3: Linear maximal margin classifier [26]. The decision function f (x) = sgn((w · x) + b) is defined by the regular vectors w and the bias b of the hyperplane. The margin defines the locations of the support vectors.

B. Naïve Bayes (NB): It is a probabilistic algorithm effective for text-based emotion recognition. The basic principle of naive Bayes is to estimate the probability that a text belongs to a specific emotion class based on its words or attributes [27]. A dataset of text samples labeled with corresponding emotions, such as happy, sad, angry, or neutral, was used to train the Naive Bayes algorithm for emotion identification. The algorithm determined the probability of each emotion class based on the presence of specific words or features in the text during the training phase. The classification of new text samples into their respective emotion classes is then performed using these probabilities. After training, the unlabeled text samples were categorized into appropriate emotion classes using the Naive Bayes method. The algorithm achieves this by calculating the probability of each emotion class given the existence of particular words or textual elements. The text sample was labeled as the emotion class with the highest probability. Nave Bayesian classification is based on Bayes' theorem. In this classification, the probability of a hypothesis is determined by applying Bayes' theorem to observed data. This is an effective strategy for managing high-dimensional and sparse data because it requires that the characteristics of the data tuple are independent of one another. The Bayes' Theorem is refer as (7):

$$P(H/X) = P\left(\frac{X}{H}\right)P(H)/P(X) \tag{7}$$

MNB is a Naive Bayes extension that uses multinomial distributions focusing on how often a word appears in text data to address classification problems. The MNB is represented as (8):

$$P(X \mid c) = \log\frac{N_c}{N} + \sum_{i=1}^{n}\log\frac{t_i+\alpha}{\sum_{i=1}^{n} t_i+\alpha} \tag{8}$$

Where P (X | c) =probability of document X present in class c

N_c = total number of documents present in class c

N = total number of documents

t_i = weight term t

$\sum^{n}_{i=1}$ t = total weight term t present in class c

α = smoothing parameter

C. Decision Tree (DT): It is a machine-learning system that can identify emotions in text [28]. Recursively dividing a dataset into subsets according to the values of specific attributes determines how a decision tree operates. The aim was to produce a tree-like model that can be used to classify new occurrences. Here, a decision tree is trained on a collection of textual data labeled with appropriate emotions in the context of emotion detection. The dataset is then divided into subsets by the decision tree algorithm according to specific characteristics such as the presence or absence of lexical items, length of the text, or use of punctuation. Textual pre-processed content samples can be classified into appropriate emotion classes using a decision tree after training. To accomplish this, the decision tree algorithm begins at the root node of the tree and moves along its branches based on the values of the particular features in the text sample. The label for the text sample was then determined using the emotion label identified at the leaf node that the algorithm reached. For example, a decision tree can initially divide information according to whether the word "happy" is present. A text sample can be categorized as "happy" if the word "happy" appears. Otherwise, the decision tree may go on to a different aspect, such as the text's length or use of exclamation points. The algorithm breaks the dataset into smaller groups until it reaches the leaf nodes of the tree that contain labels for emotions. The decision tree aims to achieve the most significant separation between the classes at each level. For example, Fig. 3.4 illustrates the decision tree framework for four different emotion classes.

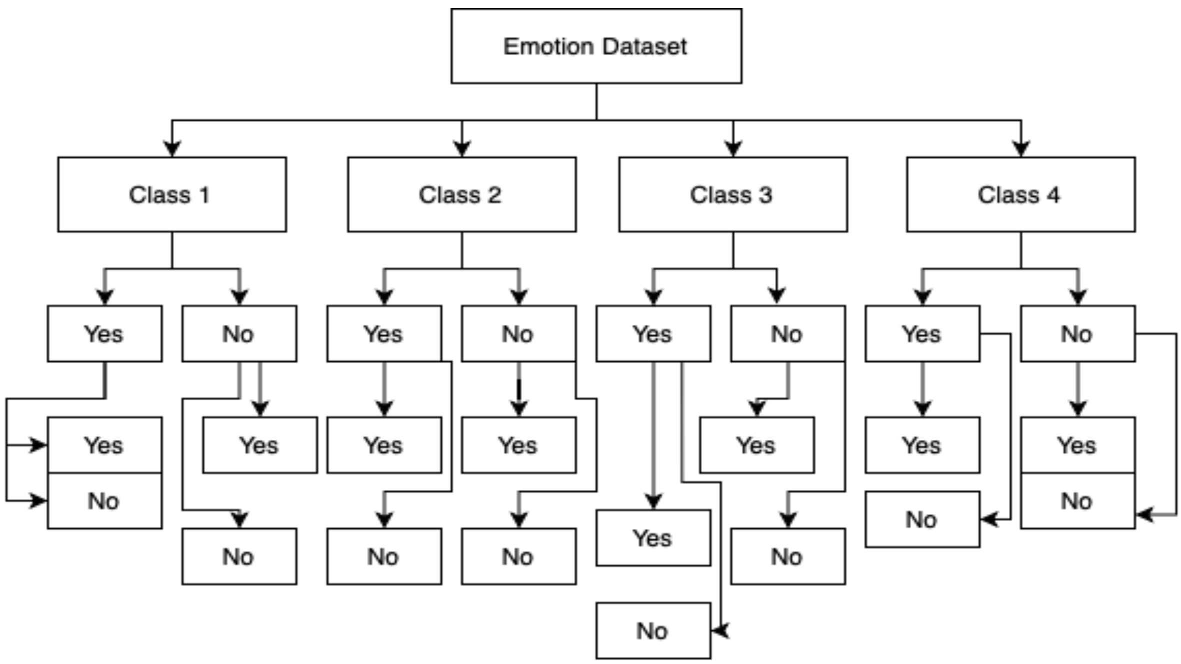

Figure 3.4: Working of Decision Tree (Root to Leaf view).

D. XGBoost (XGB): This is a popular machine learning algorithm that can detect emotions in text. The overall accuracy of the model was increased by combining predictions from various decision trees using an ensemble learning approach [29]. XGBoost was trained on a dataset of text samples labeled with their respective emotions: sad, joy, grief, happy, or neutral. Building decision trees iteratively and adding them to the ensemble determines the operation of the algorithm. The algorithm selects the most valuable features for forecasting emotion labels and applies them to create a new decision tree at each iteration. It uses a technique called "gradient boosting" to adjust the weights of samples in the training dataset. The algorithm calculated the gradients of the loss function for the predicted emotion labels and used them to adjust the weights of the samples. This allows XGBoost to focus on the samples that are most difficult to classify and to improve the model's overall accuracy. Once XGBoost is trained, it can classify new unlabeled text samples into their corresponding emotion classes. By doing this, the algorithm begins at the root node of the ensemble and journeys down the tree branches based on the values of various attributes in the text sample. The effectiveness of XGBoost lies in its scalability, which facilitates rapid memory utilization, and parallel and distributed computations for quick learning. The distributed gradient-boosting library was portable. The objective functions, that is, the loss function and regularization, are represented by (9):

$$F^t \approx \sum_{j=1}^{T}[(\sum_{i\epsilon I_j} g_i)w_j + \frac{1}{2}(\sum_{i\epsilon I_j} h_i + \lambda)w_j^2] + = \rho T \tag{9}$$

Where,

F = objective function

g_i = First derivative of Mean Score Error

w = Score vectors on leaves

h_i = Second derivative of Mean Score Error

λ = Penalty

ρ = Leaf's Complexity

I_j = data samples on Leaf Node j

3.3.2 DEEP-LEARNING CLASSIFIERS

Deep learning models are machine learning techniques that use artificial neural networks to learn from massive amounts of data. These models perform well in tasks, such as emotion recognition, natural language processing, and picture recognition. The primary benefit of deep learning models is their ability to automatically extract features from unprocessed input data, without requiring explicit feature engineering. This is accomplished by building several layers of linked nodes and neurons that gradually acquire more abstract input representations. Although deep learning models require substantial computing resources and must be trained on massive annotated datasets, they have achieved state-of-the-art performance in several applications including object detection, emotion recognition, and language translation. In our study, we used four deep-learning classifiers for emotion detection.

A. BERT: Developed by Google researchers in 2018 for natural language processing tasks, Bidirectional Encoder Representations from Transformers (BERT) is a deep learning model. It implements a pre-training approach that enables comprehension of a word's context by examining words before and after. Therefore, BERT excels in tasks, such as text categorization, question answering, and language interpretation. BERT uses transformer architecture, which is a neural network that handles sequential inputs. The Transformer has a decoder and an encoder. However, BERT only uses an encoder. The model can focus on essential segments of the input sequence because of the encoders' various levels of self-attention. Using two unsupervised learning tasks, masked language modelling and next sentence prediction, BERT's pre-training procedure entails training the model on a sizable corpus of text. In masked language modelling, a sentence with certain words that are randomly omitted is provided to the model, and the missing words must be guessed based on the context of the sentence. When predicting the following sentence, the model is given two sentences and asked to determine if the second sentence continues with the first sentence. BERT is fine-tuned on specific natural language processing tasks after pre-training, such as sentiment analysis, named entity identification, and question answering. Training the model on a smaller, more focused labelled dataset is necessary for fine tuning. On several natural language processing benchmarks, BERT has attained cutting-edge performance. BERT's capacity to deal with the ambiguity and complexity of natural language is one of its primary benefits. BERT is more efficient than traditional rule-based systems for tasks such as emotion analysis and text categorization, because it can comprehend the context and meaning of words. It handles lengthy phrases and paragraphs, and is ideal for document categorization and summarization.

The self-attention process in BERT can be mathematically described as follows. When given an input sequence of length n, X = [x_1, x_2, …, xn], BERT embeds each token, x_i, using an embedding layer in a d-dimensional vector space. The embeddings are then modified using a series of n-layer bidirectional transformer blocks, allowing each token to interact with each other in the input sequence.

At each layer of the transformer block, the BERT calculates a set of d-dimensional query, key, and value vectors, also known as Q, K and V respectively, for each character in the input sequence. The input embeddings are projected linearly to produce these vectors, as illustrated in (10), (11), and (12):

$$Q = Wq * E \tag{10}$$

$$K = Wk * E \tag{11}$$

$$V = Wv * E \tag{12}$$

Where E is the input embedding matrix, Wq, Wk, and Wv are learned linear projection matrices, and * represents matrix multiplication. BERT then computes a set of attention scores, A, between each query and key vector, which determines the relevance of each token in the input sequence with the current token. Attentiveness scores were calculated using (13):

$$A = soft\max(QK^T / \sqrt{d}) \tag{13}$$

Where QK^T is the dot product of the query and key vectors, softmax is the softmax function, and √d is a scaling factor that aids in stabilizing the gradients during training. To determine the context vector for each token in the input sequence, BERT calculates the weighted sum of the value vectors V using the attention score A. This feed-forward neural network produces the final output of the transformer block by using these context vectors as inputs.

B. BiLSTM: Bidirectional Long Short-Term Memory (BiLSTM) is a variant of the classic Long Short-Term Memory (LSTM) architecture. It was specifically developed to capture interdependence in a data sequence by considering past and future contexts [30]. BiLSTM employs two LSTM layers: one that analyzes the input sequence forward, and one that analyzes it backward. BiLSTM models can generate more informed predictions and detect complex patterns in the data by including data from both directions. BiLSTM development is composed of two distinct LSTM layers.

Forward LSTM examines the input sequencing from start to finish, whereas backward LSTM analyzes it from start to end. Each LSTM layer comprises memory cells that retain data over time, and gate cells that regulate the flow of information. The reversed LSTM layer results are then combined or blended to form the final output of the BiLSTM [31]. Fig. 3.5 and 3.6 show the architecture and memory network of the BiLSTM.

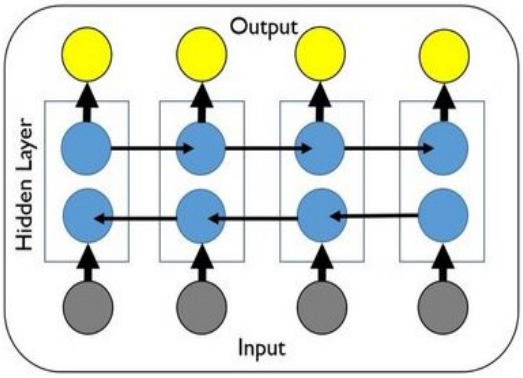

Figure 3.5: BiLSTM Architecture [32]

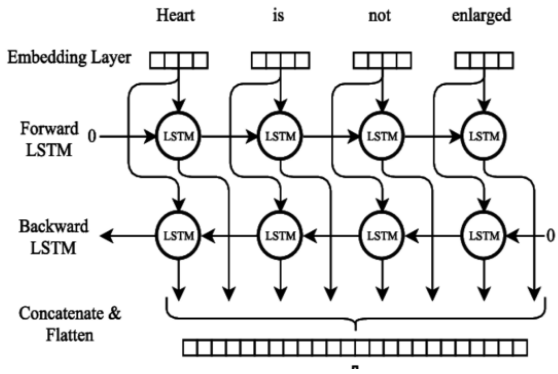

Figure 3.6: BiLSTM Memory Network [33]

Forget gate (14):

$$f_t = \sigma\,(W_f\,.\,[h_{t-1}, x_t] + b_f) \tag{14}$$

Candidate layer (15):

$$i_t = \sigma(W_i\,.\,[h_{t-1}, x_t] + b_i) \tag{15}$$

Hidden state (16):

$$o_t = \sigma(W_o\,.\,[h_{t-1}, x_t] + b_o) \tag{16}$$

Input Gate (17):

$$\widetilde{C_t} = \tanh(W_c\,.\,[h_{t-1}, x_t] + b_c) \tag{17}$$

Output Gate (18):

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{18}$$

Memory state (19):

$$h_t = o_t * \tanh\,(C_t) \tag{19}$$

Where

it = input gate.

ft = forget gate.

ot = output gate.

σ = represents the sigmoid function.

Wx = weight of respective gate(x) neurons.

ht−1 = output of the previous LSTM block (at timestamp t − 1).

xt = input at the current timestamp.

bx = biases for the respective gates(x).

Ct = cell state (memory) at timestamp (t)

C˜t = represents a candidate cell state at timestamp (t).

C. GRU: The Gated recurrent unit (GRU) is a form of recurrent neural network (RNN) developed by Cho et al. in 2014. It is intended to be a more accessible variant of LSTM, or long short-term memory, that can perform similarly in sequence modeling tasks [34]. The GRU cell has two gates, the updating gate and reset entrance, which control the movement of knowledge within the network. The GRU gates have the following equations:

Updated gate (20):

$$z_1 = \sigma(W_z \,.\, [h_{t-1}, x_t]) \tag{20}$$

Reset gate (21):

$$r_t = \sigma(W_r \,.\, [h_{t-1}, x_t]) \tag{21}$$

New memory (22):

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t} \tag{22}$$

Final memory (23):

$$\widetilde{h_t} = \tanh(W \,.\, [r_1 * h_{t-1},\ x_t]) \tag{23}$$

Where,

W = weight vector.

$*$ = element-wise multiplication.

σ = sigmoid function.

The update gate determines how much of the prior concealed state is maintained, whereas the reset gate determines how much of the new input is being absorbed. The update gate determines the final hidden state, which is a weighted blend of the previously unknown and future hidden states. GRU have been proven to be helpful for various sequence simulation programs, including translation by machine, speech recognition, and image captioning. It outperformed LSTM in terms of computing power and ease of training. Fig. 3.7 shows the internal structure of the GRU.

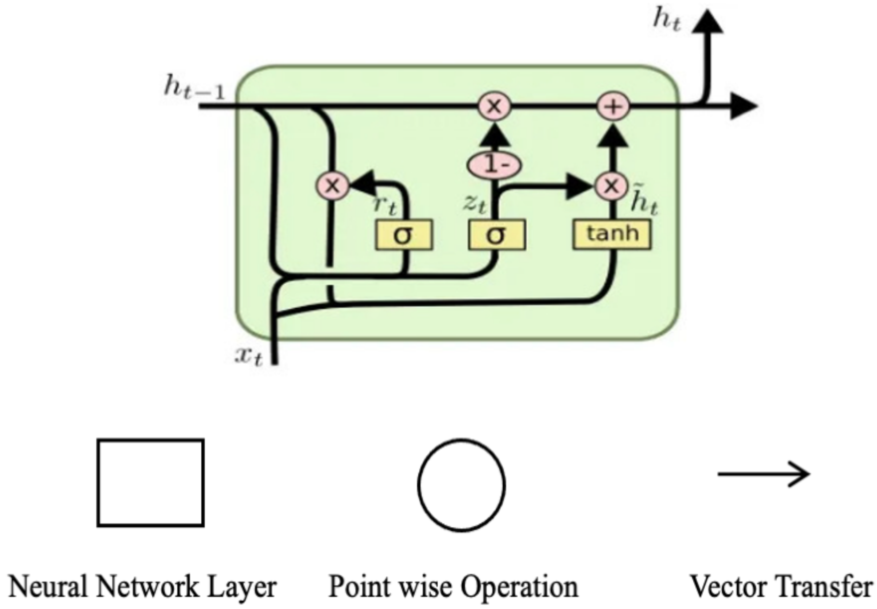

Figure 3.7: Internal structure of a Gated Recurrent Unit (GRU) [35]

D. CNN: CNNs, or convolutional neural networks, are a type of deep neural network that is extensively used for image categorization, object recognition, and other computer vision tasks. LeCun et al. first described them in their landmark 1998 study, "Gradient-based learning applied to document recognition" [36]. CNNs comprise multiple convolution layers, beginning with one or more layers across the input image and computing the dot product at each location. Each convolution layer creates an accumulation of activation maps that represent every filter response at each position in the input. Equation (24) represents the calculation of the final feature map.

$$G[m,n] = (f * h)[m,n] = \sum_j \sum_k h[j,k]\, f[m-j, n-k] \qquad (24)$$

Where,

G[m, n] = output value at location (m, n).

f[m, n] = input value at location (m, n).

h[j, k] = weight of filter at location (j, k).

In practice, CNNs employ several filters in each convolutional layer to acquire a wide range of knowledge. The output of each filter is sent through an irregular activated function, such as ReLU or sigmoid, which brings nonlinearity into the network and helps it to model complicated relations between the input and output. The output is typically flattened and sent into one or more fully connected layers, which compute a weighted sum of the inputs and run completely interconnected layers. The convolutional layers extract features at various spatial scales by applying a series of learnable filters to the input image. Each filter was a petite vector "slid" via a nonlinear activation function. The results of the final layer that are fully connected are the anticipated classes or labels for the input image. CNNs have changed computer vision by delivering leading-edge outcomes to numerous picture classification and object identification problems. They have also been used in natural language processing, speech recognition, and acoustic signal processing.

3.4 EVALUATION CRITERIA

This section highlights several evaluation criteria for assessing the effectiveness of an emotion-detection system. The most commonly used criteria include accuracy, recall, precision, and the F1-score. In machine learning, terms such as "true positives, false positives, true negatives," and "false negatives " are used to evaluate the performance of classification models. True positives (TP) are cases in which the model accurately detects a positive case, which means it correctly indicates the presence of a specific emotion in the input data. False positives (FP) are cases where the model mispredicts a specific emotion in the input data, leading to inaccurate identification of a positive case. True negative (TN) cases are those in which the model accurately identifies a negative case, that is, when it predicts that a specific emotion will not be present in the input data. False negatives (FN) are cases where the model incorrectly detects a negative case, which means that it incorrectly predicts that a specific emotion is absent from the input data. The accuracy, precision, recall, and F1-score evaluation metrics were calculated using these terms. Precision and recall measure the ability of the model to accurately identify positive and negative cases, respectively, whereas accuracy measures the accuracy of the model's predictions. The accuracy of a model that predicts the correct emotion for a given input was measured based on its accuracy. The recall measures the odds of significant events that a model correctly detects. Recall describes how well a model detects specific emotions. Precision is the percentage of relevant examples among the total number of instances predicted by the model. Precision in emotion detection refers to the frequency with which a model accurately predicts a given emotion. Precision is functional when the goal is to avoid misidentifying the emotions.

High precision can result in low recall, implying that some instances of a particular emotion may go undetected. The F1-score is calculated as the harmonic mean of precision and recall, yielding a single result that considers both precision and recall. A better value denotes better performance and ranges from zero to one. A high F1 score in the context of emotion recognition suggested that the model effectively recognized various emotions in text data. Conversely, a low F1 score indicated that the model must operate at a higher level. Therefore, a sound emotion detection system conveys high accuracy, recall, and precision [37]. The accuracy, precision, recall, and F1-score were computed using (25), (26), (27), and (28), respectively.

$$\text{Accuracy} = \frac{\text{Number of accurately categorized predictions}}{\text{Total number of the predictions}} \tag{25}$$

$$\text{Precision} = \frac{\text{True Positive(TP)}}{\text{True Positive(TP)+False Positive(FP)}} \tag{26}$$

$$\text{Recall} = \frac{\text{True Positive(TP)}}{\text{True Positive(TP)+False Negative(FN)}} \tag{27}$$

$$\text{F1} - \text{score} = 2 * \frac{\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}} \tag{28}$$

3.5 DATASET DESCRIPTION

In this study, three distinct datasets that are focused on the emotional content of Twitter are used. These publicly accessible statistics offer significant knowledge of various emotions that users demonstrate on social media platforms. These datasets are used to train and evaluate a variety of machine-learning and deep-learning models for emotion classification:

3.5.1 Dataset 1 (SemEval2018)

Dataset 1 contains 74792 samples, each of which is assigned one of the 13 different emotion classes: sadness, joy, neutral, worry, surprise, fear, happiness, anger, love, fun, relief, hate, disgust, empty, enthusiasm, boredom, and shame. It has two columns, emotion and content. Fig. 3.8 represents the distribution of different classes of emotions in the dataset.

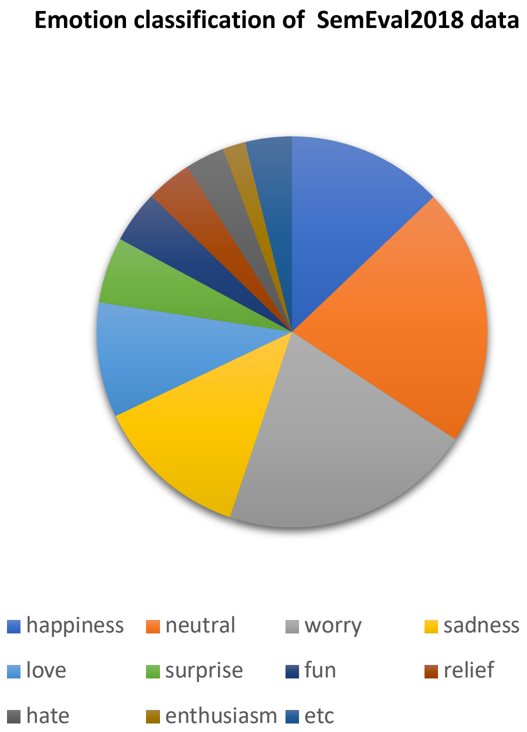

Figure 3.8: Emotion data classification in the Dataset 1

3.5.2 Dataset 2 (Jcharis Collection of Tweet)

In this dataset, it contains 34791 samples, each of which is labeled with one of the eight emotion classes: joy, sadness, fear, anger, surprise, neutral, disgust, and shame. It comprises two columns: Text and Emotion. Fig. 3.9 illustrates the different emotion classes in the dataset.

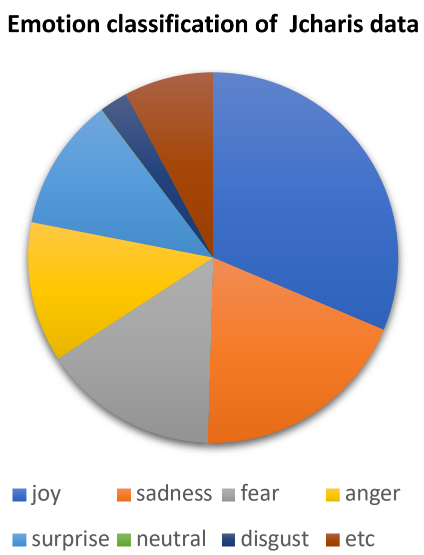

Figure 3.9: Emotion data classification in the Dataset 2

3.5.3 Dataset 3 (Dair-Ai Emotion)

In the third dataset, we have 20000 samples, each of which is tagged with one of six different emotion classes: joy, sadness, fear, anger, surprise, neutral, and love. It contains two columns, text and labels. Fig. 3.10 shows the emotion class distribution of the dataset.

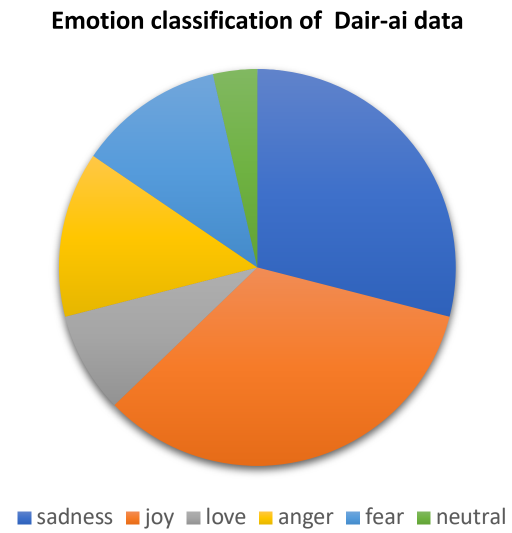

Figure 3.10: Emotion data classification in the Dataset 3

Emotion	Distribution in Dataset 1	Emotion	Distribution in Dataset 2	Emotion	Distribution in Dataset 3
happiness	13%	joy	31.7%	sadness	29%
sadness	12.9%	sadness	19.3%	joy	33.8%
fun	4.4%	fear	15.5%	love	8.2%
neutral	21.6%	anger	12.4%	anger	13.5%
love	9.6%	surprise	11.7%	fear	11.9%
relief	3.8%	neutral	6.5%	neutral	3.6%
worry	21.1%	disgust	2.5%		
surprise	5.5%	others	7.9%		
hate	3.3%				
others	3.9%				

Table 3.1: Emotion Class Distribution in different Datasets

4. RESULT AND DISCUSSION

The results and analysis of the experiments and assessments that were conducted are presented in this chapter. This section seeks to explore the results from the implemented emotion detection models and offer a thorough analysis of their performance.

The accuracy, precision, recall, and F1 score are the main results that are being stated. These metrics are used to evaluate how well the emotion detection system is working. The paper also explores the advantages, disadvantages, and restrictions of the used models, showing how well they perform across various scenarios and datasets. Additionally, this section also explores the reasons behind of the results, looking at the influence of several factors like the feature selection, preprocessing methods, and the quality and quantity of training data. It also looks into any observed differences between actual and expected emotions, highlighting potential problems and possibilities for development.

This section intends to advance emotion detection methods, promote field discussion, and offer insightful information for future study and development in the area of emotion recognition through the analysis and interpretation of the results.

4.1 EXPERIMENTAL SETUP

This section discusses the optimal configuration used to conduct the experiments and evaluations. It also includes details on the software frameworks, libraries, and hardware resources utilized during the experiments.

Operating System:	Windows 11 / MacOS Ventura
System Manufacturer:	HP Inc. / Apple Inc.
Processor:	i5 8^{th} gen Intel Core @ 1.4Ghz
Memory:	8 GB 2133MHz LPDDR3
GPU:	NVIDIA GTX1650 / Intel Iris Plus Graphics 645

Models are built in Python 3.11.0 environment. List of several library used in our model are given in table 4.1.

Package Name	Version
nltk	3.8.1
pandas	2.0.0
sklearn	0.19.2
Tensorflow	2.12.0
Ktrain	0.36.0
numpy	1.23.5
transformers	4.28.1

Table 4.1: Python Packages

4.2 PERFORMANCE EVALUATION AND RESULT ANALYSIS

The experiments focused on assessing the extraction of language-based features using TF-IDF vectorization, a popular NLP technique utilized in opinion-mining research. Moreover, this study sought to ascertain whether merging the extracted linguistic, stylistic, and sentiment elements would improve the model's performance. Traditional ML and DL models were trained using various input representations, and the XGBoost, SVM, and BERT classifiers performed well. The primary goals of these experiments were to analyze the importance of various feature types and gain a deeper understanding of the classification models. TF-IDF was chosen as the preferred technique for extracting linguistic patterns from the dataset because it has been deemed the most effective [38]. TF-IDF considers the significance of each phrase in the tweets that convey emotions.

In contrast, the BoW representation merely assesses whether a term is present in a corpus and assigns each term equal weight [39]. Consequently, it was discovered that emotion recognition models trained using BoW representation outperformed those utilizing the TF-IDF input representation. Therefore, TF-IDF can develop efficient representations without requiring a large dataset. Additionally, capturing the significance of each word rather than the connections between them is more important for tasks involving emotion recognition. Based on our experimental results, the TF-IDF input format was used to train the emotion detection models more successfully. The available datasets were noisy and required preprocessing such as tokenization, removing stop words, and lemmatization [40].

We used regular expressions to remove @mentions, hashtags, URLs, punctuation, numbers, and extra whitespaces. Then, the dataset was divided into 80:20 for training and testing. To answer RQ1, "Can a system based on ML or DL techniques be developed to precisely recognize emotions in short, informal, and unstructured text?" ,We developed four different traditional ML models and four different DL models using the input representations, and found that XGBoost, Naive Bayes, Support Vector Machine, Decision Tree, BERT, GRU, CNN, and BiLSTM fared well. Using a pipeline, we orchestrate the data flow into the TF-IDF vectorizer and then perform XGB, NB, SVM, DT, BERT, GRU, CNN, and BiLSTM. Here, an array data structure is used to store the collection of words. Each word was stored in a separate element of the array. Arrays of words can be used for various tasks such as spell-checking, text analysis, machine translation, and text generation. This is a powerful tool for NLP. For BERT, BiLSTM, CNN, and GRU, we used sparse categorical cross-entropy. It is a loss function used in machine learning to train models to classify data into multiple categories. It was used to train the model to predict the correct label for each word in a sequence. The loss function was calculated by taking the negative log-likelihood of the predicted labels. It is efficient and effective and can be used to train models on large datasets. In this study, we also used the Adam optimizer to train the deep learning models. The main objectives of these experiments were to determine the importance of various feature types independently and to understand categorization methods.

Tables 4.2, 4.3, and 4.4 compare the performances of different classifiers based on the metrics of precision, recall, and F1-score. These metrics provide an objective way to compare the effectiveness of classifiers in different scenarios and also provide answers to 2 different research questions RQ2, "How does the performance of the classifiers vary across different emotions and sentiments?" and RQ5, "How does the classifier performance compare when implementing several evaluation criteria, such as the average, precision, recall, and F1-score?".

To respond to RQ3,"How does classifier performance change when various text preparation methods, such as stemming and stop word removal, are applied?", we discovered that stemming and stop word removal can affect the performance of machine-learning classifiers. We can decrease the dimensionality of the feature space by performing stemming and stop word removal, which also enhances the performance of classifiers by utilizing the word frequency or co-occurrence features. The size of the training dataset can significantly affect the effectiveness of machine learning

classifiers. More training data can result in better performance because the classifier can generalize to new, unknown data more effectively, and has more instances to learn. However, the impact of the size of the training dataset on performance can vary depending on several variables, including the difficulty of the classification task, accuracy of the data, and type of classifier used. Thus, answering the RQ4, "The classifiers performed significantly differently based on the size of the training dataset?".
Regarding accuracy, Fig. 4.1 illustrates that Traditional Machine-Learning approaches, such as SVM and XGBoost classifiers, perform similarly when identifying emotions, suggesting that these approaches are effective at identifying essential patterns in the data and obtaining high accuracy. Similarly, Fig. 4.2 illustrates that deep-learning approaches such as BERT and GRU outperform other classifiers in precisely identifying emotions.

However, it is crucial to remember that depending on the specifics of the dataset and the classification problem, SVM, XGBoost, BERT, and GRU have varying strengths and drawbacks that can make one technique more appropriate.

	Classifier	Precision	Recall	F1-Score
Traditional classifier	SVM	0.48	0.42	0.42
	DT	0.40	0.36	0.38
	XGB	0.49	0.41	0.45
	NB	0.37	0.31	0.30
Deep Learning classifier	BERT	0.39	0.43	0.40
	BiLSTM	0.37	0.30	0.31
	CNN	0.22	0.19	0.18
	GRU	0.31	0.30	0.30

Table 4.2: Performance comparison of classifiers w.r.t Dataset 1

	Classifier	Precision	Recall	F1-Score
Traditional classifier	SVM	0.68	0.61	0.60
	DT	0.56	0.52	0.52
	XGB	0.70	0.50	0.54
	NB	0.52	0.36	0.38
Deep Learning classifier	BERT	0.66	0.67	0.66
	BiLSTM	0.59	0.58	0.59
	CNN	0.44	0.45	0.44
	GRU	0.66	0.60	0.62

Table 4.3: Performance comparison of classifiers w.r.t Dataset 2

	Classifier	Precision	Recall	F1-Score
Traditional classifier	SVM	0.85	0.86	0.84
	DT	0.80	0.82	0.80
	XGB	0.84	0.86	0.84
	NB	0.80	0.51	0.53
Deep Learning classifier	BERT	0.93	0.94	0.93
	BiLSTM	0.85	0.88	0.86
	CNN	0.78	0.80	0.78
	GRU	0.88	0.89	0.88

Table 4.4: Performance comparison of classifiers w.r.t Dataset 3

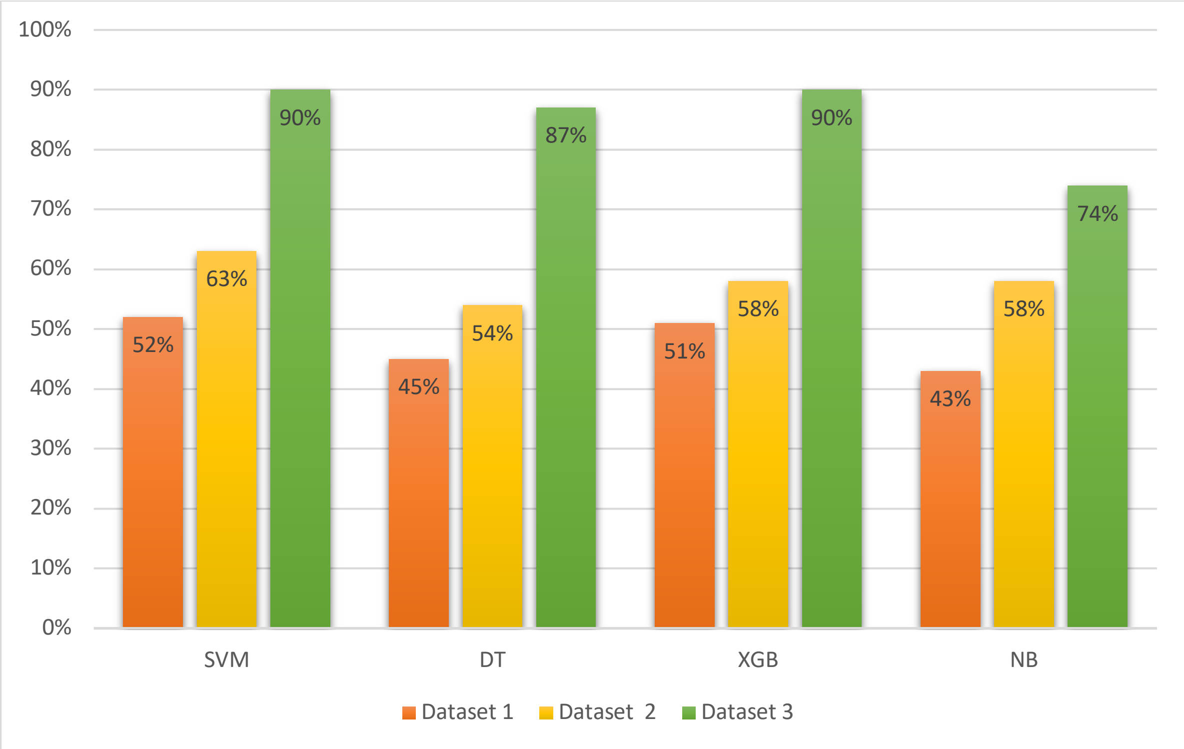

Figure 4.1: Accuracy analysis of different ML classifiers w.r.t three different datasets

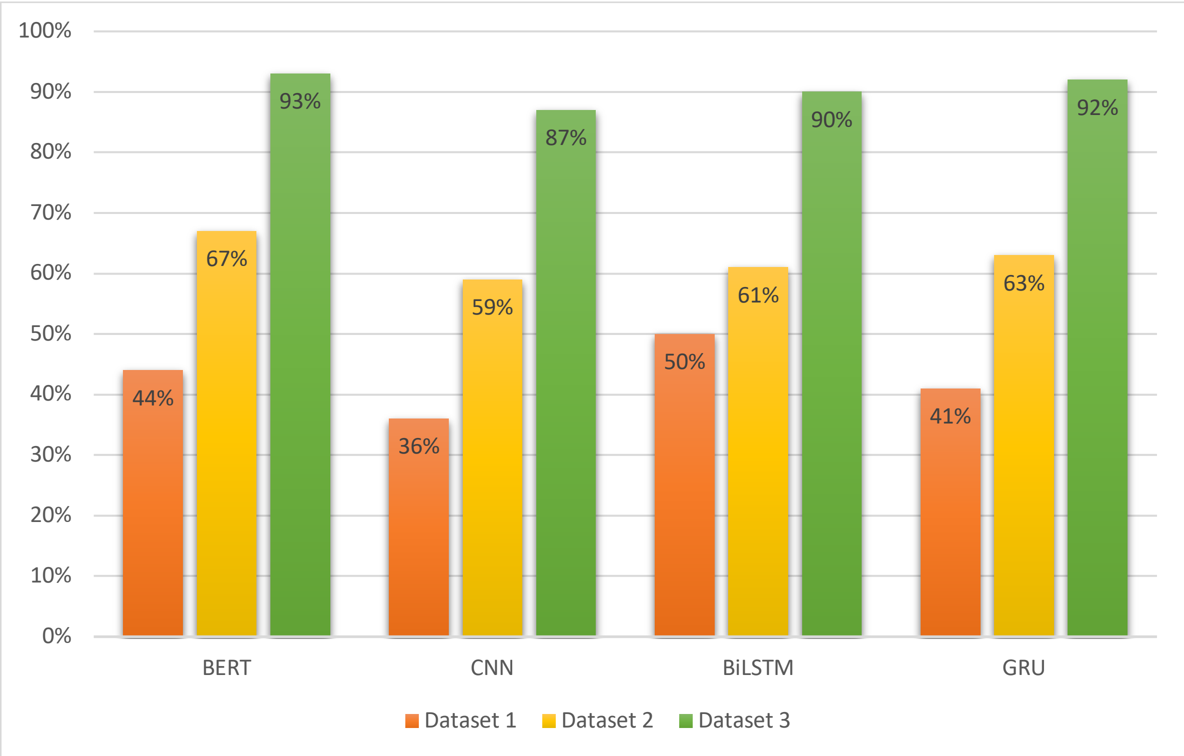

Figure 4.2: Accuracy analysis of different DL classifiers w.r.t three different datasets

4.3 COMPARISON TO BASELINES

This subsection compares the top results of the proposed emotion detection and recognition model with the performances of various state-of-the-art summarizers. These approaches are briefly discussed below.

1) Ahmet Mert and Aydin Akan [41] adopted a multivariate synchrosqueezing transform (MSST) method for emotion recognition and feature extraction that can obtain correct labeled EEG signals from pictures, music, and video clips.

2) Tian Chen et al. [42] proposed an electroencephalogram (EEG) emotion recognition technique to implement the sentiment classification on the discrete emotion model and dimensional emotion model based on the LIBSUM classifier.

3) Locally robust feature selection (LRFS) [43] is a method that determines the generalized features of EEG along with inter-individual consistency evaluation, fragmented dimension, and high-frequency crossings extracted to indicate emotion fluctuations.

4) Emotion Dataset (ED) [44] is a dataset used to investigate physiological responses as a basis for emotion identification. To capture the emotional state and enable a versatile analysis of a participant from physiological and facial expressions, this dataset considers nine discrete emotions and three affective dimensions.

5) Leave-One-Out Cross-Validation (LOOCV) [45] is a technique introduced with SVM classifier to study emotion states based on the level of valence-arousal score. Finding multidomain features from the wavelet, time, and frequency was the primary approach of this study.

6) Cross-lingual multitask model (XLMR) [16] is a transformer-based multitask framework for sentiment detection and emotion recognition that uses the SentiMix code-mixed dataset. The framework's efficient transfer-learning technique benefited from the opportunity to offer significant resources to the community of Hindi-English speakers with mixed emotions.

7) Hande Aka Uyamez and Senem Kumova Metin [46] applied a lexicon-based method to identify emotion and sentiment. Discrimination of several emotions has been achieved based on category, dimension, and conversion to numerical form.

8) The Feature Selection Scheme (FSS) [47] was proposed for term selection based on frequent terms considering the relevance scores. The proposed system additionally employs a bag-of-words (BOW) approach to construct vectors for document representation, where each selected term is assigned a weight of 1 if it exists or assigned a weight of 0 if it does not exist. The benchmark dataset used in this experiment was ISEAR.

9) XLnet Multiple Attention (XLnet-MA) [15] is a model proposed in this study to identify multiple emotions from brief text. This model reduces the impact of irrelevant features while preserving the emotional information.

10) The Pyramid Attention Network (PAN) [48] is the suggested model employed in this study for emotion detection in microblogs. This approach allows the evaluation of sentences from different perspectives to capture multiple emotions in a given text. The ekphrasis tool was introduced here for tokenization, word normalization, spell correction, and hashtags as preprocessing steps.

Article Title & Reference	Year	Dataset	Classification method	No. of classes	Discrete emotions	Accuracy (%)
Emotion recognition based on the time-frequency distribution of EEG signals using multivariate synchrosqueezing transforms [36]	2018	DEAP	ANN	3	NO	82.11
EEG emotion recognition model based on the LIBSVM classifier [37]	2020	DEAP	SVM	2	NO	74.88
Locally robust EEG feature selection for individual-independent emotion recognition [38]	2020	MAHNOB-HCI	LSSVM	2	NO	65
Emognition dataset: emotion recognition with self-reports, facial expressions, and physiology using wearables [39]	2022	Emognition,	rmANOVA	3	YES	70

Multi-domain feature fusion for emotion classification using the DEAP dataset [40]	2021	DEAP	SVM	3	YES	65.92
Multitasking of sentiment detection and emotion recognition in code–mixed Hinglish data [41]	2023	SentiMix	XLMR	3	YES	66.03
Vector-based sentiment and emotion analysis from the text: A survey [42]	2022	ISEAR	Random Forest	2	YES	84.3
Vector-based sentiment and emotion analysis from the text: A survey [42]	2022	EmoBank	Random Forest	2	YES	82.8
A new feature selection scheme for emotion recognition from text [43]	2020	ISEAR	SVM with linear kernel	2	YES	87.9
Multi-label emotion classification in texts using transfer learning [44]	2023	SemEval-2018	LSTM	2	YES	62.4
Gated Recurrent Neural Network Approach for Multi-label Emotion Detection in Microblogs [45]	2019	SemEval 2018 Task1	LSTM	3	YES	58.9
Proposed BERT Model		Dair-Ai Emotion (DTS3)	GRU	6	YES	93

Table 4.5: Performance comparison of the Proposed Model with the State-Of-The-Art methods

5. CONCLUSIONS AND FUTURE DIRECTION

This study aims to explore the effectiveness of various lexical knowledge-based classifiers for emotion recognition. The research utilizes a dataset comprising text documents that have been tagged with emotion labels. The performance of traditional classifiers, including Naive Bayes (NB), XGBoost, Support Vector Machines (SVM), and Decision Trees (DT), is compared with deep learning classifiers, namely BERT, CNN, GRU, and BiLSTM.

The importance of feature selection is one of the study's main focal points. It implies that lexical features like n-grams and word embeddings can be added to classifiers to enhance their performance. This emphasises how crucial it is to take into account various language factors and how they affect the recognition of emotions.

Preprocessing and careful feature selection and combination are emphasized as critical steps in achieving optimal results. These steps ensure that the input data is appropriately transformed and that relevant features are extracted. The study provides valuable insights into the effectiveness of different classifiers in utilizing lexical knowledge for accurate emotion recognition.

The evaluation of the classifiers' performance using various datasets serves as a benchmark for this research. The experimental results shed light on the accuracy achieved by different classifiers, allowing for a comparative analysis of their capabilities in emotion detection tasks.

Furthermore, the findings of this study have broader implications beyond emotion recognition alone. The insights gained from this research can be extended to applications such as sentiment analysis, social media monitoring, and mental health analysis. By understanding and accurately detecting emotions in text data, businesses and organizations can improve their understanding of customer feedback, identify trends, and gain valuable insights for enhancing product development and marketing strategies.

The study also highlights the relevance of emotion detection in social media. Analyzing emotions expressed on platforms like social media can help businesses better comprehend customer sentiments, enhance customer service experiences, and make data-driven decisions to meet customer expectations effectively.

REFERENCES

[1] E. Vainik, Lexical knowledge of emotions: the Estonian emotion vocabulary's structure, variability, and semantics. Tartu University Press, 2004.

[2] P. Ekman, “Basic emotions”. In T. Dalgleish & M. J. Power (Eds.), Handbook of cognition and emotion (pp. 45–60). John Wiley & Sons Ltd, 1999.

[3] J. A. Russell, "Measures of emotion." The measurement of emotions. Academic Press, pp. 83-111, 1989.

[4] D. Seal, U. K. Roy, and R. Basak, "Sentence-level emotion detection from text based on semantic rules." In Information and Communication Technology for Sustainable Development: Proceedings of ICT4SD 2018, pp. 423-430. Springer Singapore, 2020.

[5] A. A. Alnuaim, M. Zakariah, P. K. Shukla, A. Alhadlaq, W. A. Hatamleh, H. Tarazi, R. Sureshbabu, and R. Ratna, "Human-computer interaction for recognizing speech emotions using multilayer perceptron classifier," J. Healthcare Eng., vol. 2022, pp. 1–12, Mar. 2022.

[6] J. Ranganathan, and T. Tsahai, “Sentiment Analysis of Tweets Using Deep Learning.” In: Chen, W., Yao, L., Cai, T., Pan, S., Shen, T., Li, X. (eds) Advanced Data Mining and Applications. ADMA 2022. Lecture Notes in Computer Science, vol 13725. Springer, Cham, 2022.

[7] A. Agrawal and A. An, "Unsupervised Emotion Detection from Text Using Semantic and Syntactic Relations," 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, Macau, China, 2012, pp. 346-353, doi: 10.1109/WI-IAT.2012.170.

[8] E. Batbaatar, M. Li and K. H. Ryu, "Semantic-Emotion Neural Network for Emotion Recognition From Text," in IEEE Access, vol. 7, pp. 111866-111878, 2019, doi: 10.1109/ACCESS.2019.2934529.

[9] M. Hasan, E. Rundensteiner, and E. Agu, “Automatic emotion detection in text streams by analyzing Twitter data.” Int J Data Sci Anal, vol. 7, pp. 35–51, 2019.

[10] M. Suhasini, and B. Srinivasu, Emotion Detection Framework for Twitter Data Using Supervised Classifiers. In: Raju, K., Senkerik, R., Lanka, S., Rajagopal, V. (eds) Data Engineering and Communication Technology. Advances in Intelligent Systems and Computing, vol 1079. Springer, Singapore, 2020.

[11] N. A. Sharupa, M. Rahman, N. Alvi, M. Raihan, A. Islam and T. Raihan, "Emotion Detection of Twitter Post using Multinomial Naive Bayes," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020.

[12] H. Binali, C. Wu and V. Potdar, "Computational approaches for emotion detection in the text," 4th IEEE International Conference on Digital Ecosystems and Technologies, Dubai, United Arab Emirates, 2010, pp. 172-177.

[13] F. Anzum, and M. L. Gavrilova, "Emotion Detection From Micro-Blogs Using Novel Input Representation," in IEEE Access, vol. 11, pp. 19512-19522, 2023.

[14] G. Pierre-Yves, G. Laurent-Walter, Y. Khalafaoui, E. Egyed-Zsigmond, N. Grozavu,“French translation of a dialogue dataset and text-based emotion detection", Data & Knowledge Engineering, vol. 142, 102099, ISSN 0169-023X, 2022.

[15] I. Ameer, N. Bölücü, M. H. F. Siddiqui, B. Can, G. Sidorov, A. Gelbukh, Multi-label emotion classification in texts using transfer learning, Expert Systems with Applications, vol. 213, Part A, 118534, ISSN 0957-4174, 2023.

[16] S. Ghosh, A. Priyankar, A. Ekbal, P. Bhattacharyya, “Multitasking of sentiment detection and emotion recognition in code-mixed Hinglish data," Knowledge-Based Systems, vol. 260, 110182, ISSN 0950-7051, 2023.

[17] D. Silva, F. F. Nadia, E. R. Hruschka, and E. R. Hruschka Jr., "Tweet sentiment analysis with classifier ensembles." Decision support systems, vol 66, pp 170-179, 2014.

[18] R. Erhard and H. H. Do, "Data cleaning: Problems and current approaches." IEEE Data Eng. Bull. vol. 23, no.4, pp 3-13, 2000

[19] B. Haji, C. Wu, and V. Potdar, "Computational approaches for emotion detection in text." 4th IEEE international conference on digital ecosystems and Technologies. IEEE, 2010.

[20] M. Pak, and K. Sanghoon, "A review of deep learning in image recognition." 2017 4th international conference on computer applications and information processing technology (CAIPT). IEEE, 2017.

[21] N. A. Bou, et al. "Speech recognition using deep neural networks: A systematic review." IEEE Access, vol.. 7, pp. 19143-19165, 2019.

[22] K. R. Chowdhary. "Natural language processing." Fundamentals of artificial intelligence, pp. 603-649, 2020.

[23] J. Perols, "Financial statement fraud detection: An analysis of statistical and machine learning algorithms." Auditing: A Journal of Practice & Theory, vol. 30, no.2, pp. 19-50, 2011.

[24]N. Mishra, and S. Silakari, "Predictive analytics: a survey, trends, applications, opportunities & challenges." International Journal of Computer Science and Information Technologies, vol.3, no.3, pp. 4434-4438, 2012.

[25] A. Bajpai, and K. Chadha, "Real-time facial emotion detection using support vector machines." International Journal of Advanced Computer Science and Applications, vol.1, no.2, pp. 37-40, 2010.

[26] A. Emrullah, and A. Müslime, “Classification of the Agricultural Crops Using Landsat-8 NDVI Parameters by Support Vector Machine.” Balkan Journal of Electrical and Computer Engineering. 10.17694/bajece.863147, 2021.

[27]H. Krishnan, M. S. Elayidom, and T. Santhanakrishnan, "Emotion detection of tweets using naïve Bayes classifier." Emotion, vol.4, no.11, pp. 457-62, 2017.

[28] S. Mastura Md, N. Jamil, and R. Hamzah, "Evaluation of support vector machine and decision tree for emotion recognition of Malay folklores." Bulletin of Electrical Engineering and Informatics, vol.7, no.3, pp. 479-486, 2018.

[29] A. M. Zubair, et al., "Performance evaluation of supervised machine learning techniques for efficient detection of emotions from online content." arXiv preprint arXiv:1908.01587, 2019.
[30] A. Graves, J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures." Neural Networks, vol.18, no.(5-6), pp. 602-610, 2005.
[31] M. Schuster, K. K. Paliwal, "Bidirectional recurrent neural networks." IEEE Transactions on Signal Processing, vol. 45, no.11, 2673-2681, 1997.
[32] A. Mohan, and D. Gaitonde, "A Deep Learning Approach to Reduced Order Modeling for Turbulent Flow Control using LSTM Neural Networks." , 2018.
[33] S. Cornegruta, R. Bakewell, S. Withey, and G Montana, "Modeling Radiological Language with Bidirectional Long Short-Term Memory Networks." pp.17-27. 10.18653/v1/W16-6103, 2016.
[34] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724-1734, 2014.
[35] R. Adnan, N. Muhammad, M. Khan, and H. Jamil, "SBAG: A Hybrid Deep Learning Model for Large Scale Traffic Speed Prediction." International Journal of Advanced Computer Science and Applications. Vol. 11, pp. 287-291. 10.14569/IJACSA.2020.0110135, 2020.
[36] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, vol. 86(11), pp. 2278-2324, 1998.
[37] S. Al-Saqqa, H. Abdel-Nabi, and A. Awajan. "A survey of textual emotion detection." 2018 8th International Conference on Computer Science and Information Technology (CSIT). IEEE, 2018.
[38] V. Sundaram et al. "Emotion analysis in text using TF-IDF." 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2021.
[39] Jin, Qin, et al. "Speech emotion recognition with acoustic and lexical features." 2015 IEEE international conference on Acoustics, speech and signal processing (ICASSP). IEEE, 2015.
[40] B. Srinivasu, and M. Suhasini. "Emotion detection on Twitter data using knowledge base approach." International Journal of Computer Applications, vol.162, no.10, pp. 28-33, 2017.
[41] A. Mert, and A. Aydin. "Emotion recognition based on the time-frequency distribution of EEG signals using multivariate synchrosqueezing transforms." Digital Signal Processing, vol. 81, pp. 106-115, 2018.
[42] T. Chen, J. Sihang, R. Fuji, F. Mingyan, and G. Yu. "EEG emotion recognition model based on the LIBSVM classifier." Measurement, vol. 164, 108047, 2020.

[43] Y. Zhong, L. Liu, J. Chen, B. Zhao, and Y. Wang. "Locally robust EEG feature selection for individual-independent emotion recognition." Expert Systems with Applications, vol. 162, 113768, 2020.
[44] S. Saganowski, J. Komoszyńska, M. Behnke, B. Perz, D. Kunc, B. Klich, Łukasz D. Kaczmarek, and P. Kazienko. "Emognition dataset: emotion recognition with self-reports, facial expressions, and physiology using wearables." Scientific data, vol. 9, no. 1, 158, 2022.
[45] K. Muhammad, S. M. Anwar, and M. Alnowami. "Multi-domain feature fusion for emotion classification using DEAP dataset." Ieee Access, vol. 9, pp. 12134-12142, 2021.
[46] U. H. Aka, and S. K. Metin. "Vector-based sentiment and emotion analysis from the text: A survey." Engineering Applications of Artificial Intelligence." Vol. 113, 104922, 2022.
[47] E. Zafer, O. R. Adegboye, and H. Kusetogullari. "A new feature selection scheme for emotion recognition from text." Applied Sciences, vol. 10, no. 15, 5351, 2020.
[48] R. Prabod, S. Abeysinghe, C. Samarajeewa, I. Manchanayake, M. J. Walpola, R. Nawaratne, T. Bandaragoda, and D. Alahakoon. "Gated recurrent neural network approach for multi-label emotion detection in microblogs." arXiv preprint arXiv:1907.07653 (2019).

APPENDIX 1: ETHICAL ASPECT OF EMOTION DETECTION

This project's development and implementation heavily depend on ethical concerns. It is crucial to address potential ethical concerns, assure fairness, and ensure responsible usage of the model because the project's goal is to analyse and understand human emotions based on textual data. Privacy and consent of the people whose text data is utilised to train the emotion detection model are two crucial ethical considerations. It is critical to guarantee compliance with applicable data protection laws and secure adequate consent from data sources. The data can be made anonymous or de-identified to help safeguard personal information.

The possibility of bias in the data and model projections is another thing to take into account. The social, cultural, or gender biases reflected in text data may have an impact on the emotion detection model's fairness and accuracy. By carefully selecting and preparing the training data, as well as by routinely checking in on and assessing the model's performance across various demographic groups, is essential to reduce these biases.

Furthermore, controls against abuse should be included in the appropriate application of the emotion detection model. Ensuring that the model is not applied to manipulate or take advantage of people based on their emotions. Potential harm can be avoided and responsible usage of the model can be encouraged by putting in place the proper governance and rules.

To resolve ethical issues, ongoing oversight, audits, and user input are essential. Any biases or problems that may develop during deployment can be found and fixed through regular evaluation of the model's performance, retraining with updated data.

This project can work towards responsible and ethical use of technology by addressing these ethical issues, ensuring fairness, privacy, openness, and eliminating any biases.

APPENDIX 2: DATA COLLECTION

In order to train an efficient emotion recognition algorithm, data collecting is essential. An overview of the data collection procedure used for this project is provided in this appendix.

1. Data sources: This project's data came from a variety of places, including social media, online platforms, and public text corpora. A variety of websites, including Kaggle, Hugging Face, and social media sites, were scraped to gather text data.
2. Sample Size: 80:20 ratio of data in the datasets were used for training and analysing the emotion recognition model. Each instance includes a text sample and the emotion category that it refers to. The dataset was thoughtfully selected to include a variety of emotions and textual settings.
3. Data Preprocessing: The data underwent a number of preprocessing procedures before being used for training. These included executing text normalisation procedures, handling special characters, and deleting unnecessary metadata, such as usernames or timestamps.
4. Data Labelling: Human annotators were included in the labelling process to support supervised learning. Based on predetermined emotion categories, annotators assigned the relevant emotion labels to the text samples. Techniques for inter-annotator agreement were used to guarantee the labelled dataset's consistency and quality.
5. Privacy of Data and Consent: The protection of privacy was a top priority while collecting the data. There have been measures made to ensure compliance with data protection laws and to erase any personally identifying information. Where necessary, data sources' consent was secured to ensure that data gathering followed ethical guidelines.

Transparency and validity are guaranteed by outlining the data gathering procedure in this appendix. Readers, researchers, and evaluators can use this information as a guide to understand the origin, scope, and preprocessing procedures utilised with the dataset used to train the emotion detection model.

REFLECTION OF THE TEAM MEMBERS ON THE PROJECT

We evolved as a team throughout the duration of the project, both individually and as a whole. We learned significant lessons on effective teamwork, time management, and the importance of clear communication as team members. We learned the value of utilizing each team member's special skills and strengths, which helped us handle difficult problems more successfully. We also understood how important it was to keep an open mind and aggressively seek out different viewpoints in order to encourage creativity and innovation within our team.

We were able to establish a collaborative environment that encouraged open exchange of thoughts and ideas was one of the features of our design approach. We encouraged brainstorming meetings where everyone could freely share their ideas. This gave us the opportunity to come up with a variety of original ideas and explore various design options for our project. In addition, we understood the value of continuous improvement and regularly incorporating input from supervisor and team members, which significantly raised the quality of our final design.

However, a weakness we found in our design process was an impulse to occasionally ignore some practical constraints. While we were motivated about breaking down boundaries and exploring original ideas, we occasionally forgot to take into account possible challenges with execution and feasibility. This sometimes-necessitated rework and changes to our design, which caused brief changes in the project's plan. In order to assure an easier implementation going forward, we realize the necessity for an in-depth review of practical considerations of the design process.

So, our project experience has taught us the importance of strong collaboration, interaction, and accepting different viewpoints. We have experienced how effective a teamwork is in creating innovative solutions. Our design method shown benefits for promoting partnership and idea sharing while also emphasizing the significance of taking practical restrictions into account.