Multi-Label Emotion Classification Using Machine Learning and Deep Learning Methods

by

Drashtikumari Kher

A thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Science (MSc) in Computational Sciences

The Faculty of Graduate Studies

Laurentian University

Sudbury, Ontario, Canada

THESIS DEFENCE COMMITTEE/COMITÉ DE SOUTENANCE DE THÈSE
Laurentian Université/Université Laurentienne
Faculty of Graduate Studies/Faculté des études supérieures

Title of Thesis Titre de la thèse	Multi-Label Emotion Classification Using Machine Learning and Deep Learning Methods	
Name of Candidate Nom du candidat	Kher, Drashitkumari	
Degree Diplôme	Master of Science	
Department/Program Département/Programme	Computational Science	Date of Defence Date de la soutenance February 16, 2021

APPROVED/APPROUVÉ

Thesis Examiners/Examinateurs de thèse:

Dr. Kalpdrum Passi
(Supervisor/Directeur de thèse)

Dr. Ratvinder Grewal
(Committee member/Membre du comité)

Dr. Julia Johnson
(Committee member/Membre du comité)

Dr. Guatam Srivastava
(External Examiner/Examinateur externe)

Approved for the Faculty of Graduate Studies
Approuvé pour la Faculté des études supérieures
Dr. Lace Marie Brogden
Madame Lace Marie Brogden
Acting Dean, Faculty of Graduate Studies
Doyenne intérimaire, Faculté des études supérieures

ACCESSIBILITY CLAUSE AND PERMISSION TO USE

Abstract

Emotion detection in online social networks benefits many applications like personalized advertisement services, suggestion systems, etc. Emotion can be identified from various sources like text, facial expressions, images, speeches, paintings, songs, etc. Emotion detection can be done by various techniques in machine learning. Traditional emotion detection techniques mainly focus on multi-class classification while ignoring the co-existence of multiple emotion labels in one instance. This research work is focussed on classifying multiple emotions from data to handle complex data with the help of different machine learning and deep learning methods. Before modeling, first data analysis is done and then the data is cleaned. Data pre-processing is performed in steps such as stop-words removal, tokenization, stemming and lemmatization, etc., which are performed using a Natural Language Processing toolkit (NLTK). All the input variables are converted into vectors by naive text encoding techniques like word2vec, Bag-of-words, and term frequency-inverse document frequency (TF-IDF). This research is implemented using python programming language. To solve multi-label emotion classification problem, machine learning and deep learning methods were used. The evaluation parameters such as accuracy, precision, recall, and F1-score were used to evaluate the performance of the classifiers Naïve Bayes, support vector machine (SVM), Random Forest, K-nearest neighbour (KNN), GRU (Gated Recurrent Unit) based RNN (Recurrent Neural Network) with Adam optimizer and Rmsprop optimizer. GRU based RNN with Rmsprop optimizer achieves an accuracy of 82.3%, Naïve Bayes achieves highest precision of 0.80, Random Forest achieves highest recall score of 0.823, SVM achieves highest F1 score of 0.798 on the challenging SemEval2018 Task 1: E-c multi-label emotion classification dataset. Also, One-way Analysis of Variance (ANOVA) test was performed on the mean values of performance metrics (accuracy, precision, recall, and F1-score) on all the methods.

Keywords: multi-label emotion classification, Twitter, python, deep learning, machine learning, Naïve Bayes, SVM, Random Forest, KNN, GRU based RNN, ensemble methods, one-way ANOVA

Acknowledgements

I would like to express my gratitude towards my supervisor Dr. Kalpdrum Passi for the continuous support of this research, for his knowledge, innovative ideas, dedication to work has always amused me. It was a great opportunity to work under his guidance and supervision. His guidance helped me in all the time of research, implementation and writing of this thesis. This thesis would not have been possible without his guidance and persistent help. I could not have imagined having a better mentor and advisor for this thesis. I will be always thankful to him.

Also, I would like to thank my parents, partner, and friends whose constant inspiration and support encouraged me throughout my thesis completion. I have no valuable words to give my thanks to all, but my heart is still full of the favours received from each and every person.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

TF-IDF	Term-frequency Inverse-document-frequency
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
SVM	Support vector machine
KNN	K-nearest neighbors
RNN	Recurrent Neural Network
BNet	Binary Neural Network
GRU	Gated Recurrent Unit
CSV	Comma-Separated Values
ReLU	Rectified Linear Unit

Chapter 1

Introduction

1. Introduction

With the increasing popularity of online social media, people like expressing their emotions or sharing meaningful events with other people on the social network platforms such as twitter, Facebook, personal notes, blogs, novels, emails, chat messages, and news headlines [1].

Emotion is a strong feeling that deriving from person's mood or interactions with each other. Emotion mining is very interesting topic in many studies such as cognitive science, neuroscience, and psychology [2]. Whereas emotion mining from text is still in its early stages and still has a long way to proceed, developing systems that can detect emotions from text has many applications. In customer care services, emotion mining can help marketers gain data about how many of their customers are satisfied and what aspects of their service should be improved or revised accordingly to make a strong relationship with their end users [2]. Also, users emotions can be used for sale predictions of a specific product [2].

The intelligent tutoring system can decide on teaching materials, based on users mental state and feelings in E-learning applications. The computer can monitor users emotions to suggest appropriate music or movies in human computer interaction [2]. Moreover, output of an emotion-mining system can serve as input to the other systems. For instance, Rangel and Rosso [2][3] use the emotions identified within the text for author identification, particularlly identifying the writers age and gender. Lastly, however not the least, psychologists can understand patients emotions and predict their state of mind consequently. On a longer period of time, they are able to detect if a patient is facing depression, stress that is extremely helpful

since he/she can be referred to counselling services [2]. With the explosive development of web 2.0 technology, different media are available for people to express their emotions and feelings [2]. This has included another viewpoint to the area. There is analysis on detecting emotions from text, facial expressions, images, speeches, paintings, songs, etc. Among all, voice recorded speeches and facial expressions contain the most dominant clues and have largely been studied [4][5]. Some types of text can convey emotions such as personal notes, emails, blogs, novels, news headlines, and chat messages. Specifically popular social networking websites such as Facebook, Twitter, Myspace are appropriate places to share one's feelings easily and largely.

1.1 Multi-Label classification for emotion classification

Emotion mining is a multi-label classification problem that requires predicting several emotion scores from a given sequence data. Any given sequence data can possess more than one emotion, so the problem can be posed as a multi-label classification problem rather than a multi-class classification problem. Both machine learning and deep learning were used in this research to solve the problem.

1.1.1 Machine Learning based approach:

For the machine learning models, data cleaning, text preprocessing, stemming, and lemmatization on the raw data were performed. The text data was transformed to vectors by using the Term-frequency Inverse-document-frequency (TF-IDF) method, then multiple methods were used-to predict each emotion. SVM, Naive Bayes, Random Forest, and K-nearest neighbors (KNN) classifiers were used extensively to build the machine learning solution. After all the training, various performance metrics measures were plotted for each model concerning every emotion label as a bar plot.

1.1.2 Deep Learning based approach:

For the deep learning, dataset is loaded, then preprocessed, and encoded to perform deep learning techniques on it. From this research shows that Recurrent Neural Networks (RNN) based model performs well on text data, Gated Recurrent Unit (GRU) model was built with an attention mechanism to solve the problem by training for multiple epochs to obtain the best accuracy.

1.2 Motivation

Emotion recognition is the interpretation and classifications of emotions (joy, sad, happy, anger, anticipation, etc.) that use raw sequence data (audio, text, or video) to build systems that can predict emotion scores on future unseen sequence data with extremely high accuracy. Recent advances in Machine Learning and natural language processing (NLP) techniques possess almost human-level performance. These systems are extensively used to solve a wide spectrum of business problems in internet companies. Traditional emotion analysis focuses on multi-class classification, whereas ignoring the co-existence of multiple emotion labels in one instance [1]. There is a possibility of multiple emotions available in one instance/sentence. If the sentence is complex or large, then classification is very complex and time consuming. For each new input a classification label must be determined. Deep learning techniques may give a better performance compared to other machine learning techniques to detect emotions.

1.3 Objectives

Main objective of this work is to detect and identify correlative emotions from the complex and sequential data with a higher accuracy and precision. It will be helpful in many areas, like give better personalized advertisement services, recommending products based on user

emotions, E-Learning, entertainment, medicine, and law etc. More specifically, the goal is to assign automatically an emotion label to each sentence in the given emotion classification dataset, indicating the predominant emotion type expressed in the sentence. The possible labels are joy, sadness, anger, fear, trust, disgust, surprise, and anticipation, optimism, pessimism, and love. In that, eight (joy, sadness, anger, fear, trust, disgust, surprise, and anticipation) basic emotion categories were identified by Plutchik emotion model [6].

1.4 Methodology

In thesis, machine learning and a deep learning-based approaches were used to solve the multi-label emotion recognition problem on twitter data. Both machine learning and deep learning algorithms were applied after applying domain knowledge-based data cleaning, Natural Language Processing (NLP) based data preprocessing, and feature engineering techniques. Different feature engineering and preprocessing techniques were applied for both the solutions, those techniques have been discussed in Chapter 3. Figure 1.1 shows the flow diagram for the classification of emotions in a sequence data.

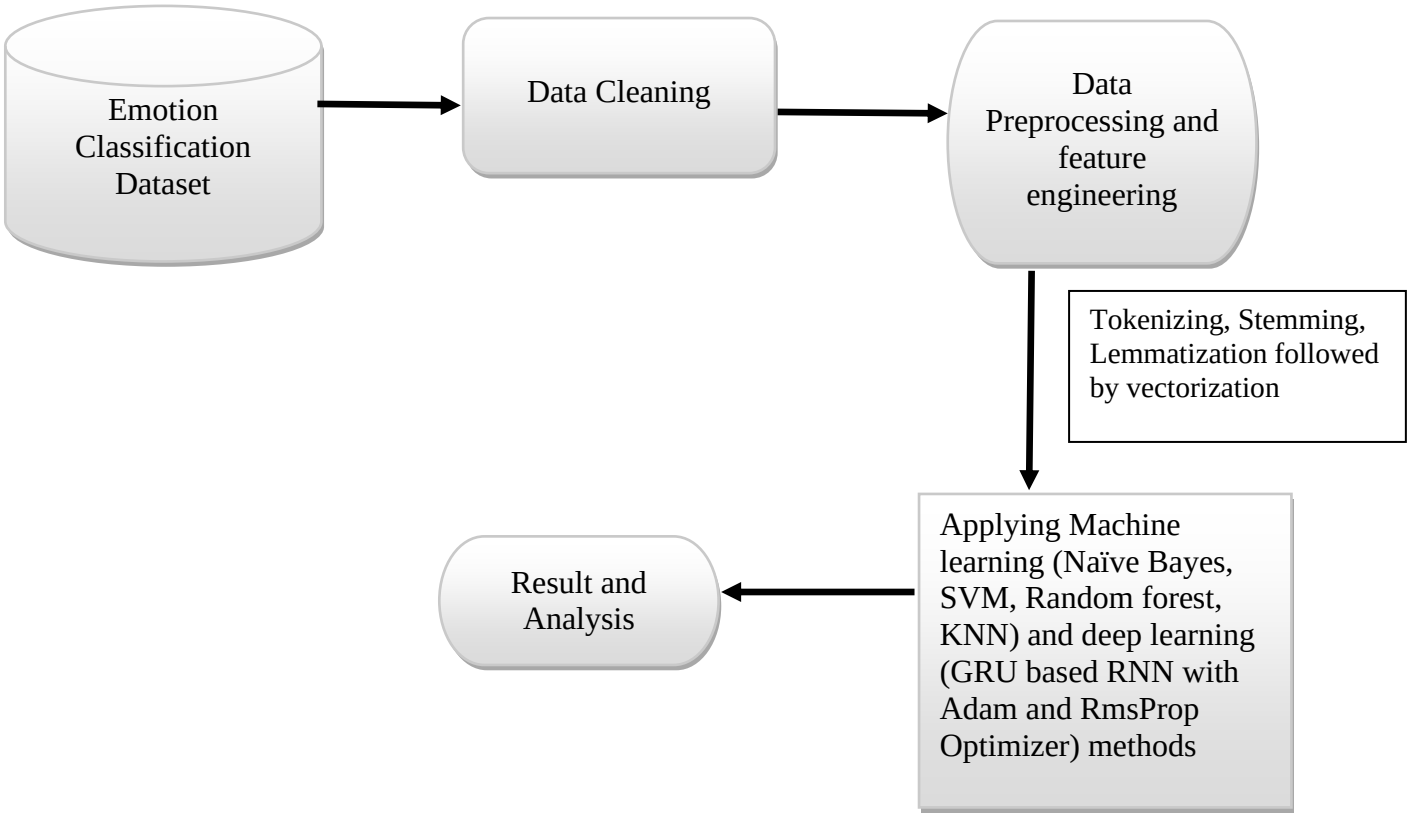

Figure 1.1. Flow Diagram of Multi-Label Emotion Classification

1.5 Outline

In the following, the rest of the thesis is organized as follows:

Chapter 2: Literature Review

The Literature survey has been done during the research work and concepts about emotion detection and emotion classification techniques have been discussed in this Chapter.

Chapter 3: Data and Processing

This Chapter describes the data used for emotion classification and the preprocessing techniques for cleaning and preparation of data for emotion classification by feature extraction techniques.

Chapter 4: Methods

This Chapter presents implementation tools and discusses about the proposed approach of the emotion classification through various phases. The machine learning and deep learning models are discussed in this chapter.

Chapter 5: Results and Discussion

This Chapter provides result of the proposed approach and discussion about all those results. The visualization of the results is shown by Matplot library.

Chapter 6: Conclusion and future work

This Chapter concludes the work and discusses the future scope that can be done as a future extension.

Chapter 2

Literature Review

Many ways are available for detecting emotions from the textual data, for example social media has made our life easier and by pressing just one button everyone can share personal opinion with the whole world. Emotion can be detected from the data with the help of data mining techniques, machine learning techniques and with the help of neural networks [7]. From the examination it was expressed that emotion detection approaches can be classified into three following types: keyword based or lexical based, learning based and hybrid. The most commonly used classifiers, such as support vector machine (SVM), naive bayes and hybrid algorithms [7].

2.1. Related work

Many researches have explored the quality of data in social media and social network, but still a huge gap exists between what was achieved and the expectations. A few researchers have studied social media and social networks mainly using data procured from Twitter. Over the past few years, research in natural language processing has made remarkable progress. Nowadays, most of the natural language processing (NLP) research uses Attention based models extensively.

Lin, Yang and Chen [8] used a procedure to rank readers' emotions in Chinese news articles from Yahoo! Kimo News. They considered eight emotion classes and they used a support vector machine (SVM) for the classification. After performing feature engineering, they got the Chinese Words, news metadata, Chinese character bigram, affix similarity and word

emotion as features, then they trained a SVM classifier on top of these features. The best reported accuracy was 76.88%. On the other hand, reader perspective emotion analysis performed on raw training text data, here one text segment can evoke more than one emotion in readers' mind. News articles evokes some emotional response in readers' minds. So, news data can be used as a potential data source for the computational study of emotions.

Feng et al. [9] used a CNN and LSTM-ATT model for performing emotion classification in their experience. Also, they used student comments records on online learning platform for data preparation, which is more than 2,00000. The author used literature analysis and data pre-analysis to construct a dimensional classification system of academic emotion aspects, as well as to develop an aspect-oriented academic emotion automatic recognition method. An aspect-oriented convolutional neural network (A-CNN) and an academic emotion classification algorithm based on the long short-term memory with attention mechanism (LSTM-ATT) was used to implement their methods. The experiments showed that this model can provide effective and quick identification. The accuracy of the LSTM (Long Short - Term Memory) based network has 71.12% accuracy on the test data.

Yasminaa, Hajarb and Hassana [10] used unsupervised machine learning algorithms to perform emotion classification, they used YouTube comment as their data corpus. They used unsupervised learning algorithm and got accuracy ranging from 67% to 71% for different target emotions. They performed word level classification that computes the relatedness between the word to classify and a particular emotion, then they used the pointwise mutual informative (PMI) parameter for classification. Finally, they got nearly 68.82% average accuracy using Support Vector Machine (SVM) algorithm.

Chew-Yean (2015) [11], from Microsoft research trained a multi-layered neural network with 3 hidden layers (125, 25 and 5 neurons respectively), the model learns to tackle the task to identify emotions from text using a bi-gram as the text features representation. ISEAR dataset is used, which consists of 2500 sentences with 5 categories (angry, sad, fear, happy and excited) of emotions to train the deep learning model; after training the model she got 60.60% accuracy on the test data.

Vishwakarma and Bhattacharya [12] used 2809 annotated tweets. They showed a way which uses self-attention and bidirectional long short-term memory (LSTM) network for emotion detection. They also trained SVM and Naive Bayes algorithms in multilabel settings, in the best case they got an accuracy of 80.35% using support vector machine (SVM).

Kozareva et al. [13] used 686 headlines dataset. They pursued an emotion classification approach that is based on frequency and co-occurrence of a bag of word counts collected from the World Wide Web and its pairs with different emotions like joy, disgust, fear, anger, sadness, surprise. The hypothesis is that words which tend to co-occur over many documents with a given emotion are exceedingly probable to express this emotion. Their system has a better detection of the negative emotions such as anger, disgust, fear and sadness in comparison to the positive emotions such as joy and surprise. They got better accuracy 86.70% but low precision score which is only 18%.

Vaughan, Mulvenna and Bond [14] used a talklife dataset which was generated by recording all talklife messages over a 12-hour period. To detect emotions, they first categorised the emotions in a model so that it can be analysed against the text. Their solution for emotion classification had 14% classification accuracy.

Saputri, Mahendra and Adriani [15] used five emotion classes (anger, joy, sadness, fear and love) and 4403 Indonesian tweet data to build an emotion classification system which got nearly 69.73% F1 score which outperformed their baseline model which has 26.64% F1 score. They used lexicon, parts of speech tag, word embeddings and orthographic features to build the model, which outperformed their baseline model. Before training they performed extensive data cleaning and preprocessing which helped to get a decent prediction out of the model.

Lee, DongIl Shin, and DongKyoo Shin [16] performed emotion classification by using machine learning techniques. They used support vector machine (SVM) and K-means clustering to extract emotions and perform classification on top of them. They collected brain-wave signals of the user and then preprocessed the data using FIR (finite impulse response) filters. They used Independent Component Analysis (ICA) to detect eye blink. Then they transformed the time domain signal to frequency domain using Fourier transform. Finally, they got 71.85% accuracy using SVM and 70% accuracy using K-means algorithm.

Wang et al. [17] applied two different machine learning algorithms to classify emotions and after preparing data they got highest accuracy of 65.57% on the training data containing about 2 million tweets. They used user generated content on twitter as their training dataset. The models that they used were Liblinear and multinomial naive bayes, Logistic regression with default parameters.

Hussien et al. [18], showed an automated approach which can annotate the training data of twitter dataset using certain emojis, these annotations can generate new emotion results or outcomes. Classifiers tend to give more accurate results when trained on the data which were

annotated by that automated system, than the training data which were manually annotated. Their dataset has emotional Arabic tweets (22,752) which were collected from Arabic tweets dataset (134,194). Emotion categories include four parts: anger, disgust, joy and sadness. They used Support Vector Machines (SVM) and Multinomial Naive Bayes (MNB) to perform classification. The approach showed that the automated annotation data labelling can bring us more accurate models and got highest precision of 75.7% using MNB.

Danishman and Alpocak [19] used ISEAR dataset and classify emotions into 5 classes (anger, fear, disgust, joy and sadness). They used wordnet and Wisconsin Perceptual Attribute rating database (WPARD) and they performed TF-IDF encoding technique to get the feature vector. They trained a classifier on top of TF-IDF based features and got 70.2%, 60.8% and 34.8% accuracy respectively using support vector machine (SVM), Naïve Bayes and vector space model.

Emotion detection conducted by El Gohary [20] et al., where they implemented a model that can classify emotions on Arabic children stories, they used a lexicon-based approach. In that, 100 documents (2514 sentence) were used as a dataset. Also, they applied the model on other texts and determined their emotion scores. Their training dataset has 6 emotions (joy, fear, anger, sadness, disgust, surprise) in 65 documents. The model achieved 65% accuracy for emotion detection in Arabic text.

Burget and Karasek [21] used 1000 Czech Newspaper Headlines dataset. The system depends on the pre-processing of the Czech Newspaper Headlines dataset and labeling using a classifier. The pre-processing was done at the word and sentence levels by applying removing stop words, POS tagging, stemming and lemmatization. Term Frequency-Inverse Document Frequency (TF-IDF) was used to calculate the relevance between each term and six emotion classes (joy,

fear, anger, surprise, sadness, disgust). They achieved average accuracy of 80% using support vector machine (SVM linear kernel) with 10-fold cross validation.

Trohidis et al. [22] performed multilabel emotion classification on a music dataset which has 593 songs categorized with one or more out of 6 classes of emotions. It is based on the Tellegen-Watson-Clark model of affect. They used 72 features to perform emotion classification on music data. They trained multiple models to detect emotions on time series dataset. They got an accuracy of 81.51% using random k-label sets (RAkEL) method.

In this thesis, multi-label emotion classification problem was used instead of single-label or multi-class classification problem on twitter data. Moreover, deep learning and machine learning methods were used to solve this problem. For emotion detection emotion classification dataset [23][24][25] was used from twitter data. Many researchers used multi-label classification [1][6] problem but they achieved very less accuracy scores of 40-60%. In this research an accuracy of 82.3% was achieved using Gated Recurrent Unit (GRU) based recurrent neural network (RNN) with RmsProp optimizer.

Chapter 3

Data and Preprocessing

3.1 Introduction of Twitter

Twitter is a social networking site where we can share our ideas, opinions with a wide variety of people. Twitter was founded in 2006 [26] and the founder wanted to make a communication platform where users can upload their status, send messages to the people they are connected with. The idea was proposed by Jack Dorsey to his colleagues and they started working on the idea, later it became successful. Twitter allows its users to promote interesting ideas, blogs and research to a large number of people through tweets. The types of tweets depend upon the fields of interest and the organizations people belong. As of now, Twitter is one of the most engaging social media platforms in the world. Table 3.1 shows some of the statistics on the number of users of Twitter.

Table 3.1. Basic statistics of the twitter platform

Total created accounts	1.3 billion
Monthly active users	335 million
Daily active users	145 million
Users who sent tweets after creating an account	550 million

3.2 Characteristic of Twitter data

The main objective of building such a platform was to represent one's idea concisely. Most of the tweets are crisp and on point. People can also share photos, videos via tweets, there is also a way to attach certain hashtags to tweets, attaching hashtags can enrich the spread of the tweet on the platform. There is also a way to promote tweets, it helps to get more followers and tweets. It has been observed that twitter users share a lot of blog content, and research work. Sometimes, users accumulate and organize interesting events to enrich their knowledge. One user can post a tweet with a maximum of 140 characters long in size [26]. Tweets can be effective for emotion classification as most of them are crisp and on point.

Username or handle represents an identity of a user on the platform, after signing up onto the platform, a username is created for further procedure. For instance, if someone's name is David James, he might want to choose a username such as @david_james. "@" taken after by a word represents a username of a particular user, it is worth specifying that usernames are unique to every user and that we can find any particular user if we all know his username. "#" taken after by some words presents a hashtag, it is used to attach some particular posts to some particular group of individuals, hashtags help the user to find particular posts which he/she could be interested in. Hashtags include context and that they provide conversation longevity.

3.3 Dataset

A dataset is a collection of related sets of documents or information of separate members that can be accessed, stored and manipulated by a computer. A dataset has values for each member which can depict interesting features that are certain members or classes. In an ideal case, the dataset should be balanced, that is, the dataset should not be biased towards any specific class and the dataset should contain a sufficient amount of observations. For text data, the data is expected in comma-separated values (CSV) format as these files can easily be loaded into

memory by using pandas, which is a data analysis library for Python programming language. In this research, 10,983 English tweets were used for multi-label emotion classification from SemEval-2018 [23]. Here, sample dataset is shown in Table 3.2. The dataset of emotions classification includes the eight basic emotions (joy, sadness, anger, fear, trust, disgust, surprise, and anticipation) as per Plutchik (1980) [6], as well as a few other emotions that are common in tweets which are love, optimism, and pessimism.

Table 3.2. A sample of the emotion classification dataset

0, *“worry is a down payment on a problem you may never have '. joyce meyer. <hashtag> motivation </hashtag> <hashtag> leadership </hashtag> <hashtag> worry </hashtag>*,"['0', '1', '0', '0', '0', '0', '1', '0', '0', '0', '1']"
1, *whatever you decide to do make sure it makes you <hashtag> happy </hashtag>* .,"['0', '0', '0', '0', '1', '1', '1', '0', '0', '0', '0']"
2, *"<user> it also helps that the majority of <allcaps> nfl </allcaps> coaching is inept. some of bill o ' brien ' s play calling was wow,! <hashtag> gopats </hashtag>*","['1', '0', '1', '0', '1', '0', '1', '0', '0', '0', '0']"
3, *accept the challenges so that you can literally even feel the exhilaration of victory. ' - - george s. patton* 😉,"['0', '0', '0', '0', '1', '0', '1', '0', '0', '0', '0']"
4, *my roommate: it ' s okay that we can not spell because we have autocorrect .<hashtag> terrible </hashtag> <hashtag> first world probs </hashtag>*,"['1', '0', '1','0', '0', '0', '0', '0', '0', '0', '0']"
5, *no but that ' s so cute. atsu was probably shy about photos before but cherry helped her out uwu*,"['0', '0', '0', '0', '1', '0', '0', '0', '0', '0', '0']"
6, *do you think humans have the sense for recognizing impending doom? “['0'*, '1', '0', '0', '0', '0', '0', '1', '0', '0', '0']"

The above dataset can be used to pose both multi-label and multiclass classification problems. The input variables of the dataset are the features that are chosen to predict from and are independent variables. For emotion classification, input variables are sequence data such as text, speech, etc. In this research, the input variables are raw tweets. Output variables are the dependent variables that depend upon the input variables. For emotion classification, each input variable may have one or more output variables. If each input variable has more than one output variable corresponding to it then the problem can be posed as a multilabel classification, else the problem is posed as a multiclass classification. In this thesis, dataset has more than one emotion score, therefore the problem is posed as a multilabel emotion classification problem.

3.4 Data Preprocessing

Data preprocessing is the most crucial data mining technique that transforms the raw data into a useful and efficient format. Real-world information is frequently inconsistent, incomplete, or missing in specific behaviours and is likely to contain lots of errors. It is a demonstrated technique of resolving such issues. It prepares raw data for further processing. Different tools are available for data preprocessing e.g., data Preprocessing in R, Weka, Rapid Miner, python and nltk NLP tool. Data preprocessing is divided into a few stages which include Data cleaning, data transformation, dimensionality reduction etc. Data preprocessing techniques can differ in Machine Learning and Deep Learning. Data Preprocessing structure is show in Figure 3.1.

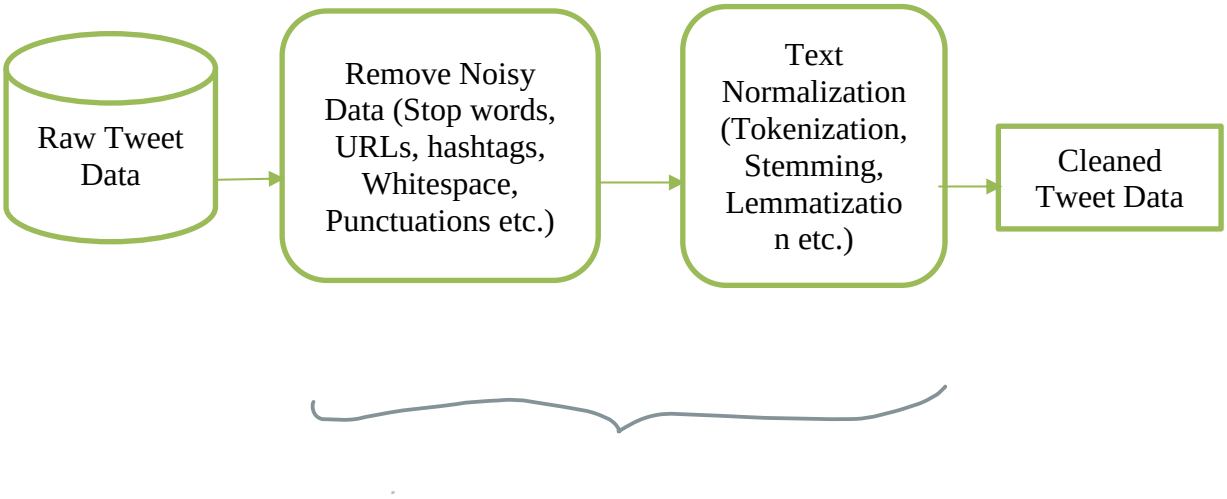

Figure 3.1. Structure of Data Preprocessing

3.4.1 Data Preprocessing in Machine Learning:

The data preprocessing steps that are performed before starting machine learning algorithms are as follows.

1. Data Cleaning

For data cleaning, sometimes tweets possess certain usernames, URLs, hashtags, whitespace, Punctuations, etc., which is not helpful in machine learning algorithms to get better accuracy. Then, remove all noisy data from every tweet. All special characters are replaced with spaces. This step is performed as special characters do not help much in machine learning modeling. Every tweet is transformed into lower case. Also, duplicate tweets are identified and removed. Table 3.3 shows an example of cleaning the tweets.

Table 3.3. Data cleaning in Tweets

Original Tweets	1. Worry is a down payment on a problem you may never have**'. Â** Joyce Meyer**. #motivation #leadership #worry** 2. Whatever you decide to do make sure it makes you #happy**.**
Tweets after Cleaning	1. worry is a down payment on a problem you may never have joyce meyer motivation leadership worry 2. whatever you decide to do make sure it makes you happy

2. Remove Stop words:

Stop words are words that are finalized in the Natural Language Processing (NLP) step. “Stop words” or “Stop word lists” consists of those words which are very commonly used in a Language, not just English. Stop word removal is important because it helps the machine learning models to focus on more important words which result in more accurate prediction. Stop word removal also helps to avoid problems like the curse of dimensionality as it reduces the dimensionality of the data. It is important to note that there is a total of 179 stop words available in the English language using NLTK library [27].

Removing stop words using python libraries is pretty easy and it can be performed in many ways. For example, “NLTK” module is used to remove stop words from each tweet. More specifically, the “stop words” module is used in the “corpus” package of NLTK. Example of English stop words are: {a, about, above, after, again, against, all, am, an, and, any, are, aren't, as, at, be, because, been, being, below, between, before, both, but, by, can't, cannot, could, couldn't, did, didn't, do, does, doesn't, doing, don't, down, during, each, few, for, from, further, had, hadn't, has, hasn't, have, haven't, having, he, he'd, he'll, he's, her, here, here's, hers, herself, him, himself, his, how, how's, I, I’d, i'll, I’m, I’ve, if, into, in, is, isn't, it, it's, its, itself, let's, me, more, most, mustn't, my, myself, no, nor, not, of, off, on, once, only, or, other, ought, our,

ours, etc}. Table 3.4 shows an example of stop word removal.

Table 3.4. Remove stop words in Tweets

Original Tweets	1. worry **is a down** payment **on a** problem **you** may never have joyce meyer motivation leadership worry 2. whatever **you** decide **to do** make sure **it** makes **you** happy.
Tweets after removing stop words	1. worry payment problem may never joyce meyer motivation leadership worry 2. whatever decide make sure makes happy

3. Tokenization

In simple terms, tokenization is a process of turning sequence data into tokens. Tokenization is the most important natural language processing pipeline. Tokenization turns a meaningful piece of text into a string character named tokens. An example of tokenization is shown in Table 3.5.

Table 3.5. Tokenization

Original Tweets	1. worry payment problem may never joyce meyer motivation leadership worry 2. whatever decide make sure makes happy
After Tokenization	1. [‘worry’, ‘payment’, ‘problem’, ‘may’, ‘never’, ‘joyce’, ‘me yer’, ’motivation’, ’leadership’, ’worry’] 2. [‘whatever’, ‘‘decide’, ‘sure’, ‘makes’, ‘happy’]

There are many tokenization algorithms available, some of the most used tokenization algorithms are:

1. Penn-tree tokenizer

2. Word Piece tokenizer

3. Byte pair encoding tokenizer

4. Sentence piece tokenizer

5. Sub word tokenizer

Before proceeding into advanced preprocessing techniques, it is really important to perform tokenization. Tokenization can be performed by using the NLTK's tokenizer tool.

4. Stemming

Stemming is a process of turning inflected words into their stemmed form. Stemming also helps to produce morphological variants of a base word. **Stemming is the part of the word which adds inflected word with suffixes or prefixes such as (-ed, -ize, -s, -de, mis).** So, stemming results in words that are not actual words. Stemming is created by removing the suffixes or prefixes used with a word. Stemming algorithms are extensively used before Machine learning modeling. The Natural Language Processing Toolkit (NLTK) in python posses inbuilt stemming algorithms. The two most commonly used stemming algorithms are:

1. Snowball Stemmer

2. Porter stemmer

NLTK module implements both of them. It is important to note that the input of the stemmers is tokenized words. Table 3.6 shows an example of stemming on tweets.

Table 3.6. Stemming on Tweets

Original Tweets	1. worry payment problem may never joyce meyer motivation leadership worry 2. whatever decide sure makes happy
Stemming on tweets	1. **worri** payment problem may never **joyc** meyer **motiv** leadership worri 2. **whatev decid** sure make happy

5. Lemmatization

The key to this process is linguistics and it depends on the morphological analysis of each word. Lemmatization removes the inflectional endings of words and returns the dictionary form of the word, which is also known as “Lemma”. Lemmatization also uses wordnet, which is a lexical knowledge base. Lemmatization is performed after stemming, and it is performed on the tokenized words. After stemming, lemmatization is the most important pipeline of Natural Language Processing. In this research, stemming and lemmatization are used extensively.

Some of the examples of lemmatization are given below. Table 3.7 shows an example of lemmatization.

“Rocks” becomes “rock”

“Corpora” becomes “corpus”

“Better” becomes “Good”

Table 3.7. Lemmatization on Tweets

Original Tweets	1. worry payment problem may never joyce meyer motivation leadership worry 2. whatever decide sure makes happy
Lemmatizing on tweets	1. worry payment problem may never joyce meyer motivation leadership worry 2. whatever decide sure make happy

3.4.2 Data Preprocessing in Deep Learning:

The data preprocessing steps that has been performed before deep learning modeling is quite different from the preprocessing steps for Machine Learning modeling. The preprocessing for deep learning steps is demonstrated as follows.

Step 1: Tokenizer tokenizes each tweet using Keras tokenizer and encodes texts using a different approach.

Step 2: The Keras tokenizer builds a vocabulary dictionary which has words as keys and count of the word in the whole corpus as a value.

Step 3: Then the dictionary is sorted in descending order, which means the words which occur most in our corpus have a very high-count value.

Step 4: Then index of the dictionary (such that the most occurred word in the vocabulary dictionary has index value 1, the second-most occurred word should have an index 2, and so on) have been used to represent words.

Step 5: After that each token of each tweet/input variable gets replaced by the index of the corresponding token or key in the dictionary.

Step 6: After performing step 5 each tweet gets a vector representation. But it's also important to pad each tweet using a padding technique such as Zero-padding. Each text sequence is padded because sequence models expect sequences of the same length.

For deep learning models, stemming and lemmatization have not been performed as deep learning models can automatically learn to embed word representation in numbers. Except that, word2vec models used to get word embeddings for tokens. For 310-dimensional embedding, word2vec models return 300-dimensional vectors and 10 affective dimensional vectors as an output of each word, these models also take care of semantic similarity between words.

Chapter 4

Machine Learning and Deep Learning Methods

In this chapter, the tools are discussed that have been used as a part of experimental analysis for emotion recognition from text data. Also, different methods are discussed for emotion recognition, starting from traditional Machine Learning methods to state-of-the-art Deep Learning methods, that are formulated as multi-label classification problem.

Section 4.1 discusses basic introduction to Python programming language and various packages and libraries that have been used within Python programming environment. Section 4.2 discusses traditional Machine Learning based approaches to solve the problem of emotion classification. Finally, Section 4.3 briefs about Deep Learning methods, relevant data processing techniques, and state-of-the-art architectures of the area.

4.1 Multi-label Emotion Classification using Python

Python is a dynamic, interpreted language. It was created by Van Rossum [28]. It has no type declarations of variables, functions, parameters, or methods in source code. This makes the coding short, saves time, and is flexible. Also, it avoids the compile-time type checking of the source code. It tracks the types of all values at runtime and flags code that does not make sense as it executes. A key feature of python are easy syntax and reliability, high-level language, Object-oriented programming, and cross platform. In this research, python 3.7.4 version was used for data preprocessing, multi-label emotion classification, and data visualization. Some of the most common libraries were used extensively in Natural Language Processing (NLP) and Deep Learning. Also, python libraries are shown in Figure 4.1.

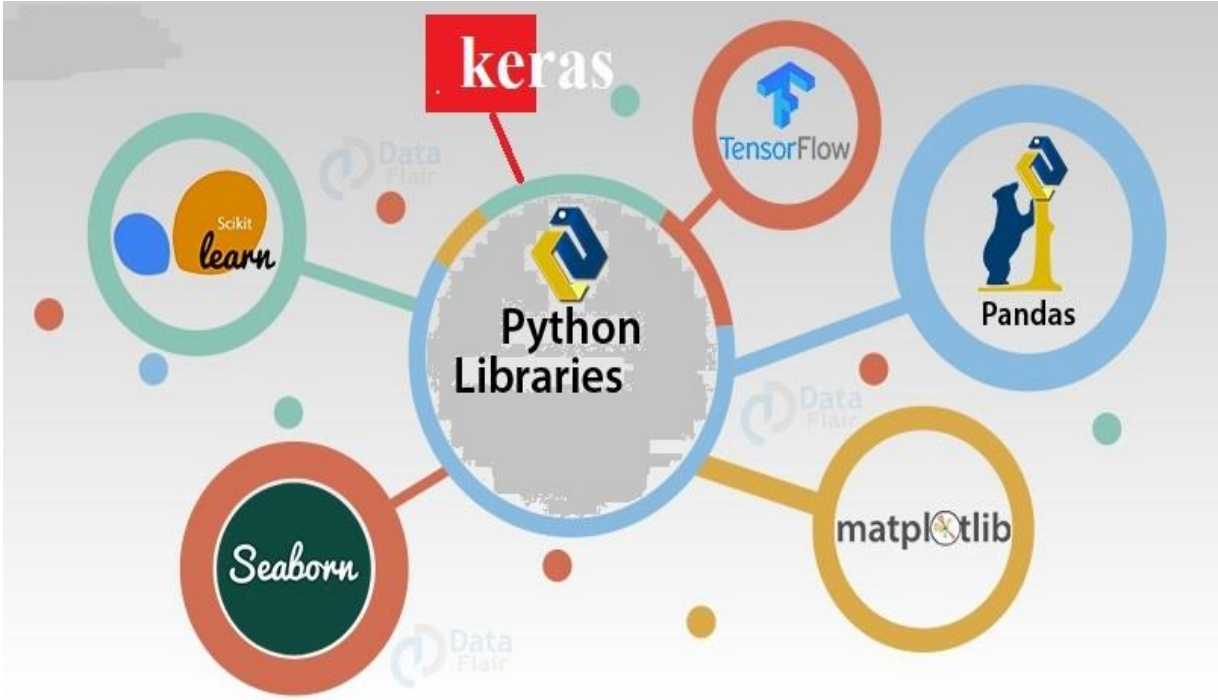

Figure 4.1. Python libraries [29]

The following libraries have been used in this research:

1. **Scikit-Learn**: A library for machine learning modeling which includes various regression, classification and clustering algorithms and measures performance metrics depending on the type of the problem.

2. **Pandas, Seaborn, NumPy,** and **Matplotlib**: These libraries have been used for data analysis, data cleaning, plotting, and statistical analysis.

3. **Natural Language Processing Toolkit (NLTK):** This is used for processing and cleaning Natural Language from raw data. The process of changing data to a form that a computer can read and process is referred to as pre-processing. One of the major forms of pre-processing is to filter out unnecessary data.

4. **TensorFlow:** It is an open source library which is used for deep learning and traditional machine learning applications. It can run on single CPU and GPU system as well as mobile devices and expansive scale distributed systems of hundred of machines [30]. Also, It allows to create a large scale neural networks with many layers. It is used for classification, perception, prediction, etc.

5. **Keras:** It was developed with a focus on enabling quick experimentation. It is more flexible, constantly being updated and being further integrated with TensorFlow. It allows for simple and fast prototyping through user friendliness, scalability, and extensibility. It supports both convolutional neural networks (CNN) and recurrent neural networks (RNN), and also supports combinations of both RNN and CNN [31]. Keras programming runs orderly on CPU and GPU and it is compatible with Python version 2.7-3.9.

6. **Re:** Module used for pattern matching and data cleaning.

4.2 Machine Learning methods for Emotion Classification

The most popular machine learning methods such as Naïve bayes, Support Vector Machines (SVM), Random Forest, and K-Nearest Neighbor (KNN) have been discussed in this section. For the Machine learning models, data cleaning, text preprocessing, stemming, and lemmatization on the raw data were performed. Feature engineering converts the text/string data to a format that machine learning algorithms would interpret. It is an important step before applying any of the previously mentioned machine learning algorithms. The overview of applying machine learning techniques to the emotion classification labeled data and analysis is shown in Figure 4.2.

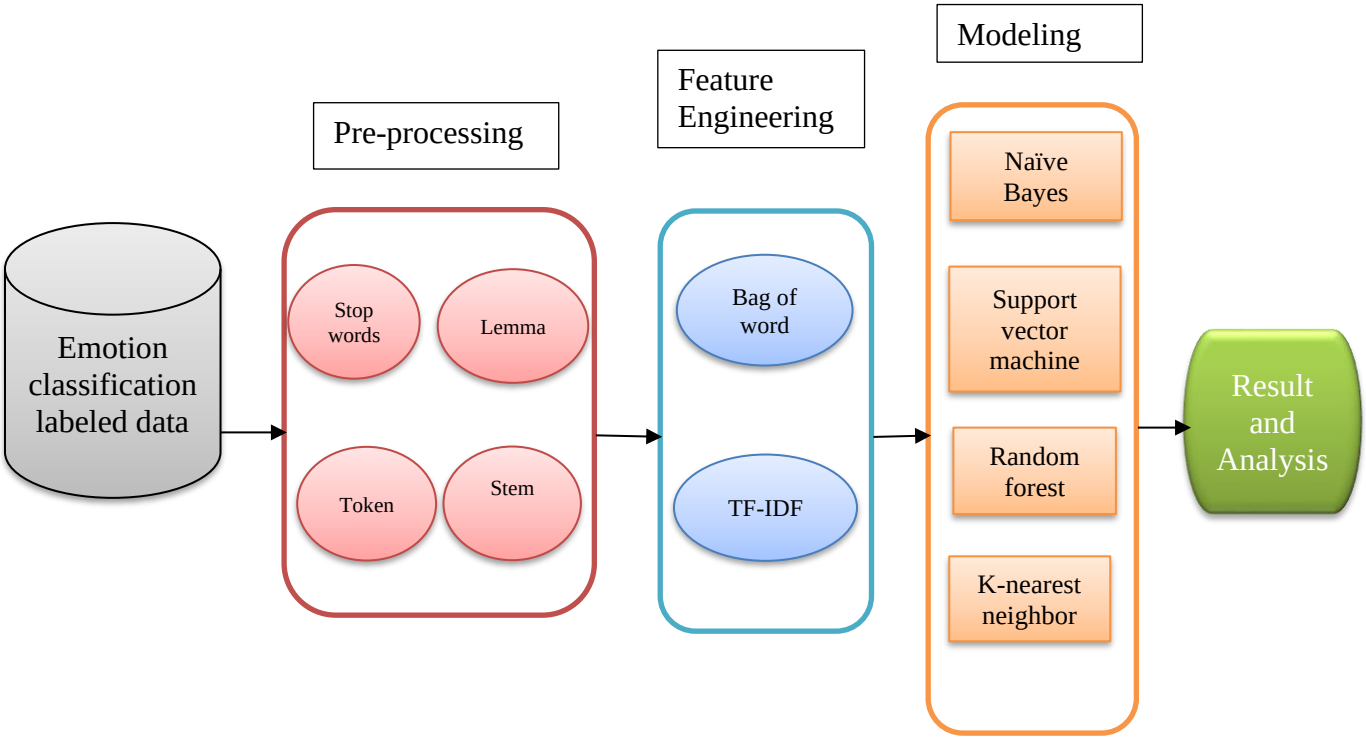

Figure 4.2. Overview of applying machine learning techniques

Feature Engineering

The cleaned and preprocessed tokens of tweets are obtained after all the preprocessing where each token is a "string". Machine learning models cannot work with strings, they only work with numbers. The tokens are transformed into numbers by using the methods given below.

1. Bag of Words (BOW)
2. Term frequency and Inverse document frequency (TF-IDF)

It is always a better idea to use TF-IDF rather than BOW as the TF-IDF feature engineering technique also preserves some semantic nature of the sequence. For this research, the TF-IDF feature engineering technique was used to encode tokens as numbers. The way Term Frequency and Inverse Document Frequency work is explained below.

Term Frequency- Inverse Document Frequency (TF-IDF):

The term "term frequency" means how often a word occurs in a text sequence, it can be thought of as the probability of a word in a text document. The equation of TF-IDF consists of two parts, the first part computes the Term Frequency and the second one computes Inverse Document Frequency.

$$Term\ Frequency = \frac{Number\ of\ times\ w\ occured\ in\ r}{total\ number\ of\ words\ in\ r}$$

Where, *w* is word and r is a document

The term frequency of any word in a document lies between 0 and 1. On the other hand, inverse document frequency gives importance to rare words, and it is defined as follows.

$$Inverse\ Document\ Frequency\ (w, document) =$$

$$\log\left(\frac{Number\ of\ document}{total\ number\ of\ documents\ which\ consists\ w}\right)$$

If a word has more frequency in the text corpus then the inverse document frequency will be low, else if the word is rare then the inverse document frequency will be higher. The way the TF-IDF value is computed is as follows.

$$TF - IDF\ value =$$

$$Term\ frequency\ (w, r) * Inverse\ document\ frequency\ (w, document)$$

The term frequency value will be higher if the word is frequent and the inverse document frequency value will be higher if the word is rare in the document corpus. This operation was performed for each word in a tweet to get the number representation of the tweet or the input variable. TF-IDF also preserves some semantic value regarding the sequence. Before machine learning modeling, TF-IDF was used to encode tokens into vectors. It is important to note that each of the vectors will be of the same dimension, which makes our modeling easier! It is important to note that only unigram-based TF-IDF feature engineering technique was used as the n-gram based TF-IDF technique can lead to problems such as the curse of dimensionality.

4.2.1 Naive Bayes

First, very simple probabilistic model is used whose model name is Naive Bayes. This model uses Bayes theorem extensively for training. In this multilabel classification s, single Naive Bayes model is trained for predicting each output variable. Moreover, 11 Naive Bayes models are trained for predicting 11 emotion scores. The way of Naive Bayes work is as follows. Suppose data point x with n features and there are a total of K possible classes. Then,

$x = (x_1 , x_2 , \dots , x_n)$

$classes = (c_1 , c_2 , \dots , c_k)$

Given a text sequence "*x*", we want to find the conditional probability for every class and predict the class which possesses the highest conditional probability.

Probability of a class label $\boldsymbol{C_i}$ *given a point x =* $\boldsymbol{p(C_i \mid x)}$

Subsequently, Bayesian probability of the given class $\boldsymbol{C_i}$ with an instance $\boldsymbol{x}$ can be computed $\boldsymbol{p(C_i \mid x)}$ using following Bayesian theorem.

$$p(C_i \mid x) = \frac{p(C_i)\, p(x \mid C_i)}{p(x)}$$

For text input, the model is supposed to predict the emotion scores in multilabel classification, where each model is trained for predicting each of the 11 emotions.

From given a query text, 11 different emotion scores are to be obtained.

$p(emotion = sad \mid text) = ?$

$p(emotion = anger \mid text) = ?$

$p(emotion = joy \mid text) = ?$

……

$p(emotion = love \mid text) = ?$

4.2.2 Support Vector Machine (SVM)

The support vector machine is a supervised learning distance-based model. Support vector machine (SVM) is extensively used for classification and regression. In this research, support vector machine (SVM) is used as a classification problem. There are variations of support vector machines algorithm with different kernel functions.

1. Linear Support vector machine
2. Gaussian Support vector machine

In this research, the linear kernel of the support vector machine was extensively used where several linear support vector machine models were trained for predicting several emotion scores. It was assumed that the data points are linearly separable in the high dimensional space, so Linear SVM was used instead of Gaussian SVM. The way linear SVM works is as follows, Linear SVM tries to find a hyperplane that best separates all input variables. It is important to realize that the plane which best separates the data points is the same place which maximizes the margin. So, the decision boundary for Linear SVM is a hyperplane. Figure 4.3 shows the separating hyperplane with margins.

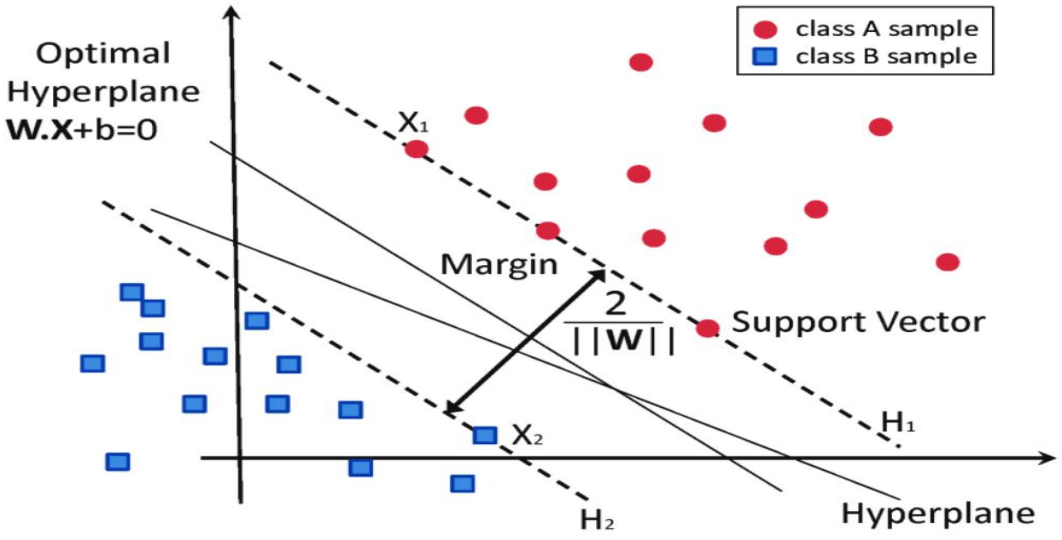

Figure 4.3. Optimal hyperplane for Support vector machine (SVM) [32]

Suppose w is a vector that is perpendicular to the optimal hyperplane, if so then w should also be perpendicular to the other two hyperplanes as they are parallel to the optimal hyperplane. Suppose H_0 is the optimal hyperplane. H_1 and H_2 are hyperplanes that are perpendicular to H_0. These are related through the following equations.

H_0: $w^{\mathrm{T}} x + b = 0$

H_1: $w^{\mathrm{T}} x + b = 1$

H_2: $w^{\mathrm{T}} x + b = -1$

Where, *w* is weight vector

x is input vector

b is bias

But w is weight vector not a unit vector. So,

$w^{\mathrm{T}}.w \neq 1$

Suppose the margin is "d". Now, by using linear algebra,

$$d = \frac{2}{||W||}$$

This is some basic information about optimal hyperplane for support vector machine (SVM) algorithm.

In this thesis, each support vector machine classifier was trained for predicting each emotion and the accuracy of each emotion was obtained, which is discuss in Chapter 5.

4.2.3 Random Forest

Random forest classifier is used for solving the multilabel emotion recognition problem, where random forest classifier is trained for predicting each emotion score (joy, sad, angry, happy etc.). Random forest is a tree-based classifier that uses bagging and aggregation for training and predicting. Random forest uses shallow decision trees, and they train those decision trees on randomly sampled columns and rows from the training data. It is important to note that those

decision trees are not deep, they are shallow. Multiple base models were trained, and their output was aggregated for predicting the output variable. The number of decision trees that train for the random forest algorithm is a hyperparameter, with 100 decision trees trained for random forest model. The way random forest works is as follows.

First, the random forest algorithm performs bagging on all the input variables. Bagging is also known as bootstrapped aggregation which reduces the variance of the base models. On the training data, first take a random sample of a certain number of rows with replacements, then repeat the process a number of times in order to train base models by using the bagged data and use out of bag points for testing the performance of these models. Visualization can be helpful to understand bagging more clearly which is shown in Figure 4.4.

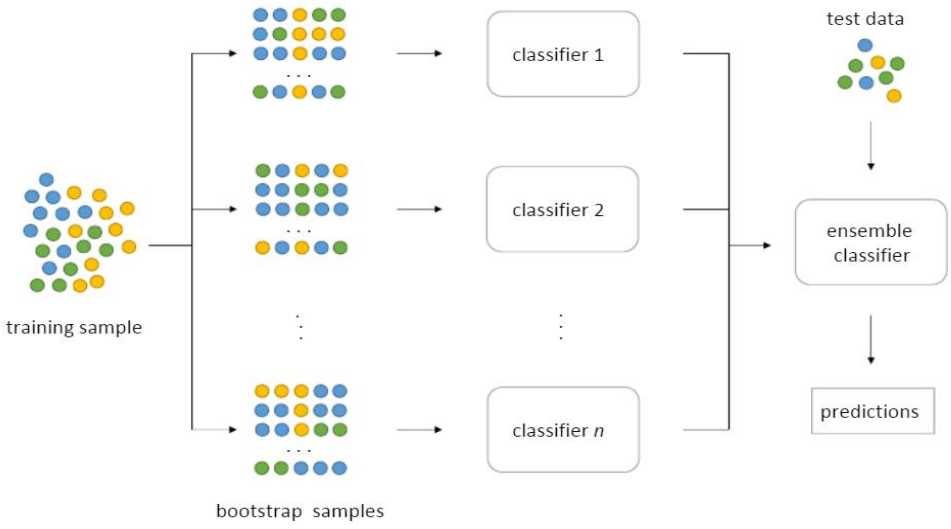

Figure 4.4. The Bagging approach [33]

After performing bagging, n decision tree models are trained on the n sampled data points. For prediction, a majority vote of all the n decision tree models is taken and the class label with maximum votes is selected.

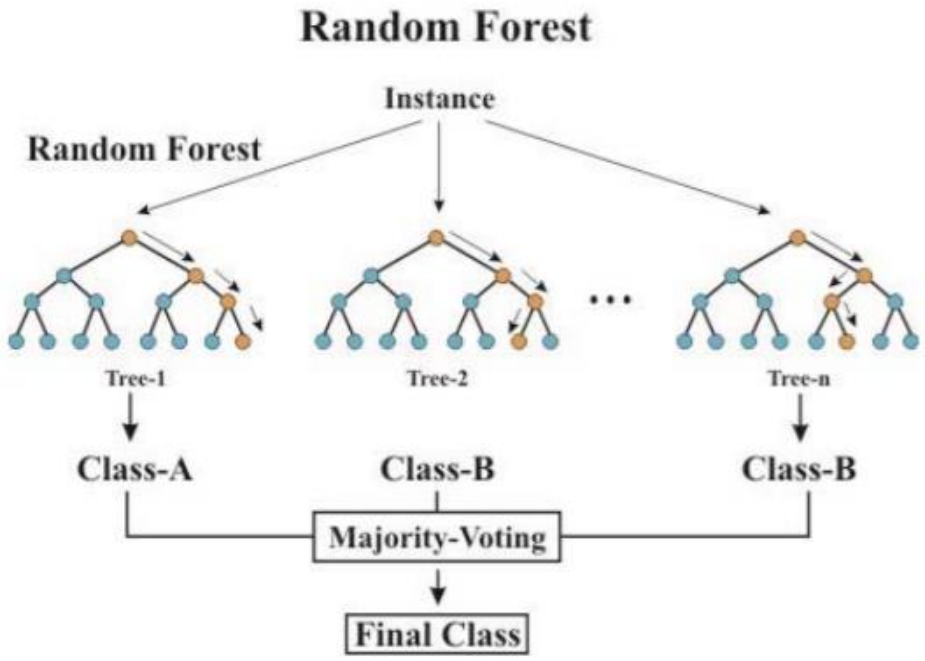

Figure 4.5. Random forest classifier [34]

In the case of regression, first take the mean or median of all the predictions instead of majority voting. But since it is a classification problem, so majority voting is applied. In this research, random forest classifiers are trained for predicting 11 emotion scores. The accuracy scores with respect to emotions are discussed in Chapter 5.

4.2.4 K-Nearest Neighbor (KNN)

K-Nearest neighbor (KNN) is a supervised learning algorithm which is used for classification problem. K-nearest neighbors is also used for the same purpose where KNN classifier is trained for predicting each emotion. K-nearest neighbor is a distance-based learning algorithm. As the data is high dimensional there is a possibility to get a problem called the **curse of dimensionality**. For this multilabel emotion recognition task, KNN is applied on the feature engineered input variables. The way K nearest neighbor works is as follows.

Step 1: First initialize K to chosen number of neighbors.

Step 2: Then calculate the Euclidean distance between the query point and the current example from our data.

Step 3: After that repeat step 2 for each instance of the data.

Step 4: Now sort the Euclidean distances in ascending order.

The formula for finding Euclidean distance between two points is given as follows.

For two points p and q, the Euclidean distance between them is calculated by the below formula,

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

$p, q \;\; = two\ points\ in\ Euclidean\ n - space$

$q_i, p_i \;\; = Euclidean\ vectors, starting\ from\ the\ origin\ of\ the\ space\ (initial\ point)$

$n \;\;\;\; = n - space$

Step 5: Then pick the first K entries from the sorted collection.

Step 6: For classification, just perform the majority voting technique (among top k neighbors) to get us the predicted class label.

Step 7: After that perform the above steps multiple times by changing the K values and settle for the exact same value which given us maximum accuracy/performance metrics score. Typically, when K=1 overfits and when K increases the accuracy increases up to a certain point then it again decreases. In this research, K value is 7.

4.3 Deep Learning based Emotion Classification

Deep learning adjusts a multilayer approach to the hidden layers of the neural network. In machine learning approaches, features are defined and extracted either manually or by making use of feature selection methods. In any case, features are learned and extricated automatically in deep learning, achieving better accuracy and performance. Figure 4.6 shows the overview of deep learning technique. deep learning currently provides the best solutions to many problems in the fields of image and speech recognition, as well as in natural language processing. Deep learning technique is discussed in this section.

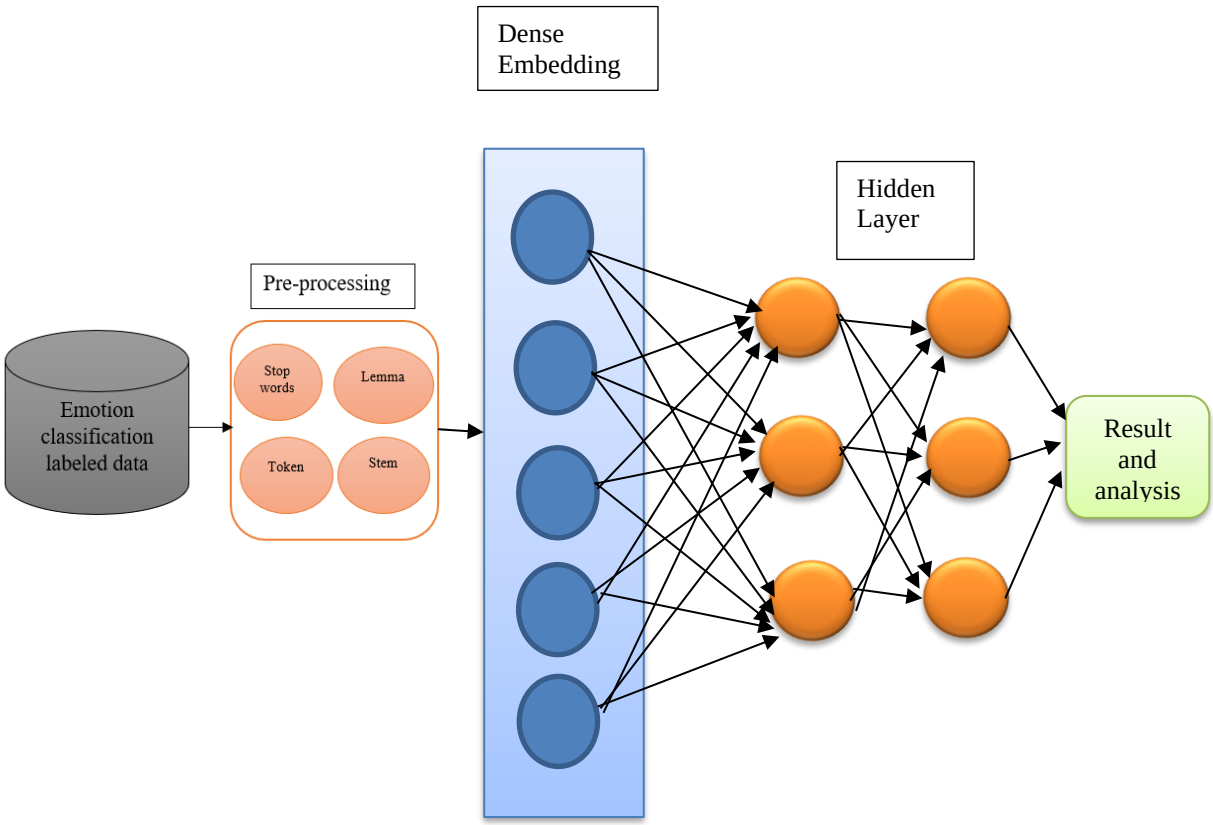

Figure 4.6. Overview of applying deep learning technique

Feature Extraction in Deep Learning

Word Embeddings

Word embeddings are the texts changed into numbers and there may be different numerical representations of the same content. As it turns out, most of the machine learning algorithms and deep learning architectures are unable to process strings or plain text in their raw form [35]. They require numbers as inputs to perform any sort of work, which is classification, regression etc. Moreover, with the huge amount of data that is present within the text format, it is basic to extract knowledge out of it and build applications [35]. So, word embeddings are used for converting all text documents into a numeric format.

Word2vec

It could be a two-layer neural net that processes text [36]. The text corpus take as an input, and its output may be a set of vectors. Whereas it is not a deep neural network it turns text into a numerical form that deep neural network can process. The main purpose and usefulness of Word2vec is to group the vectors of similar words together in vector space [36]. It creates vectors that are distributed numerical representations of word features. It can make highly accurate prediction about a word meaning based on past appearances. Those predictions can be used to establish a words association with other words [36]. It is applied in many streams like sentiment analysis and recommendations in such fields as E-commerce, scientific research, and customer relationship management. For output, Word2vec neural net is a lexicon in which each item contains a vector attached to it, which can be simply queried to detect relationships between words [36].

4.3.1 GRU based Recurrent Neural network

For a deep learning-based solution, gated recurrent unit (GRU) based Recurrent neural network (RNN) is used for multilabel emotion classification. In this thesis, BNet (Binary Neural Network) system is used for solving the multilabel emotion classification problem. Figure 4.7 shows the graphical study of the BNet system architecture. It is divided in 3 categories:

Embedding Module

Encoding Module

Classification Module

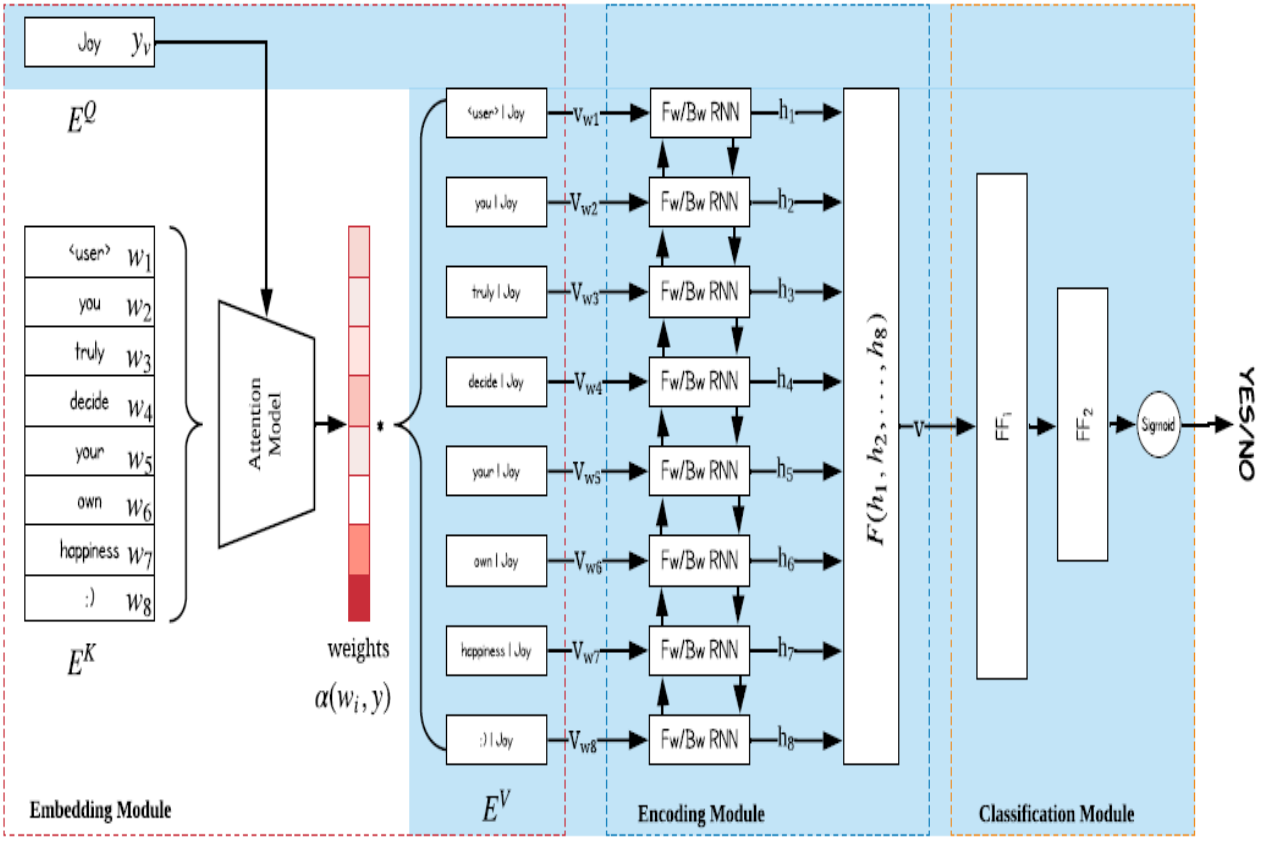

Figure 4.7. BNet system Architecture [6]

Embedding Module

Let (W, y) be the pair of input data to the system, where W = {w_1, w_2, . . . , w_l} is the set of the words which is contain in a tweets and y label is the corresponding to an emotion. The main

objective of the embedding module is to represent each word w_i by a vector v_{wi} and the label by a vector v_y [6].

Encoding Module

The main objective of the encoding module is to map the sequence of word representations $\{v_{w1}, v_{w2}, \ldots, v_{wl}\}$ that is achieved from the embedding module to a single real-valued dense vector. In this work, Recurrent Neural Network (RNN) was used to design the encoder [6].

Classification Module

Classifier is composed of two feed-forward layers with the rectified linear unit (ReLU) activation function followed by a Sigmoid unit [6]. Simple recurrent neural networks are not used because they do not have long term dependencies. The way the gated recurrent unit works is shown in Figure 4.8.

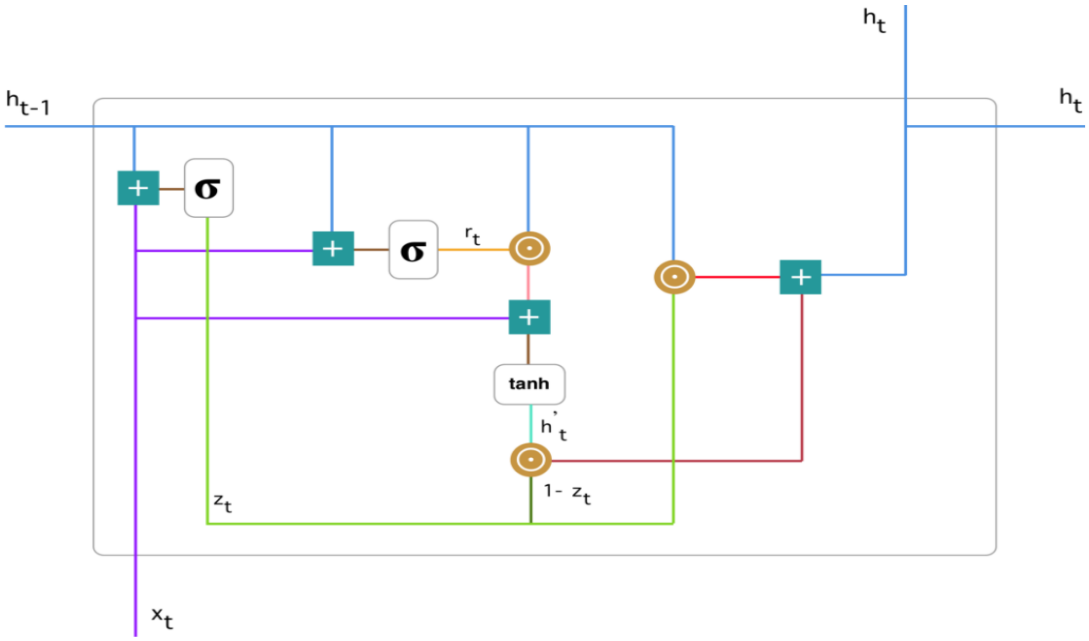

Figure 4.8. Gated Recurrent Unit (GRU) [37]

For solving, the vanishing gradient problem of a standard RNN, Gated Recurrent Unit uses two gates: update gate and reset gate**.** Basically, these are two vectors that decide what sort of

information should be passed to the output. GRUs can be trained on data stored for a long time without removing irrelevant data or cleaning the data. The equation of the update GATE is as follows [37].

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1})$$

The reset gate used to determine how much past information the model should forget. The equation is given below [37].

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1})$$

The memory gate is used to tell the model how much previous data the model should remember. The equation is given below [37].

$$h'_t = \tan h\,(Wx_t + r_t \odot Uh_{t-1})$$

As the final step, the network needs to calculate the h'$_t$ vector which holds data for the current unit and passes it down to the network. In order to, the update gate is needed. It decides what to collect from the current memory content. The equation is given below [37].

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t$$

A trained word embeddings model has been used for training the whole multilabel emotion recognition model. Attention is motivated by how pay visual attention to distinct regions of correlate words in one sentence. Attention becomes an increasingly well concept and valueable tool in developing deep learning models. attention was also used for getting the best out of the GRU model, the emotion scores with respect to all tweets which is discuss in Chapter 5. Adam and Rmsprop optimizers were used to train the same model to get a higher accuracy.

Chapter 5

Results and Discussion

In this Chapter, the experimental results are discussed for the research of emotion classification. Various algorithms have been evaluated and the performance metrics compared. Finally, ANOVA: one-way statistical analysis was performed on all techniques.

5.1 Evaluation parameters

The following evaluation metrics were used to evaluate the performance of the classifiers.

Accuracy: It is a ratio of correctly predicted emotion class to the total number of observation emotion class.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Precision: It is a ratio of correctly predicted emotion class to the total number of positive predicted class.

$$Precision = \frac{TP}{TP + FP}$$

Recall: It is a ratio of correctly predicted positive emotion class to all observation in true actual class.

$$Recall = \frac{TP}{TP + FN}$$

F1 score: F1 score is the degree of calculating the weighted average of precision and recall. It ranges between 0 to 1 and it is considered perfect when it is 1 which means that the model has low false positives and low false negatives.

$$F1\ score = 2 * \frac{recall * precision}{recall + precision}$$

Confusion Matrix: A confusion matrix is used for summarizing the performance of a

classification algorithm.

Table 5.1. Confusion matrix for two-class classification problem

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Where, TP=True Positive

FP=False Positive

TN=True Negative

FN=False Negative

5.2 Results

Tables 5.2-5.7 show the Multi-label classification of test sentences from emotion classification dataset [21] using different machine learning and deep learning methods. The test sentences are classified in 11 category of emotion label which is Joy, sadness, anger, fear, trust, disgust, surprise, anticipation, optimism, pessimism, and love.

Table 5.2. Multilabel classification of test sentences from emotion classification dataset using Naïve Bayes

Tweets	Joy	Sadness	Anger	Fear	Trust	Disgust	Surprise	Anticipation	Optimism	Pessimism	Love
@Adnan__786__ @AsYouNotWish Dont worry Indian army is on its ways to dispatch all Terrorists to Hell	0.6668	0.0056	0.662	0.9402	0.0612	0.0017	0.1337	0.004	0.09	0.00002	0.005
Academy of Sciences, eschews the normally sober tone of scientific papers and calls the massive loss of wildlife a â€œbiological annihilation	0.411	0.6992	0.2292	0.0046	0.4886	0.0107	0.3365	0.017	0.314	0.2411	0.004
I blew that opportunity -__- #mad	0.1874	0.0422	0.2453	0.3149	0.3815	0.1021	0.4707	0.53	0.2376	0.069	0.06

Table 5.2 shows the emotion scores for each emotion for a test sentence as labeled by Naïve Bayes classifier. The highest percentage of the emotion label indicates the most appropriate emotion label for the test sentence. For example, in the first test sentence, we can see that fear emotion gives the highest percentage which shows that fear is the dominant emotion in the sentence.

Table 5.3. Multilabel classification of test sentences from emotion classification dataset using Support vector machine (SVM)

Tweets	Joy	Sadness	Anger	Fear	Trust	Disgust	Surprise	Anticipation	Optimism	Pessimism	Love
@Adnan__786__ @AsYouNotWish Dont worry Indian army is on its ways to dispatch all Terrorists to Hell	0.6789	0.3037	0.7363	0.6859	0.1475	0.3148	0.3895	0.3358	0.2806	0.2197	0.3591
Academy of Sciences, eschews the normally sober tone of scientific papers and calls the massive loss of wildlife a â€œbiological annihilation	0.5192	0.3992	0.3242	0.2107	0.5818	0.2694	0.3927	0.2277	0.4338	0.3115	0.2358
I blew that opportunity -__- #mad	0.3585	0.2693	0.3648	0.4682	0.3385	0.2648	0.5146	0.416	0.3463	0.2987	0.3369

Table 5.3 shows the emotion scores for each emotion for a test sentence as labeled by the Support Vector Machine (SVM) classifier. In the first test sentence, we can observe that joy, anger and fear emotions give the highest percentage which shows that these emotions are the most dominant in the sentence.

Table 5.4. Multilabel classification of test sentences from emotion classification dataset using Random forest

Tweets	Joy	Sadness	Anger	Fear	Trust	Disgust	Surprise	Anticipation	Optimism	Pessimism	Love
@Adnan__786__ @AsYouNotWish Dont worry Indian army is on its ways to dispatch all Terrorists to Hell	0.6703	0.2961	0.6136	0.647	0.0505	0.0169	0.3993	0.0328	0.0625	0.002	0.0272
Academy of Sciences, eschews the normally sober tone of scientific papers and calls the massive loss of wildlife a â€œbiological annihilation	0.2071	0.148	0.1804	0.036	0.3818	0.0221	0.3729	0.0299	0.2327	0.0765	0.0146
I blew that opportunity -__- #mad	0.164	0.001	0.148	0.268	0.5846	0.3	0.6913	0.2346	0.1188	0.066	0.1734

Table 5.4 shows the emotion scores for each emotion for a test sentence as labeled by the Random Forest classifier. In first test sentence, we can observe that joy and fear emotions give the highest percentage which shows that these emotions are the most dominant in the test sentence.

Table 5.5. Multilabel classification of test sentences from emotion classification dataset using K-nearest neighbor (KNN)

Tweets	Joy	Sadness	Anger	Fear	Trust	Disgust	Surprise	Anticipation	Optimism	Pessimism	Love
@Adnan__786__ @AsYouNotWish Dont worry Indian army is on its ways to dispatch all Terrorists to Hell	0.5928	0.0189	0.6384	0.6273	0.0516	0.0182	0.4352	0.0415	0.0528	0.0011	0.0027
Academy of Sciences, eschews the normally sober tone of scientific papers and calls the massive loss of wildlife a â€œbiological annihilation	0.2537	0.1849	0.1637	0.1829	0.3928	0.0252	0.3628	0.0371	0.2129	0.0373	0.0183
I blew that opportunity -__- #mad	0.1825	0.0143	0.1294	0.1193	0.2819	0.32	0.7139	0.2983	0.1482	0.0459	0.018

Table 5.5 shows the emotion scores for each emotion for a test sentence as labeled by the K-nearest neighbor (KNN) classifier. In last test sentence, we can observe that surprise emotion gives the highest percentage which shows that surprise emotion is the most dominant in the sentence.

Table 5.6. Multilabel classification of test sentences from emotion classification dataset using GRU based RNN with Adam optimizer

Tweets	Joy	Sadness	Anger	Fear	Trust	Disgust	Surprise	Anticipation	Optimism	Pessimism	Love
@Adnan__786__ @AsYouNotWish Dont worry Indian army is on its ways to dispatch all Terrorists to Hell	0.976	0.57	0.642	0.811	0.201	0.8702	0.9564	0.2364	0.9247	0.1109	0.514
Academy of Sciences, eschews the normally sober tone of scientific papers and calls the massive loss of wildlife a â€œbiological annihilation	0.38	0.2275	0.6071	0.6053	0.523	0.3137	0.5218	0.0821	0.7771	0.7127	0.038
I blew that opportunity -__- #mad	0.576	0.3246	0.776	0.473	0.545	0.2775	0.7479	0.4745	0.0314	0.684	0.106

Table 5.6 shows the emotion scores for each emotion for a test sentence as labeled by the GRU based RNN with Adam optimizer classifier. In first test sentence, we can observe that joy, surprise and optimism emotions gives the highest percentage which shows that these emotions are the most dominant in the sentence.

Table 5.7. Multilabel classification of test sentences from emotion classification dataset using GRU based RNN with RmsProp optimizer

Tweets	Joy	Sadness	Anger	Fear	Trust	Disgust	Surprise	Anticipation	Optimism	Pessimism	Love
@Adnan__786__ @AsYouNotWish Dont worry Indian army is on its ways to dispatch all Terrorists to Hell	0.86	0.9086	0.402	0.561	0.0217	0.9346	0.5899	0.1563	0.9017	0.9587	0.799
Academy of Sciences, eschews the normally sober tone of scientific papers and calls the massive loss of wildlife a â€œbiological annihilation	0.22	0.1866	0.27	0.35	0.7962	0.9585	0.9795	0.0269	0.2631	0.2438	0.414
I blew that opportunity -__- #mad	0.476	0.2071	0.996	0.631	0.6762	0.8718	0.1418	0.9876	0.4613	0.5357	0.181

Table 5.7 shows the emotion scores for each emotion for a test sentence as labeled by the GRU based RNN with RmsProp optimizer classifier. In first test sentence, we can observe that sadness, disgust and pessimism emotions give the highest percentage which shows that these emotions are the most dominant in the sentence.

Tables 5.8 – 5.13 present the performance metrics (precision, Recall and F1-score) for each classifier. All the metrics are presented for each category of emotion. In Tables 5.8 – 5.13, we can observe that some emotion scores are high and some emotion scores are low in the performanance matrix. Figure 5.1 shows that the data for all the emotions is not balanced and it can be noticed that some of the emotions like pessimism and love have very less data. This means that the number of training samples for few emotions are very less compared to others. This may create some variation in the performance metrics (precision, recall and F1-score) of the emotions. Also sometimes models might not be able to learn some of the emotions due to errors in the labeled training data or not find enough patterns in the data corresponding to those emotions.

Table 5.8. Performance matrix using Naïve Bayes classifier

Naïve Bayes			
Emotion	Precision	Recall	F1-score
Joy	0.748	0.738	0.699
Sadness	0.787	0.862	0.811
Anger	0.716	0.72	0.681
Fear	0.847	0.868	0.841
Trust	0.758	0.691	0.652
Disgust	0.841	0.857	0.814
Surprise	0.723	0.674	0.575
Anticipation	0.813	0.881	0.832
Optimism	0.74	0.721	0.626
Pessimism	0.909	0.947	0.923
Love	0.921	0.951	0.931

Table 5.9. Performance matrix using Support vector machine classifier

Support vector machine			
Emotion	Precision	Recall	F1-score
Joy	0.73	0.738	0.732
Sadness	0.796	0.851	0.815
Anger	0.7	0.706	0.702
Fear	0.851	0.869	0.855
Trust	0.736	0.732	0.725
Disgust	0.848	0.867	0.845
Surprise	0.7	0.711	0.699
Anticipation	0.823	0.869	0.839
Optimism	0.714	0.73	0.718
Pessimism	0.927	0.948	0.927
Love	0.911	0.948	0.928

Table 5.10. Performance matrix using Random forest classifier

Random forest			
Emotion	Precision	Recall	F1-score
Joy	0.75	0.756	0.739
Sadness	0.791	0.866	0.811
Anger	0.713	0.725	0.707
Fear	0.868	0.882	0.866
Trust	0.721	0.72	0.716
Disgust	0.845	0.863	0.834
Surprise	0.657	0.676	0.656
Anticipation	0.827	0.881	0.837
Optimism	0.733	0.75	0.719
Pessimism	0.928	0.948	0.926
Love	0.908	0.951	0.929

Table 5.11: Performance matrix using K-nearest neighbor classifier

K-nearest neighbor			
Emotion	Precision	Recall	F1-score
Joy	0.741	0.666	0.539
Sadness	0.756	0.87	0.809
Anger	0.699	0.67	0.537
Fear	0.822	0.853	0.792
Trust	0.576	0.465	0.35
Disgust	0.846	0.852	0.797
Surprise	0.583	0.384	0.283
Anticipation	0.822	0.884	0.832
Optimism	0.683	0.707	0.59
Pessimism	0.951	0.948	0.923
Love	0.908	0.953	0.93

Table 5.12. Performance matrix using GRU based RNN with Rmsprop optimizer

GRU based RNN with RmsProp optimizer			
Emotion	Precision	Recall	F1-score
Joy	0.548	0.658	0.598
Sadness	0.584	0.609	0.596
Anger	0.559	0.627	0.59
Fear	0.549	0.598	0.572
Trust	0.62	0.638	0.629
Disgust	0.618	0.659	0.639
Surprise	0.573	0.63	0.60
Anticipation	0.575	0.624	0.599
Optimism	0.581	0.66	0.618
Pessimism	0.588	0.632	0.609
Love	0.565	0.615	0.589

Table 5.13. Performance matrix using GRU based RNN with Adam optimizer

GRU based RNN with Adam optimizer			
Emotion	Precision	Recall	F1-score
Joy	0.515	0.459	0.485
Sadness	0.53	0.458	0.491
Anger	0.521	0.457	0.487
Fear	0.491	0.416	0.45
Trust	0.562	0.46	0.506
Disgust	0.562	0.452	0.501
Surprise	0.522	0.461	0.49
Anticipation	0.50	0.469	0.484
Optimism	0.519	0.445	0.479
Pessimism	0.555	0.465	0.506
Love	0.502	0.437	0.498

The results of all the methods are summarized in the Table 5.14. Notice that RNN with RmsProp achieved high accuracy of 82.3% even though the other methods performed well in terms of the other metrics.

Table 5.14. Mean value of evaluation results of all emotions

Mean value for all methods				
Methods	Accuracy	Precision	Recall	F1-score
Naïve Bayes	0.809	0.80	0.812	0.762
SVM	0.815	0.794	0.815	0.798
Random Forest	0.819	0.794	0.82	0.794
KNN	0.757	0.762	0.75	0.67
RNN with Adam optimizer	0.79	0.526	0.452	0.486
RNN with RmsProp optimizer	0.823	0.596	0.632	0.595

5.3 Discussion

A total of 10983 tweets dataset has been used as a part of this study. Each tweet exhibits multiple emotions making the problem multi-label emotion classification. The following are the 11 different emotions that are present in the tweets: Joy, Sadness, Anger, Fear, Trust, Disgust, Surprise, Anticipation, Optimism, Pessimism, and Love. The distribution of each of these emotions in the dataset is as shown in the Figure 5.1.

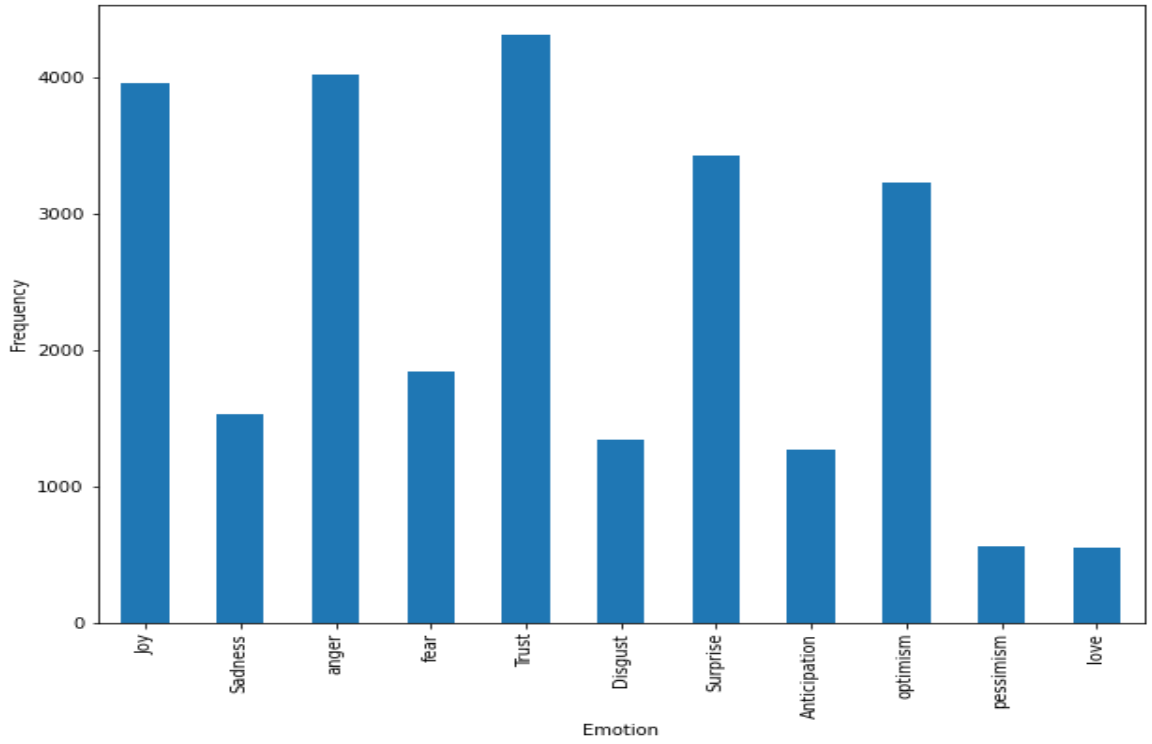

Figure 5.1. Distribution of various emotions present in the tweet dataset

This data has been split into 3 parts, training, validation, and test dataset and each comprising of 6838 (62%), 886 (8%), and 3259 (30%) tweets data respectively. The algorithms were trained on the train dataset and validation dataset has been used to fine-tune the parameters of the models. Various algorithms discussed in Chapter 4 have been implemented on this dataset and performance metrics have been captured. The following classifiers were used for the evaluation: Naïve Bayes, Support Vector Machine (SVM), Random Forest, K-Nearest Neighbor (KNN), GRU based RNN model with RMS Prop optimizer, and GRU based RNN with Adam optimizer.

Figure 5.2 shows the accuracy of different machine learning and deep learning models. From the multi-label emotion classification (SemEval-2018) dataset, 30% of the data was used for the testing the model and remaining data was used for training and validation.

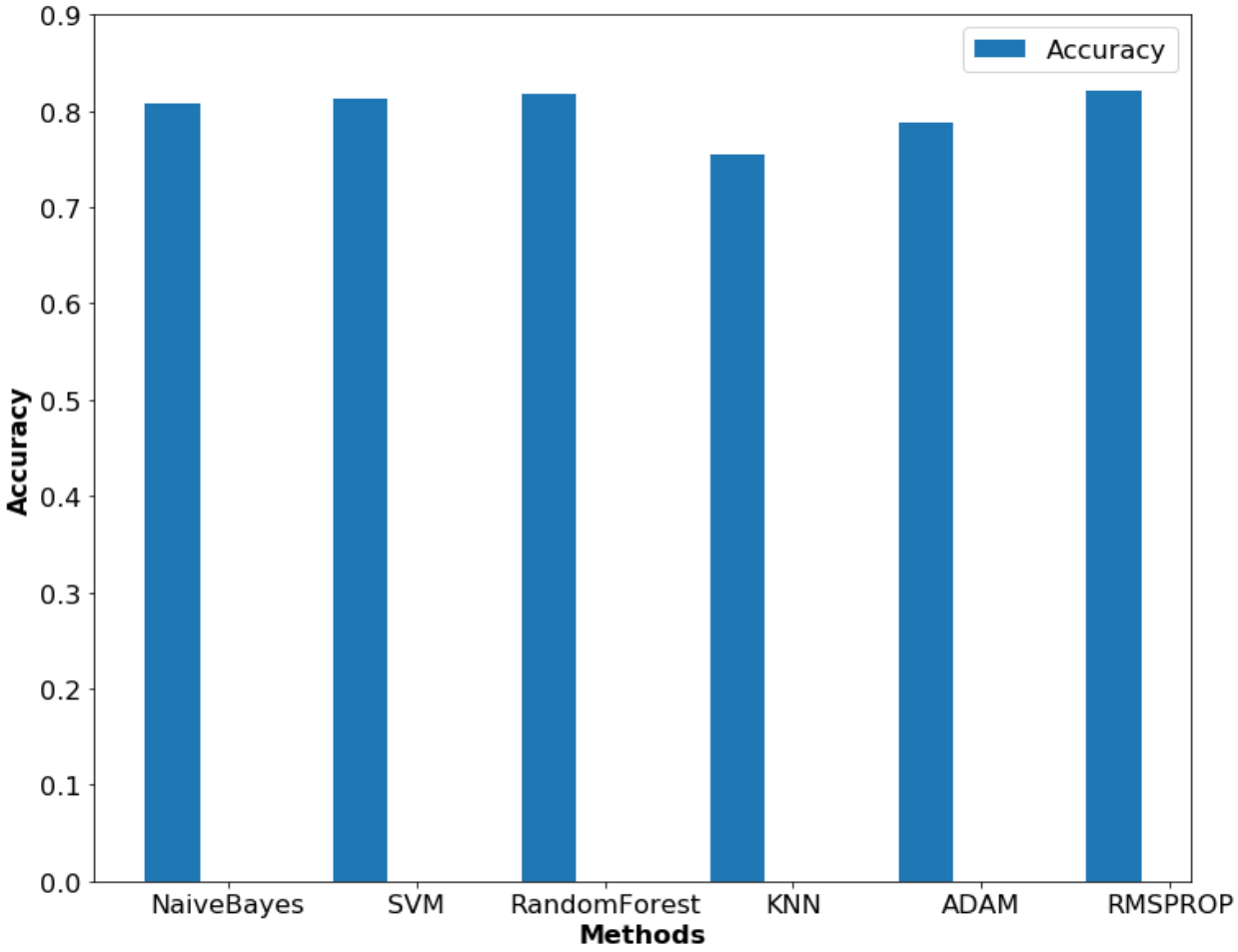

Figure 5.2. Accuracy of different models

The most commonly used performance evaluation metrics for classification problems are: accuracy, Precision, recall and F1 score. Evaluation parameters are measured with the help of confusion matrix. The data was evaluated on the previous mentioned classifiers and the performance metrics were compared.

Figure 5.3 shows that the Naïve Bayes classifier achieved the best performance with respect to precision (0.80) on average of all emotions. Moreover, K-nearest Neighbor (KNN) method has high precision for Pessimism (0.951) emotion compared to the other methods but did not perform well overall compared to Naïve Bayes. For precision, machine learning methods achieved better result compared to deep learning methods. For deep learning models, GRU based RNN with RmsProp optimizer (0.59) performed well compare to Adam optimizer (0.52).

Figure 5.3 shows precision of the classifiers for each emotion category.

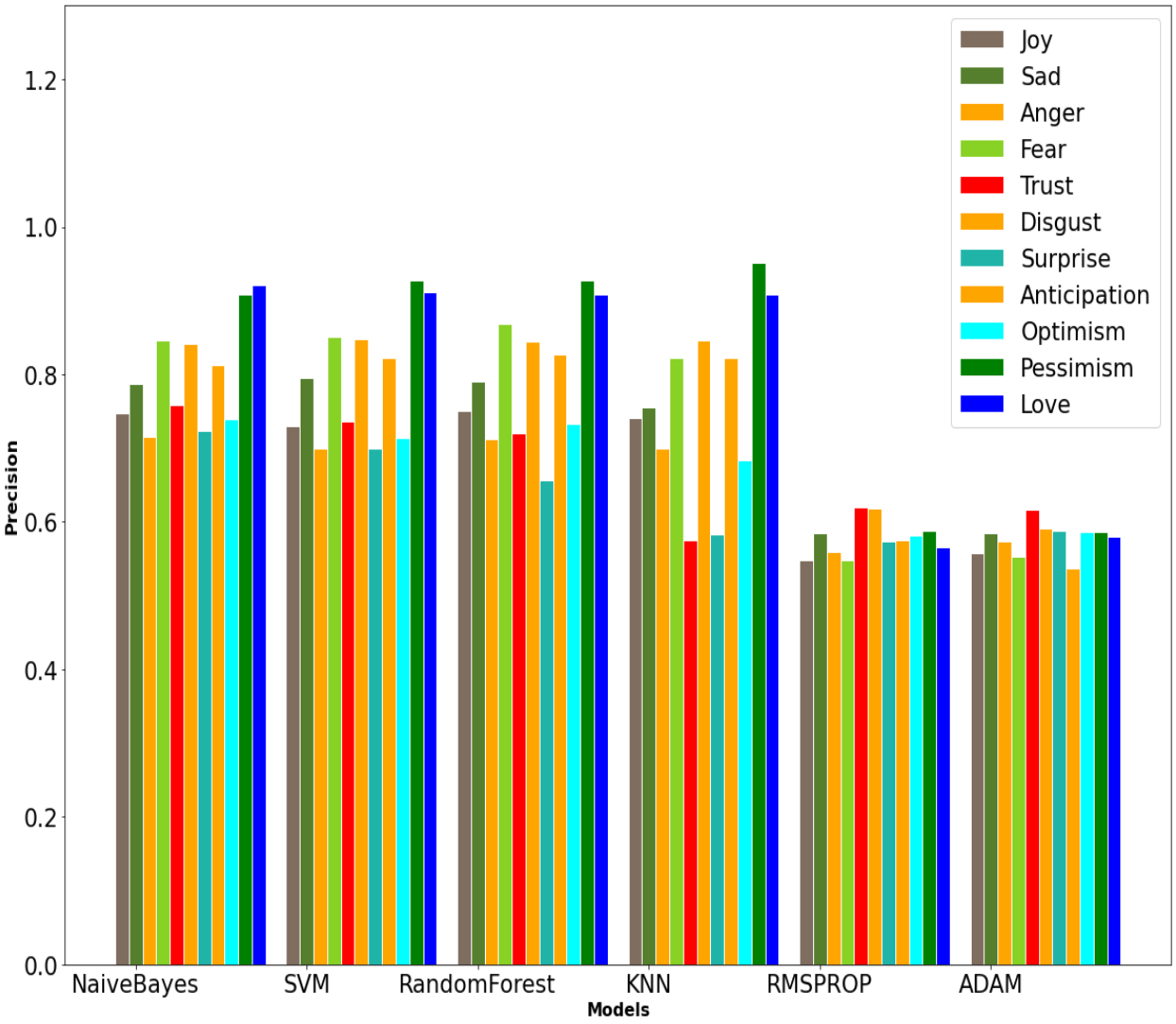

Figure 5.3. Precision of various algorithms at emotion category

Figure 5.4 shows that the Random Forest classifier achieved the best performance with respect to recall (0.819) for average of all emotions. Also, SVM and Naïve Bayes perform well with a recall of 0.81 and 0.815, respectively. Moreover, K-nearest Neighbor (KNN) classifier has low recall value for trust (0.465) and surprise (0.384) emotion but overall KNN performed well with an average recall of 0.749. For deep learning methods, GRU based RNN with RmsProp optimizer (0.632) performed well compare to Adam optimizer (0.452). Figure 5.4 shows the recall of the classifiers for each emotion category.

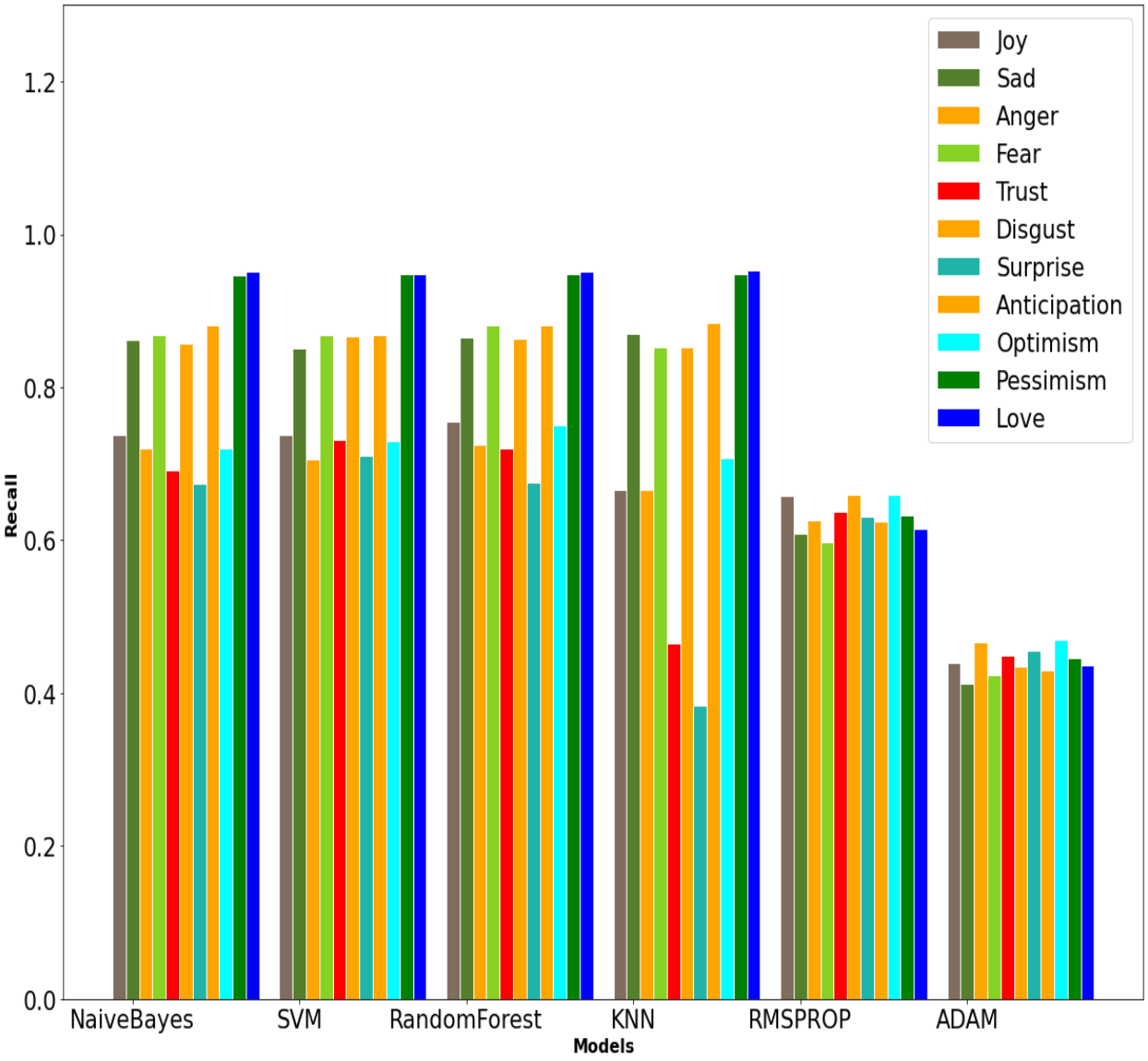

Figure 5.4. Recall of algorithms at emotion category

Figure 5.5 shows that the support vector machine (SVM) classifier achieved the best performance with respect to F1 score (0.798) for average of all emotions. Moreover, K-nearest Neighbor (KNN) classifier has quite low result (0.671) compared to Random Forest (0.794), Naïve Bayes (0.762), and SVM. For deep learning models, both the models performed similar in all emotions. But GRU based RNN with RmsProp optimizer (0.595) performed well compare to Adam optimizer (0.486). Figure 5.5 shows F1 score of the classifiers for each emotion category.

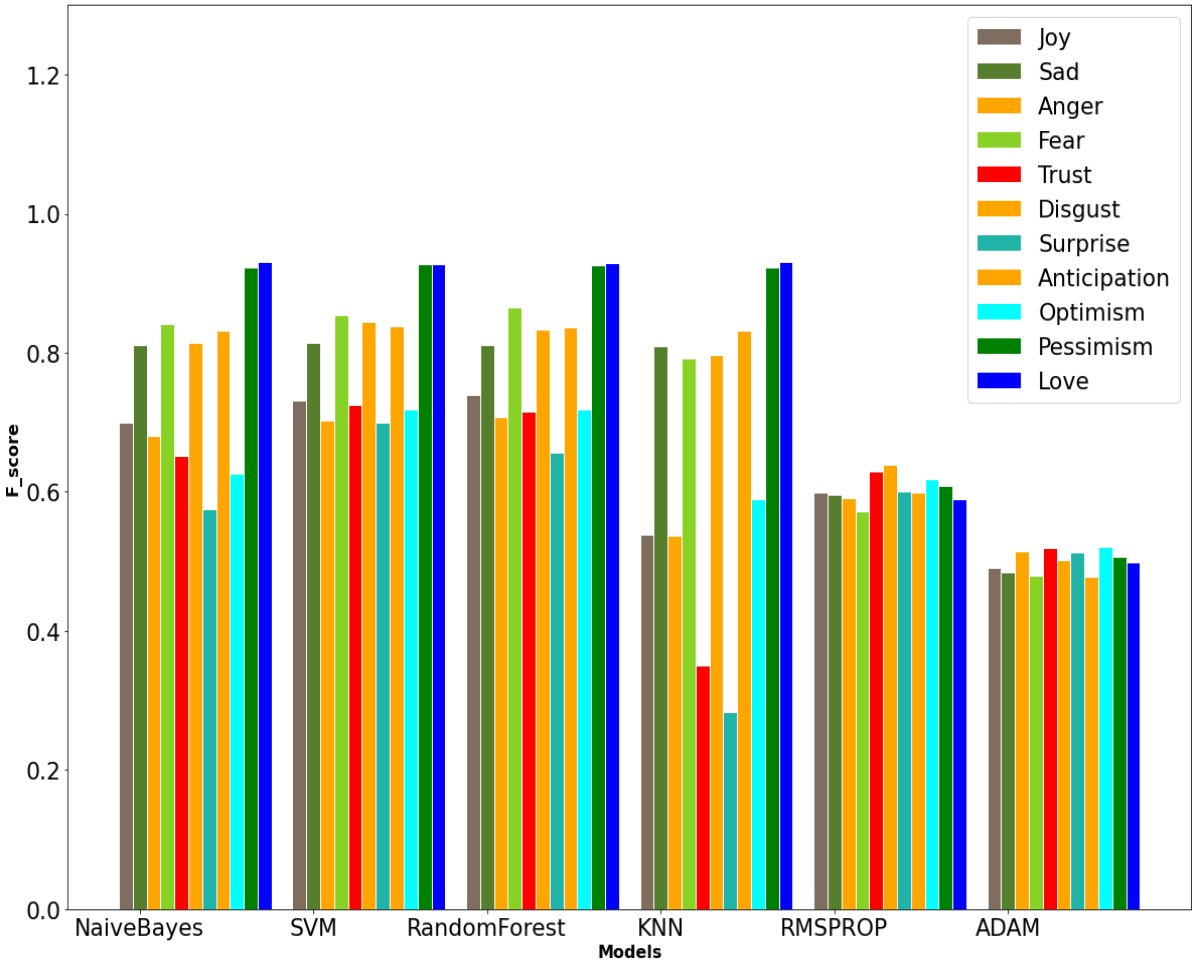

Figure 5.5. F1 score of algorithms at emotion category

Notice that GRU based RNN with RmsProp, Random Forest and SVM perform relatively better over other methods. The efficacy of the task is achieved through the ensemble modelling. In ensemble modelling, the predictions of different models are combined to produce improved performance over any individual model in classifying the emotions. This approach helps in reducing the variance and improves the generalization. The following two popular ensemble techniques have been used in this study: (i) **majority voting**, and (ii) **weighted average.**

In majority voting approach, predictions of different algorithms have been combined and the majority vote is predicted. In weighted average approach, predictions of algorithms have been combined with certain weightage. The weightage of each algorithm is generally assigned based

on the individual performance of that algorithm on the data. In this research, F1 score of the algorithm is considered to be its weight.

The ensemble methods combine the predictions of all the other methods to produce an improved prediction. These ensemble methods considered in this research are parallel in nature which means all the models are independent of each other. Figure 5.6 shows that both ensemble techniques (majority voting and weighted average) achieved the best result with respect to precision (0.818, 0.813), recall (0.829, 0.83) and F1 score (0.789, 0.799) for average of all emotions respectively. Moreover, both the ensemble techniques perform better than any individual method. Figure 5.6 compares performance metrics of ensemble methods against other individual algorithms.

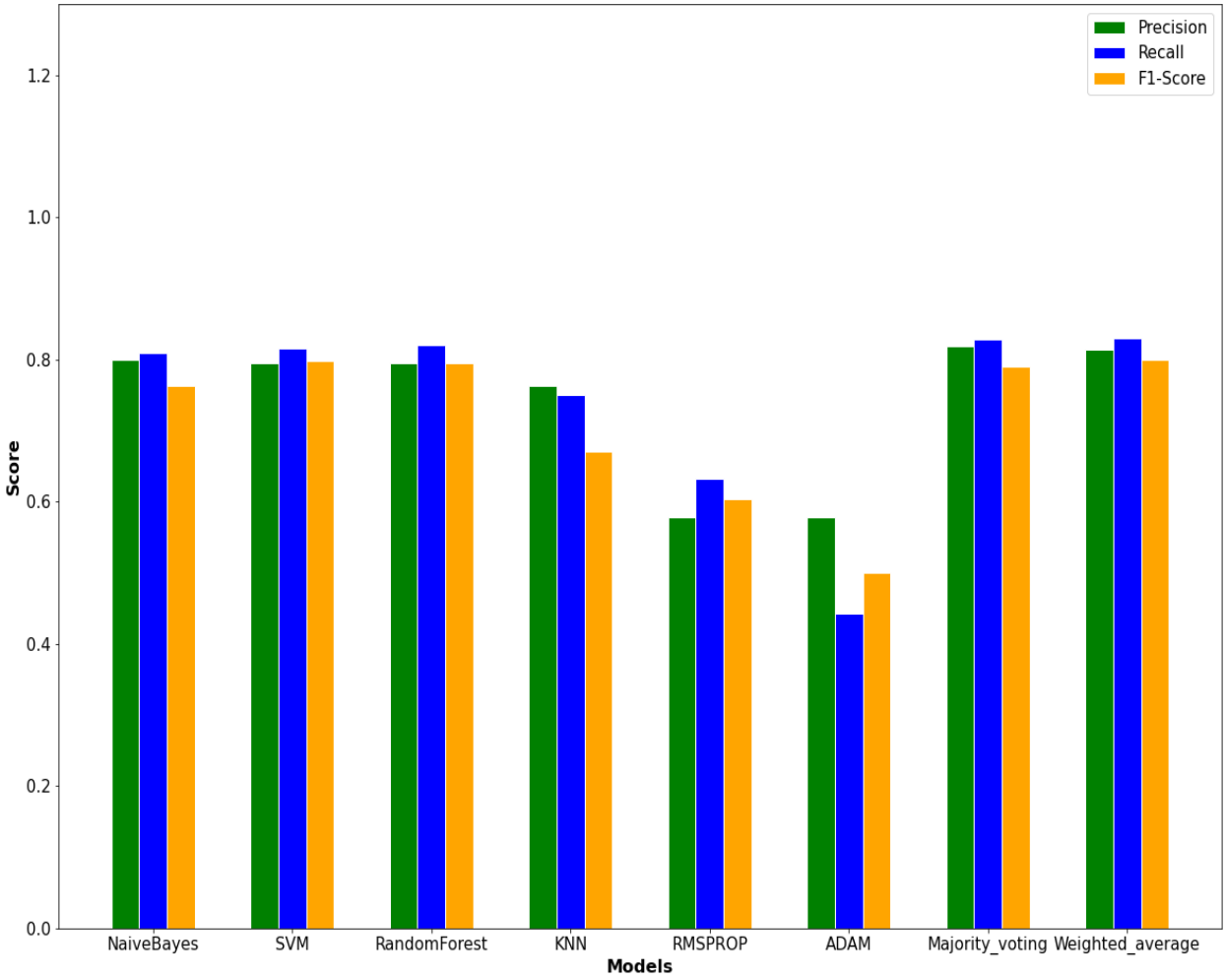

Figure 5.6. Comparison of performance metrics of algorithms against ensemble methods

Despite achieving greater performance, sometimes these models are given to sampling errors. To check the robustness of the model, similar experiments were performed by shuffling the whole data and re-splitting it to train, validation, and test sets.

For precision and accuracy, majority voting method achieved better results compared to weighted average method in all 3 experiments. The mean value of precision is 0.819 and 0.814 for majority voting and weighted average methods respectively. Also, the mean value of accuracy is 0.811 and 0.799 for majority voting and weighted average methods respectively. For recall and F1 score, weighted average method achieved better result compare to majority voting method in all 3 experiments. The mean value of recall is 0.829 and 0.832 for majority voting and weighted average methods respectively. Also, the mean value of F1 score is 0.789 and 0.802 for majority voting and weighted average methods respectively. Overall, the performance metrics did not change much in the different experiments. It is important to know the significance of this variation in the results. The combined results of various experiments have been presented in the Table 5.15.

Table 5.15. Comparison of Performance metrics for ensemble methods for 3 different experiments

Metric	**Majority Voting method**			**Weighted Average method**		
	Experiment 1	**Experiment 2**	**Experiment 3**	**Experiment 1**	**Experiment 2**	**Experiment 3**
Precision	0.818	0.815	0.823	0.813	0.816	0.813
Recall	0.829	0.831	0.827	0.83	0.834	0.833
F1 Score	0.789	0.793	0.785	0.799	0.805	0.802
Accuracy	0.829	0.815	0.79	0.829	0.797	0.77

From table 5.15, it is clear that the performance metrics haven't changed much for the changes in the sampling data. To conclude, or to choose the best method from these ensemble methods as well as all classifiers, statistical one-way ANOVA test was performed. Test for statistical significance helps to measure whether the difference between the performance metrics observed via all methods is significant or not.

In this research, One-way Analysis of Variance (ANOVA) test is performed on the mean values of performance metrics on all the methods (shown in Table 5.16). The null hypothesis (H0) states that all models demonstrate similar performance. H0 is accepted if no statistically significant difference ($P > 0.05$) is observed in the mean value of the performance metrics for the different models under study. The alternate hypothesis (H1) is accepted and H0 is rejected if a statistically significant performance difference ($P < 0.05$) is found to exist [38]. One-way ANOVA is an omnibus test and needs a post-hoc study to identify all the methods demonstrating this statistically significant performance differences [38].

Table 5.16. ANOVA test results on performance metrics

Metric	Naïve Bayes	SVM	Random Forest	KNN	RmsProp	Adam	Majority voting Method mean	Weighted average method mean	P-value
Precision	0.80	0.798	0.80	0.736	0.607	0.539	0.819	0.814	$6.85*10^{-9}$
Recall	0.812	0.819	0.824	0.763	0.588	0.463	0.829	0.832	$1.72*10^{-8}$
F1 Score	0.766	0.80	0.801	0.70	0.581	0.497	0.789	0.802	$1.36 * 10^{-14}$
Accuracy	0.812	0.819	0.824	0.763	0.827	0.795	0.817	0.805	$1.4 * 10^{-5}$

Table 5.16 summarizes the ANOVA test results for performance metrics. It is observed that the P -values are lower than 0.05 for the performance metrics. This means that the methods are statistically significant (null hypothesis H_0 is rejected) when evaluated on the basis of these performance metrics. F1 score is the consonant mean of both precision and recall. It is a better measure of incorrectly classified cases and used when it needs to maintain higher precision and recall instead of just focussing on one. In this study, the mean value of F1 score is higher for weighted average ensemble method (0.802) compared to that of majority voting ensemble method (0.789). This shows, that weighted average method has proved to be the best model in view of achieving higher F1 score and model built using weighted average method would result in higher F1 score over other methods.

Receiver Operating Characteristics (ROC) analysis

In this analysis, Area Under the ROC curve (AUC) of the methods has been computed under consideration. ROC curve is a graphical plot that determines the ability of the classifier as the threshold is varied.

First, all the prediction probabilities of the methods were computed. Prediction probability is a float value in the range of 0 and 1 which represents probability of the corresponding emotion present in the given tweet. Once these probabilities are known, we vary the threshold between 0 and 1 to classify each emotion. For each threshold value we compute True Positive rate and False positive rate as shown in Figure 5.7.

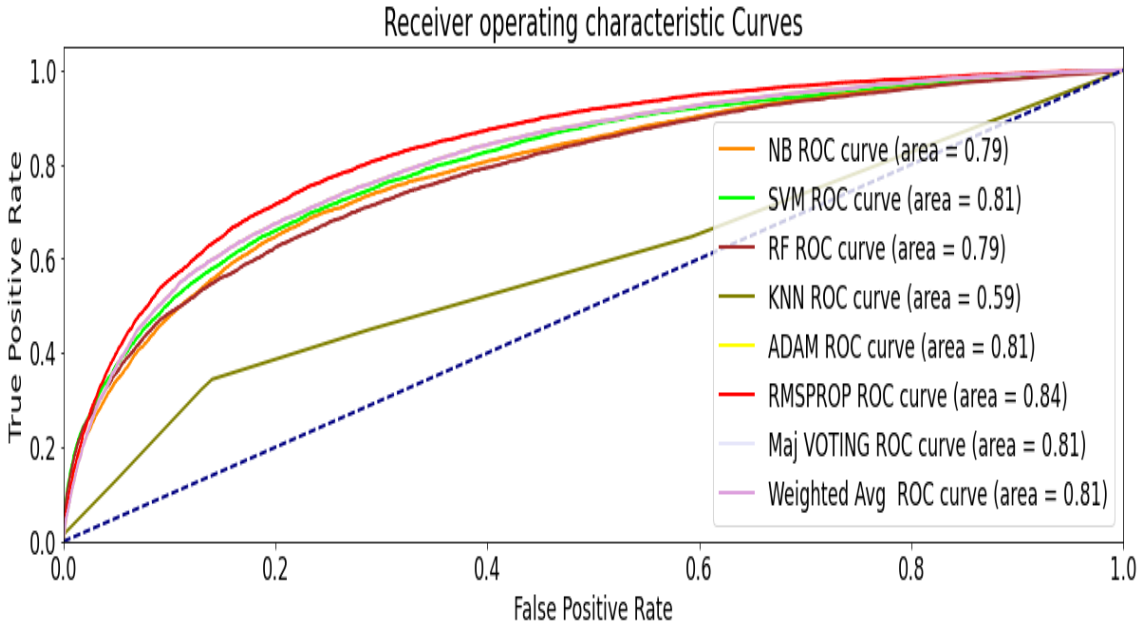

Figure 5.7. ROC analysis for all methods

After that, the area under the ROC curve was computed, AUROC metric which represents the summary of ROC analysis in a single value. Higher the AUROC more robust or accurate the model is. Figure 5.7 shows all corresponding AUROC values for the models. It can be noted RMSPROP optimizer based Deep learning approach has higher value of 0.84 compared to the rest of the models.

5.4 Results Comparison

Different machine learning and deep learning methods were used for solving multi-label emotion classification problem. The evaluation parameters such as accuracy, precision, recall, and F1-score results obtained using Naïve Bayes, SVM, Random forest, KNN, GRU based RNN with Adam and Rmsprop classifiers. Mohammed et al. [6] achieved 0.59 accuracy, 0.57 precision, 0.61 recall, and 0.56 F1 score using GRU based RNN classifier, which was used in this research as a reference. In comparison, different classifiers were used for measuring all evaluation parameters for emotion classification labeled data set. GRU based RNN with RmsProp optimizer classifier gave high accuracy for multi-label emotion classification from emotion classification dataset (SemEval-2018), even though, other methods give better performance. Table 5.17 shows that the comparison of all methods for emotion classification dataset.

Table 5.17. Comparison of all methods

Number	**Parameters**	**Naïve Bayes**	**SVM**	**Random Forest**	**KNN**	**GRU based RNN with Adam Optimizer**	**GRU based RNN with RmsProp Optimizer**
1	**Accuracy**	0.809	0.815	0.819	0.757	0.79	**0.823**
2	**Precision**	**0.80**	0.794	0.794	0.762	0.526	0.596
3	**Recall**	0.812	0.815	**0.82**	0.75	0.452	0.632
4	**F1-score**	0.762	**0.798**	0.794	0.67	0.486	0.595
5	**AUC**	0.79	0.81	0.79	0.59	0.81	**0.84**

Overall, the better performance is achieved by using machine learning methods for all evaluation parameters. But GRU based RNN with Rmsprop optimizer performed the best in

terms of accuracy, with the highest accuracy (0.823) compared to other classifiers. The results also show a huge improvement compared to the results of Mohammed et al. [6] for the same dataset.

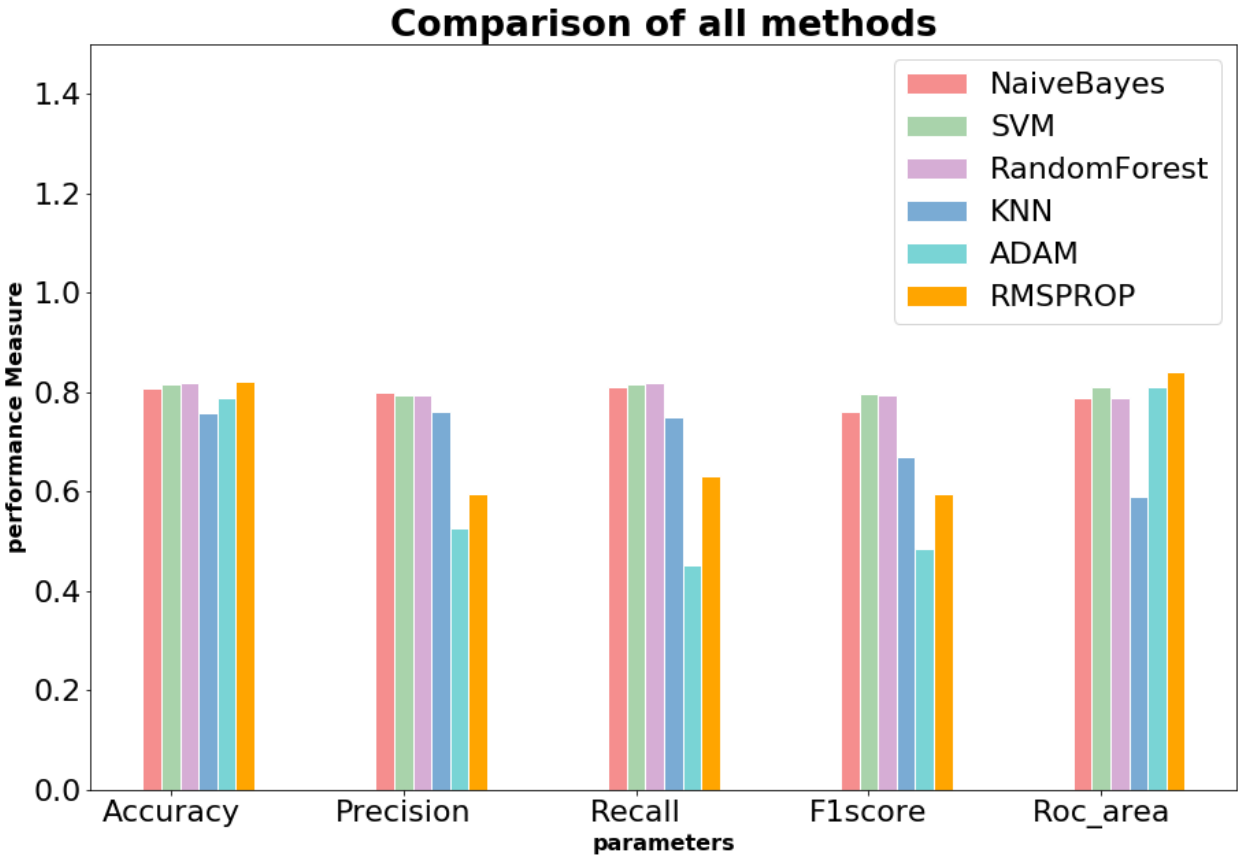

Figure 5.8. Comparison of all methods

Figure 5.8 shows that comparison of all evaluation parameters using different classifiers. All the classifiers performed well for accuracy. Also, the value for AUC is 0.84 (84%) for GRU based RNN with RmsProp optimizer, which is highest compared to other classifiers. For other evaluation parameters (precision, recall and F1 score), machine learning methods give better results compared to deep learning methods.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this research, twitter data was analysed for emotion classification. Since each tweet is associated with multiple emotions not just limited to one, this problem has been formulated as multi-label emotion classification. The popular machine learning classifiers such as Naïve Bayes, Random Forest, Support Vector Machines (SVM), K-Nearest Neighbor (KNN) were used to solve multi-label emotion classification problem. Also, GRU (Gated Recurrent Unit) based Recurrent Neural Network (RNN) with Adam and RmsProp optimizer were used to solve multi-label emotion classification problem.

The popular ensemble techniques such as Majority voting and Weighted average methods were used for reducing the variance and improve the generalization. These methods have been proved to be more accurate in terms of all the performance metrics (accuracy, precision, recall, and F1 score). Also, One-way Analysis of Variance (ANOVA) test is performed on the mean values of performance metrics on all the methods.

From the results, it is concluded that accuracy increased from 0.59 to 0.823 using GRU based RNN with RmsProp optimizer classifier which is 23.3% (0.233) higher, precision increased from 0.57 to 0.80 using Naive Bayes classifier which is 23% (0.23) higher, recall increased from 0.56 to 0.82 using Random Forest classifier which is 26% (0.26) more and F1 score increased from 0.56 to 0.798 using Support Vector machines (SVM) which is 23.8% (0.238) higher than Mohammed et al. [6] research paper results on emotion classification dataset (SemEval-2018). Highest value of AUC (0.84) was achieved for GRU based RNN with RmsProp optimizer. For visualization, Matplotlib library was used in Jupyter Notebook to

compare all the results using machine learning and deep learning methods.

6.2 Future Work

In the future, the present analysis can be extended by adding more feature extraction parameters and different models can be applied and tested on different datasets. The present research focusses on establishing the relations between the tweet and emotion labels. More research can be done in the direction of exploring relations between the phrases of tweet and emotion label. Transfer learning with some existing pre-trained models for classification and data fusion from different data sources can be a good direction to explore to improve the robustness and accuracy. In this study, dataset comes from only twitter source, but other social networks can be used for creating this type of dataset. For this research, emotion classification dataset was used from the research paper of Mohammed et al., but new dataset can be created to explore the same problem.

References

1. Xiao Zhang, Wenzhong Li1, Sanglu Lu, “Emotion detection in online social network based on multi-label learning," Database Systems for Advanced Applications- 22nd International Conference, 2017, pp. 659-674
2. Yadollahi, Ali and Shahraki, Ameneh Gholipour and Zaiane, Osmar R, “Current State of Text Sentiment Analysis from Opinion to Emotion Mining," ACM. Survey, may,2017, pp.1-25
3. Rangel and Paolo Rosso, “On the impact of emotions on author profiling”, Information Processing & Management 52, 1 (2016), pp.73–92
4. Carlos Busso, Zhigang Deng, Serdar Yildirim, Murtaza Bulut, Chul Min Lee, Abe Kazemzadeh, Sungbok Lee, Ulrich Neumann, and Shrikanth Narayanan, “Analysis of emotion recognition using facial expressions, speech, and multimodal information”, In Proceedings of the 6th International Conference on Multimodal Interfaces. ACM, 2004, pp. 205–211
5. Alicja Wieczorkowska, Piotr Synak, and Zbigniew W. Ra´s., “Multi-label classification of emotions in music”, In Intelligent Information Processing and Web Mining. Springer, 2006, pp. 307–315
6. Jabreel M., Moreno A, “A Deep Learning-Based Approach for Multi-Label Emotion Classification in Tweets”, Appl. Sci. 2019; 9:1123. doi: 10.3390/app9061123
7. Avetisyan, H and Bruna, Ondej and Holub, Jan, “Overview of existing algorithms for emotion classification Uncertainties in evaluations of accuracies, "Journal of Physics: Conference Series, 2016, vol:772
8. Lin, K.H.Y., Yang, C., Chen, H.H., “Emotion classification of online news articles from the reader’s perspective”, In: Web Intelligence, 2008, pp. 220–226

9. Xiang Feng, Yaojia Wei, Xianglin Pan, Longhui Qiu and Yongmei Ma, “Academic Emotion Classification and Recognition Method for Large-scale Online Learning Environment—Based on A-CNN and LSTM-ATT Deep Learning Pipeline Method”, Int J Environ Res Public Health, 2020, doi:10.3390/ijerph17061941
10. Douiji yasminaa, Mousannif Hajarb, Al Moatassime Hassana, “Using YouTube comments for text-based emotion recognition”, ANT/SEIT 2016: pp.292-299
11. Chew-Yean, “Emotion Detection and Recognition from Text Using Deep Learning”, CSE Developer Blog, https://devblogs.microsoft.com/cse/2015/11/29/emotion-detection-and-recognition-from-text-using-deep-learning, 2015
12. Kalyani Vishwakarma and Prof. Pushpak Bhattacharya, “Literature Survey: Multi-label Emotion Detection from Text”, Proceedings of the 32nd International BCS Human Computer Interaction Conference, 2018
13. Z. Kozareva, B. Navarro, S. V´azquez, and A. Montoyo, “Ua-zbsa: a headline emotion classification through web information,” In: 4th International Workshop on Semantic Evaluations, 2007, pp. 334–337.
14. Neil Vaughan, Maurice Mulvenna, and Raymond Bond. 2018, “Colour coded emotion classification in mental health social media”, In Proceedings of the 32nd International BCS Human Computer Interaction Conference (HCI '18). BCS Learning & Development Ltd., Swindon, GBR, Article 171, pp.1–5. DOI:https://doi.org/10.14236/ewic/HCI2018.172
15. Mei Silviana Saputri, Rahmad Mahendra, Mirna Adriani “Emotion Classification on Indonesian Twitter Dataset” IALP 2018, 90-95
16. HyunJu Lee, DongIl Shin, and DongKyoo Shin “A Study on the Emotion Classification as well as the Algorithm of the Classification Applying EEG-Data” The 2014

International Symposium on Real-time Natural User Interface and Natural User experience (RN2 2014), pp. 515-521.

17. W. Wang, L. Chen, K. Thirunarayan, and A. P. Sheth, “Harnessing twitter” big data” for automatic emotion identification”, in 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, IEEE, 2012, pp. 587–592
18. W. A. Hussien, Y. M. Tashtoush, M. Al-Ayyoub, and M. N. Al-Kabi, “Are emoticons good enough to train emotion classifiers of Arabic tweets?”, in 2016 7th International Conference on Computer Science and Information Technology (CSIT), IEEE, 2016, pp. 1–6
19. T. Danisman and A. Alpkocak, “Feeler: Emotion classification of text using vector space model,” in AISB 2008 Convention Communication, Interaction and Social Intelligence, 2008, vol. 1, p. 53
20. A. F. El Gohary, T. I. Sultan, M. A. Hana, and M. El Dosoky, “A computational approach for analyzing and detecting emotions in Arabic text,” International Journal of Engineering Research and Applications (IJERA), 2013, vol. 3, pp. 100–107
21. R.Burget and Jan Karasek, “Recognition of Emotion in Czech Newspaper Headlines” Radioengineering 20.1 (2011), 39-47.
22. Konstantinos Trohidis, Grigorios Tsoumakas, George Kalliris and Ioannis Vlahavas, “Multi-label classification of music by emotion” EURASIP Journal on Audio, Speech, and Music Processing, 2011,1–9
23. SemEval-2018 Task 1: Affect in Tweets (Emotion Classification Dataset): https://competitions.codalab.org/competitions/17751#learn_the_details-datasets

24. Mohammed, S., M.; Bravo-Marquez, F.; Salameh, M.; Kiritchenko,S, “Semeval-2018 task 1: Affect in Tweets”, In Proceedings of the 12th InternationalWorkshop on Semantic Evaluation, New Orleans, LA, USA, 5–6 June 2018; pp. 1–17
25. Mohammad, S.; Kiritchenko, S. Understanding emotions, “A dataset of tweets to study interactions between affect categories”, In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, Miyazaki, Japan, 7–12 May 2018
26. Joseph Anthony L. Reyes, Ryca Laeane C. Matro & Miguel Alfonso C. Oliva, “Twitter Usage in The Philippines During Typhoon Nockten”, New Zealand Journal of Asian Studies 20, 1 (June 2018): 81-103
27. Manmohan singh, “Stop the stopwords using different python libraries”, 2020, https://medium.com/towards-artificial-intelligence/stop-the-stopwords-using-different-python-libraries-ffa6df941653
28. K. R. Srinath. “Python – The Fastest Growing Programming Language”, International Research Journal of Engineering and Technology (IRJET) Dec 2017; pp. 354-357
29. Python for datascience: https://data-flair.training/blogs/python-for-data-science
30. Jason Brownlee, “Introduction to the python Deep learning library TensorFlow”, 2016, https://machinelearningmastery.com/introduction-python-deep-learning-library-tensorflow
31. R.Zicari. “Keras:The python Deep learning library”, 2018, http://www.odbms.org/2018/06/keras-the-python-deep-learning-library
32. García-Gonzalo, E., Fernández-Muñiz, Z., García Nieto, P.J., Bernardo Sánchez, A., Menéndez Fernández, M, “Hard-Rock Stability Analysis for Span Design in Entry-Type Excavations with Learning Classifiers”, Materials 2016, 9, 531, DOI: https://doi.org/10.3390/ma9070531

33. Galdi P and Tagliaferri, Roberto, “Data Mining: Accuracy and Error Measures for Classification and Prediction”, In Encyclopedia of Bioinformatics and Computational Biology, Elsevie, 2019, pp.431-436 DOI: https://doi.org/10.1016/B978-0-12-809633-8.20474-3
34. R. Gajjar and T. Zaveri, “Defocus blur radius classification using random forest classifier”, 2017 International Conference on Innovations in Electronics, Signal Processing and Communication (IESC), 2017, pp.219-223, DOI: https://doi.org/10.1109/IESPC.2017.8071896
35. “An intuitive understanding of Word Embedding: From Count vectors to word2vec”, 2017, https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec
36. “A Beginner’s guide to word2vec and neural word embeddings”, https://wiki.pathmind.com/word2vec
37. S.Konstadinov, “Understanding GRU networks”, 2017, https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be
38. S. Rajaraman, Sameer K. Antani, “Modality-specific deep learning model ensembles toward improving TB detection in chest radiographs”,IEEE access:practical innovations, open solutions vol.8 (2020):27318-27326, DOI: 10.1109/access.2020.2971257