International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Effectiveness of Word Embedding Models in Generating Sub-Emotions with Affinity Propagation Algorithm: A Comparative Analysis

Srinath K. S.[1*], Kiran K.[2], P. Deepa Shenoy[3], Venugopal K. R.[4]

Submitted: 19/01/2024 **Revised**: 28/02/2024 **Accepted**: 05/03/2024

Abstract: Over the past few decades, there has been a rapid increase in the prevalence of Mental Illness (MI) conditions worldwide. Our infrastructure is woefully inadequate to mitigate this problem. In MI, Depression is one of the major causes of suicidal ideation. According to a World Health Organization (WHO) report of 2022, there was a twenty-five per cent increase in MI. It is also estimated that for one lakh people, there should be three MI specialists but approximately there are only 0.75% specialists available and it varies for different countries.

Hence, to alleviate this problem, modern technological conventions like NLP and machine learning should be utilised. In this research, we have implemented and compared Word2Vec and GloVe models such as a) google-news-300, b) GloVe model twitter-25 & twitter-200 and c) fastText, by using them to convert word-to-vector for generating sub-emotions. The developed novel models are also compared to find the number of vocabularies, sub-emotions clusters, Mean **(μW)** average word per cluster, standard deviation **(σW)** per cluster and embedded users sample text. The generated sub-emotions can be further used with Machine Learning (ML) and Deep Learning (DL) algorithms to detect Mental illness in social media posts.

Keywords: *Affinity Propagation (AP), fastText, GloVe model, Machine Learning, Mental illness (MI), Social Media (SM), Word2Vec.*

1. Introduction

Mental illness refers to a wide range of mental health conditions that affect a person's mood, thinking as well as behaviour. According to WHO, 1 out of 8 people around the world were suffering from various mental disorders in 2019. In 2020, the number of people suffering from mental illness rose significantly. The COVID-19 pandemic is one of the reasons for this cause [1]. If mental illness is not diagnosed at the beginning stage, it can cause severe health problems and even lead to death. There are more than 200 classified forms of mental illness found so far [2]. In which, some of the life-threatening disorders are: clinical depression, bipolar disorder, dementia, schizophrenia and anxiety disorders similar to heart disease and diabetes. Mental disorder treatments are scarcely available throughout the world [3]. Depression is one of the major causes of suicide ideation. Emotions are important decisive elements of our daily lives that decide to do work based on the feelings we exhibit [4]. There are eight primary emotions namely Anger, Anticipation, Disgust, Fear, Joy, Sadness, Surprise and Trust and two sentiments namely Positive and Negative as shown in Figure 1 recommended in [5]. These emotions are experienced and expressed universally by users in SM and some unique patterns of emotions are expressed by the users who suffer from MI that are different from normal users in SM. These patterns of expressing emotions by mentally ill people can be analysed and using that, their mental states can be determined. There are some sets of words that belong to one or more of the above emotions at different levels that are termed sub-emotions. These sub-emotions can be used to analyse the mental health of SM users by understanding the levels of emotions expressed by them.

[1]B.E. In Information science and Engineering And M. Tech In Networks and Internet Engineering, Visvesvaraya Technological University, India*
[1]Research Scholar in department of Computer Science and Engineering in University Visvesvaraya College of Engineering, Bangalore University, Bengaluru, India, Email: solansrinath@gmail.com.*
[2]Associate professor, Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering, Bangalore University, kirank@uvce.ac.in.
[3]Professor, Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering, Bangalore, email: shenoypd1@gmail.com.
[4]Former Vice Chancellor, Ph.D, Economics, Bangalore University & Ph.D. in Computer Science from Indian Institute of Technology, Madras. Email: venugopalkr@gmail.com
**Corresponding Author: Srinath K S*
Email: solansrinath@gmail.com.

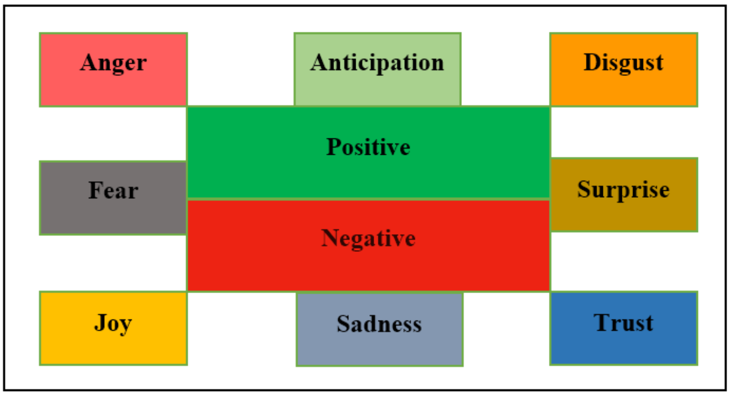


Fig 1: Eight Primary Emotions and Two Sentiments.

a) Motivation

In the current era, a large population depends on social media to update their day-to-day activities. Nine in ten internet users use social media. Approximately fifty-nine percent of the world's population depends upon social media to express their feelings and thoughts [6]. This information motivated us to work on social media data to propose sophisticated algorithms for detecting the mental state of social media users.

b) Contribution

In this research, we have implemented and compared Word2Vec and GloVe models such as a) google-news-300, b) GloVe model twitter-25 & twitter-200 and c) fastText, by using them to convert word-to-vector for generating sub-emotions. We also compared the developed novel models to find the number of vocabularies, sub-emotions clusters, mean **(μW)** average word per cluster, standard deviation **(σW)** per cluster and embedded users sample text.

c) Organization

This article contains following sub-sections: Literature Survey conducted in section 2, which examines earlier studies that explored the utilization of social media to identify mental health issues. Generating Bag of Sub-Emotions is discussed in section 3. It explains the process of "generating sub-emotions" and the tools used, including the emotion lexicon, Word2Vec, and the Affinity Propagation algorithm. The Experimental Results in section 4 explain the detailed information of the outcomes achieved through a proposed model, followed by conclusions and future work in section 5, which summarises the main findings and proposes future research directions. Finally, the References section lists the sources cited throughout the content.

2. Literature Review

This section focuses on the approaches and techniques employed in previous research, building upon the earlier exploration conducted to detect mental illness in textual content generated by social media users. The existing literature is thoroughly examined, emphasizing the strengths and limitations of each study.

K. A. Govindasamy and N. Palanichamy, in their research work, aimed to detect depressed users in social media through their posts. They have used two classifiers such as naïve Bayes and NBTree hybrid model. By using Tweepy python library the twitter data of SM users are collected in two sets of 1000 and 3000 to train and test the model. Both classification algorithms performance is analysed and found NBTree and Naive Bayas gave 97.31% accuracy for 3000 samples and 92.34% for 1000 samples. Interestingly they concluded that both classifiers gave the same accuracy. This technique uses only three sentiments like positive, negative and neutral for classification of depressed users, but it lags using emotions like anger, joy, happy, anticipation, fear, disgust for finer prediction [7].

S. Tariq et al. proposed a method to categorise the depressed users having various mental disorders such as Anxiety, Depression, Bipolar, and Attention Deficit Hyperactivity Disorder (ADHD) on the basis of data obtained from Reddit which is a popular network community platform. In their method, they have used powerful widely used classifiers namely Random Forrest (RF), Support Vector Machine (SVM), and Naïve Bayes (NB) and Co-training technique which is one of the semi-supervised learning approaches. They have also used Reddit API in order to download the posts and top five associated comments in order to construct a feature space. They have compared the effectiveness of Co-training based classifier with art classifiers. They had found that Co-training based approach is 3% more effective than each and every art classifier. Their method does not detect classification problems by extracting data from SM [8].

In [9] authors proposed an approach where the text is used to identify the symptoms of depression. This model uses Long Short-Term Memory (LSTM)-based Recurrent Neural Network (RNN) to model two different emotional states: depression and non-depression. The dataset consists of youth's own text-based questions in Norway which are verified from medical and psychological experts. Further this model gives meaningful explanations using an Artificial Intelligence (XAI) algorithm called Local Interpretable Model-Agnostic Explanations (LIME). The proposed approach yielded 98% and 99% mean prediction performance on the first and second dataset which consists of around 11,807 and 21,807 texts. This deep learning-based approach needs to be further explored at greater levels with a comprehensive dataset as the dataset used here is prone to more changes.

Reddit post are examined in order to find the factors that disclose their depression attitudes by authors in [10] using Natural Language Processing (NLP) and machine learning methods for training the data. They have used 2 classifiers namely Support Vector Machine (SVM) & Multilayer Perceptron (MLP) and found that they have detected depression with more accuracy when they used MLP classifiers. When using the SVM classifier, they were able to identify depression with an accuracy of 80% and F1 scores of 0.80. On the other hand, with the MLP classifier, they achieved an accuracy of 91% and F1 scores of 0.93 in detecting depression.

An empirical model was proposed in [11] for detecting and diagnosing depression in SM users. The dataset used is extracted from Twitter and Reddit by Olteanu et al. [12]. In the proposed model, Word2Vec language modelling is used to convert words to its appropriate vector values. SVM algorithm is used for classification and achieved 95% accuracy on Twitter and 73% accuracy on Reddit users.

In [13], the researchers conducted a detailed performance evaluation of four transformer-based pre-trained small language models with fewer than 15 million tunable parameters. These models included Electra Small Generator (ESG), Electra Small Discriminator (ESD), XtremeDistil-L6 (XDL), and Albert Base V2 (ABV). They fine-tuned these models using different hyperparameters and tested their performance on a labeled Twitter dataset for classifying depression intensity into three categories: 'severe,' 'moderate,' and 'mild.' Evaluation metrics such as accuracy, F1 score, precision, recall, and specificity were calculated. A comparison was also made with a larger model, DistilBert, which has 67 million tunable parameters. The results showed that ESG outperformed all other models, including DistilBert, achieving best F1 score of 89% with less training time. This study provides an insight for improving depression detection and selecting the most effective language model for Twitter-related NLP tasks, considering performance and training time.

a) Background of previous works

In last few years, many researchers faced towards SM to find the mental illness of SM users which mainly depends on user’s posts consisting of the textual content in which users express their emotions and feelings. English is the major language used by the SM users to express their emotions and feelings. Predicting mental illness based on emotions have gave good results but in [14] authors have come up with new idea to predict mental illness using user’s sub-emotions with the help of fastText Word2Vec converter. This idea motivated us to generate bag of sub emotions using other effective models and compare them to provide valuable insights for improving MI detection.

b) Problem Definition

Over the past few decades, the prevalence of Mental Illness (MI) conditions has risen dramatically, becoming a significant global health concern. The existing infrastructure to address this issue is inadequate, leading to a pressing need for innovative approaches to tackle the problem effectively. Among various MI conditions, Depression stands out as a major cause of suicidal ideation, underscoring the urgency of addressing mental health challenges.

The objective of this research work is to implement and compare Word2Vec models such as a) google-news-300, b) GloVe model twitter-25, twitter-200 and fastText which are used to convert word-to-vector for generating bag of sub-emotions. The generated sub-emotions can be further used with Machine Learning (ML) and Deep Learning (DL) algorithms to detect Mental illness in social media posts.

This research intends to contribute to the development of innovative technological solutions that can complement existing mental health care services and bridge the gap caused by the scarcity of mental health specialists. By providing early detection and support, we aspire to make a positive impact on individuals struggling with mental health challenges worldwide.

3. Methods

The architecture of the proposed model to generating sub-emotions and mapping text to sub emotions sequence is elaborated in this section. Figure 2 illustrates the two-stage process involved. In stage 1, the generation of sub-emotions begins by identifying emotions from the emotion lexicon, converting the words to vectors using the Word2Vec model, clustering the vectors, and ultimately generating the sub-emotions. In stage 2, the mapping of user text to the sub-emotion sequence occurs through the steps of reading the user text, tokenization, and mapping to the corresponding sub-emotions.

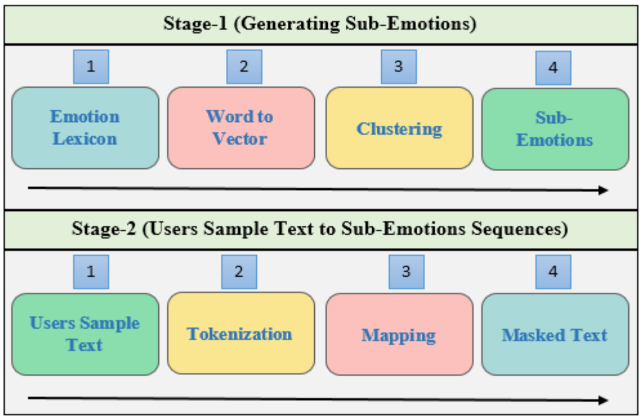


Fig 2: Architecture of Bag of Sub-Emotions Model.

3.1 Generating sub-emotions

The process of generating sub-emotions encompasses four distinct stages: (a) Emotion identification within the lexicon, (b) Word-to-vector conversion using Word2Vec, (c) Vector clustering, and (d) Sub-emotion generation.

The variables and functions required for sub-emotion generation are defined below:

- Lexicon: A set of lexicon emotions
- Word2Vec(word): A function that converts a word to a vector using the google-news-300 Word2Vec model
- Clustering Algorithm(vectors): A function that clusters the given vectors using the Affinity Propagation (AP) algorithm
- Sub-emotions: A set of generated sub-emotions

The mathematical model to generate sub-emotions is represented as follows:

Step 1: Start

Step 2: Read Lexicon emotions

Step 3: For each word in Lexicon: - Compute vector = Word2Vec (word)

Step 4: Cluster sub-emotions: - Call Clustering Algorithm on the set of vectors obtained in Step 3 - Assign the resulting clusters as Sub-emotions

Step 5: Output the generated sub-emotions

Step 6: Stop

3.1.1 Emotion Lexicon

The English language encompasses a vast vocabulary comprising millions of words, with each word being associated with specific emotions commonly experienced by humans. These words are organized and stored in a dictionary known as a lexicon. The lexicon categorizes words based on eight primary emotions commonly observed in human behaviour, namely anger, fear, anticipation, trust, surprise, sadness, joy, and disgust. Additionally, each word in the lexicon is also tagged with sentiments of either positive or negative connotation.

In the present research, an online resource known as the Emotion Lexicon corpus is used. This corpus was developed by Mohammad Saif M. and Turney Peter D. as part of their research publications [15] [16]. The Emotion Lexicon corpus can be accessed through the following URL:

https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

3.1.2 Word2Vec Conversion

Word2Vec, an open-source model, is used to convert words into vector representations. To cluster the words according to the emotion, they must be converted to its vector values so that a centroid is easily identified. There are some popular pre-trained Word2Vec models available such as:

1. Google's pre-trained Word2Vec model - Trained on a large Google News dataset of over billion words.
2. Stanford's GloVe model - Similar to Word2Vec but uses a co-occurrence matrix instead of a neural network to learn word embeddings.
3. Fast Text - A model developed by Facebook that extends Word2Vec by considering subword information also in addition to words.

The dictionary of a Word2Vec model consists of the vocabulary of words that the model has been trained on, along with their corresponding vector representations. The size of the dictionary depends on the size of the corpus used to train the model.

a) Google-news-300

The Google-News-300 model is a pre-trained word embedding model which converts words into vectors of 300-dimensions. It is developed by google researchers and this model is trained on a large corpus of news articles [17]. The Google's pre-trained Word2Vec models is accessed through 'gensim' library in Python, which provides an interface to load and use this model.

b) Twitter 25 & Twitter 200 GloVe model

The Twitter 25 and Twitter 200 GloVe (Global Vectors for Word Representation) models provide access to pre-trained word embedding models. These models have been trained on large corpora of Twitter data, with specified dimensions of 25 or 200 [18].

The GloVe model assigns a vector of a specified dimension to each word in the vocabulary, based on co-occurrence statistics with other words in the corpus. To convert a word into its corresponding vector using the Twitter 25 and Twitter 200 GloVe models, the pre-trained vector representation of the word is mapped within the model's vocabulary. In this research work, the pre-trained GloVe model is utilized by integrating the 'gensim' library.

d) fastText

fastText is a lightweight library developed by Facebook Research Team. It represents each word as a vector that captures the features of the word based on its surrounding words [19]. The fastText involves two main stages: training and inference for the process of converting words to vectors.

Training: Initially corpus of large text data is used to train the fastText model. The fastText is trained to represent each word in the vocabulary as a vector in a high-dimensional space of 300. The vectors are learned in a way that maximises the likelihood of predicting the context (surrounding words) of each word.

Inference: After training, the model is used to convert new words to their respective vector representations. This process involves breaking down words into character n-grams.

3.1.3 Clustering

In clustering, Vector form of lexicons are clustered based on their emotional meanings. To achieve this, Affinity Propagation algorithm is applied to these vectors, which groups similar words into distinct clusters [20] [21].

Affinity Propagation is an unsupervised machine learning algorithm used for clustering. The algorithm is based on the concept of message passing between data points. The mathematical model for Affinity Propagation is defined as follows:

Let X = {x1, x2, ..., xn} be a set of n data points.

Let S = {s_{ij}} be a similarity matrix, where s_{ij} represents the similarity between data points x_i and xj.

Let R = {r_{ij}} be a matrix that represents the responsibility of point i to point j.

Let A = {a_{ij}} be a matrix that represents the availability of point i to point j.

Initialization:

Set $r_{ij} = 0$ and $a_{ij} = 0$ for all i and j.

Update rules:

1. Responsibility update: r_{ij} <- s_{ij} - max{k≠i} { a_{ki} + s_{ik} } where k ranges over all data points except for i.
2. Availability update: a_{ii} <- sum{k≠i} {max(0, r_{ij})} a_{ij} <- min(0, r_{ij} + sum{k≠i, j≠k} {max(0, r_{ij})}) for i ≠ j where k ranges over all data points except for i and j.
3. Clustering: Let R+A = {r_{ij} + a_{ij}} be the matrix of summed values. Let C be a set of cluster centers. Assign each data point to the cluster center with the highest summed value in R+A.
4. Termination: Repeat steps 1 to 3 until convergence or maximum iterations are reached.

3.1.4 Sub-Emotions Lexicon

Once the vectorization and clustering process is finished, we obtain automatically generated clusters representing sub-emotions. In the subsequent step, the vectorized words within each cluster are replaced with their original counterparts. This results in clusters containing words associated with similar levels of emotion. For instance, when analyzing the clusters for anger0 and joy0 (the first cluster obtained), we observe that the words within each cluster pertain to either negative or positive emotions, respectively. This indicates that our clustering algorithm has effectively grouped the words based on their emotional connotations.

3.2 Mapping User Text to Sub-Emotions

In the second stage, the user's sentence is used as input, and each word in the sentence is tokenized using the "word_ tokenize" package in Python. These tokens are subsequently mapped to their respective bag of sub-emotions. This procedure, often known as text masking or text embedding, involves embedding each word within the context of its related sub-emotions.

To illustrate the process of mapping sample or user text to sub-emotions, the following algorithm outlines the steps involved. By executing this mapping process, it provides more nuanced understanding of the emotional content of the user's text.

Algorithm: Generating Sub-Emotions form Sample or User Text

Step 1: Start

Step 2: Read user input sentence

Step 3: Create user_token[] array

(a) Convert the words in the sentence into tokens.

(b) Store the tokens in the user_token[] array

Step 4: Output user_token[]

Step 5: Create final_array[]

(a) Map each user token with its corresponding sub-emotions

(b) Append the mapping to the final_array[]

Step 6: Output final_array[]

Step 7: Stop

4. Results

The aim of this research work is to implement sub-emotions generating model and compare Word2Vec models such as a) google-news-300, b) GloVe model twitter-25 & twitter-200 and c) fastText used to convert word-to-vector for generating sub-emotions. The number of vocabularies, sub-emotions clusters, Mean word per cluster **(μW)** and standard deviation **(σW)** per cluster [22] [23] obtained are discussed in the following sub sections.

4.1 Emotions, Vocabulary and Clusters

After conducting the experiment, the vocabulary and clusters obtained for eight emotions and two sentiments are tabulated in table Table-1, Table-2, Table-3 and Table-4. These tables contain emotions, vocabulary, clusters, Mean words (μW) and standard deviation (σW) for google-news-300, twitter-25, twitter-200 and fastText model.

Table 1: Emotions, Vocabulary and Clusters for google-news-300

Emotions	**Vocabulary**	**Clusters**	**(μW)**	**(σW)**
Anger	5499	267	20.52	39.57
Anticipation	5306	260	20.33	44.69
Disgust	4706	216	21.69	40.70
Fear	6429	282	22.72	47.76
Joy	3975	192	20.60	42.91
Sadness	5306	260	20.33	44.69
Surprise	3448	174	19.71	40.70
Trust	5000	244	20.41	45.32
Positive	9879	436	22.61	47.17
Negative	11009	493	22.29	40.79

Table 2: Emotions, Vocabulary and Clusters for twitter-25

Emotions	**Vocabulary**	**Clusters**	**(μW)**	**(σW)**
Anger	4825	243	19.78	11.27
Anticipation	4620	229	20.09	12.04
Disgust	4007	201	19.84	12.47

Fear	5578	258	21.54	11.68
Joy	3609	189	19.00	10.01
Sadness	4620	229	20.09	12.04
Surprise	3146	172	18.19	10.56
Trust	4525	216	20.86	11.41
Positive	8673	368	23.51	12.78
Negative	9232	420	21.93	13.18

Table 3: Emotions, Vocabulary and Clusters for twitter-200

Emotions	**Vocabulary**	**Clusters**	**(μW)**	**(σW)**
Anger	4825	269	17.88	22.56
Anticipation	4620	250	18.41	24.45
Disgust	4007	228	17.50	23.93
Fear	5578	187	19.37	26.08
Joy	3609	220	16.34	18.82
Sadness	4620	250	18.41	24.45
Surprise	3146	180	17.39	21.89
Trust	4525	260	17.34	21.42
Positive	8673	428	20.22	37.16
Negative	9232	455	20.25	29.45

Table 4: Emotions, Vocabulary and Clusters for fastText

Emotions	**Vocabulary**	**Clusters**	**(μW)**	**(σW)**
Anger	5638	681	8.27	18.57
Anticipation	5446	625	8.71	18.81
Disgust	4849	586	8.26	18.64
Fear	6629	758	8.73	19.20
Joy	4082	472	8.63	20.49
Sadness	5444	624	8.71	18.94
Surprise	3517	412	8.52	19.58
Trust	5152	615	8.36	20.37
Positive	10281	1170	8.78	19.35
Negative	11412	1247	9.14	19.37

4.2 Sub-Emotion Lexicon

Sub-emotion lexicons are grouped into clusters using the affinity propagation algorithm. Table 5, Table 6, Table 7, and Table 8 depict the sample groups of words within the clusters, corresponding to specific emotions such as anger, fear, joy, and sadness. These words were randomly selected from the results of the fastText model. From our results it is noted that, grouping of words varies for other models as shown in Table 9.

Table 5 Anger Clusters with Sub-Emotion

Anger0	**Anger1**	**………**	**Anger681**
abandoned banished depreciated destroyed	agony anguish pain angst		wreck barque shipwreck yap

Table 6 Fear Clusters with Sub-Emotion

Fear0	**Fear1**	**……..**	**Fear758**
accident accidental crash fatality	accused accusing mishap derailment		exorcism witch voodoo yap

Table 7 Joy Clusters with Sub-Emotion

Joy0	**Joy1**	**……**	**Joy472**
abundance abundant bountiful plenty	achieve fulfill gain attain		snow snowy excel nip

Table 8 Sadness Clusters with Sub-Emotion

Sadness0	**Sadness1**	**……..**	**Sadness624**
accident crash fatality mishap	ache aching achy adversity		bruise callus gash tort

Table 9 Comparison of Anger0 Emotion words for fastText, google-news-300, twitter-25 and twitter-200

Anger0			
fastText	**google-news-300**	**twitter-25**	**twitter-200**
abandoned banished depreciated destroyed disused	alienate odious bother chafe bruising	abandonment alienation incense murderous ordeal	advocacy judiciousness contemptuously dishonesty spending

4.3 Sample Sentence to Sub-Emotions sequences

Once the bag of sub-emotions is generated in the first stage of our proposed model, it is utilized to mask the sample sentence or user text. In this stage, the sample sentence is read as an array and tokenized into individual words. Each token is then matched with its corresponding sub-emotion from the bag of sub-emotions. During this process, the same word may occur in different clusters and at different levels. To determine the specific sub-emotion for a word, we compare the cosine similarity of the word (*w*) with all the sub-emotions (*s*), and select the closest sub-emotion with the maximum similarity. The algorithm for finding the closest sub-emotion for a word [24] is as follows:

Algorithm: finding the closest sub emotion for the word

Step 1: The target word vector is denoted as vector *T*.

Step 2: Iterate through the group of word vectors in the clusters: For each word vector in the group, denoted as vector *G*, perform the following steps:

a. Compute the dot product: Calculate the dot product between vector *T* and vector *G* by summing the element-wise products of corresponding elements in the vectors. Denoted this value as *dot Product TG*.

Dot ProductTG = sum(*T[i] * G[i]*) for i = 0 to n-1

b. Compute the magnitudes: Calculate the magnitudes (or Euclidean norms) of vector *T* and vector *G*. The magnitude of vector *X* is the square root of the sum of the squares of its elements.

magnitudeT = sqrt(sum(*T[i]^2*)) for i = 0 to n-1

magnitudeG = sqrt(sum(*G[i]^2*)) for i = 0 to n-1

c. Calculate the cosine similarity: Divide the dot product by the product of the magnitudes to obtain the cosine similarity between vector *T* and vector *G*.

*similarity = dotProductTG / (magnitudeT * magnitudeG)*

d. Store the similarity value: Keep track of the similarity value for each vector *G* in the group.

Step 3: Rank the vectors by similarity: Sort the vectors in the group based on their similarity values in descending order. This allows us to identify the closest word vectors to the target word vector.

Step 4: Select the closest word vectors: Choose the top N word vectors with the highest similarity values, where N is the desired number of closest word vectors.

Table 10 presents the masked sequence generated by the google-news-300, twitter-25, twitter-200, and fastText models for the sample or user text. Additionally, the sub-emotion sequence obtained for fastText in [14] is included in the table for comparison with our models. Interestingly, it is observed that even stop words like "the," "is," and "to" are associated with sub-emotions. As far as our knowledge goes, the fastText library has recently undergone updates and improvements.

Table 10: Sub Emotion Sequence for Sample Sentence

Sample sentence	The most important thing is to try and inspire people
google-news300	The positive231 fear145 anger193 is to positive366 and joy186 positive343
twitter25	The positive231 fear145 anger193 is to positive366 and joy186 positive343
twitter200	The positive353 negative319 positive295 is to positive132 and trust87 negative453
fastText	The positive701 negative856 negative891 is to negative1145 and surprise166 negative491
fastText (generated by [14])	anticipation27 joy27 positive5 negative62 anticipation10 anticipation29 positive20 negative80 trust23 joy16

4.4 Comparative Analysis of Word2Vec and GloVe Models

In this section, the obtained a) Vocabulary b) Clusters c) Mean **(μW) -** average word per cluster and d) Standard deviation **(σW)** for google-news-300, twitter-25, twitter-200, fastText models and fastText model by [14] are compared with each other to conclude the optimal model that can be used to detect mental illness in SM post.

a) Vocabulary

Table 11 provides a tabulation of the number of vocabularies obtained for each emotion, and Figure 3 depicts the corresponding results. It is observed from the tabulated values that fastText demonstrates a higher vocabulary count compared to the other models across all emotions. The second highest vocabulary count is observed in the google-news-300 model, followed by the GloVe models Twitter-200 and Twitter-25. Interestingly, the GloVe model exhibits the same number of vocabularies regardless of its dimensions. These findings hold significance for our ongoing research on detecting mental illness in social media posts.

Table 11: Vocabularies Associated with Emotions

Vocabulary					
Emotions	**google-news-300**	**twitter-25**	**twitter-200**	**fastText**	**fastText in [14]**
Anger	5499	4825	4825	5638	6035
Anticipation	5306	4620	4620	5446	5837
Disgust	4706	4007	4007	4849	5285
Fear	6429	5578	5578	6629	7178
Joy	3975	3609	3609	4082	4357
Sadness	5306	4620	4620	5444	5837
Surprise	3448	3146	3146	3517	3711
Trust	5000	4525	4525	5152	5481
Positive	9879	8673	8673	10281	11021
Negative	11009	9232	9232	11412	12508

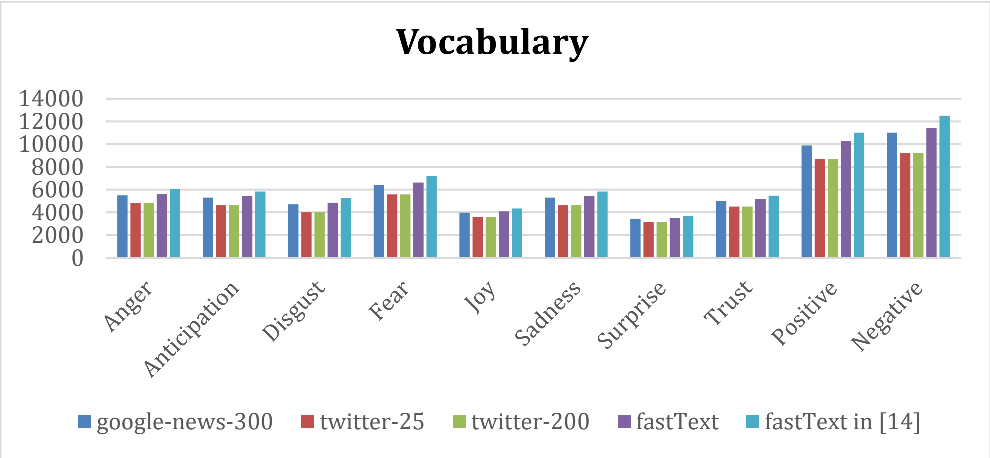


Fig 3: Vocabularies Associated with Emotions

b) Clusters

Table 12 presents the number of clusters obtained by the AP algorithm for each emotion, while Figure 4 provides a visual representation of these results. The tabulated values indicate that our fastText model generated a higher number of clusters compared to the other models.

The k-means algorithm [25] can be used to cluster the vocabulary into a specific number (*k*) of distinct clusters. However, a drawback of this algorithm [26] is that the user needs to determine the precise number of clusters required for each emotion. This aspect can lead to significant variations in the words associated with each cluster across all emotions.

To address this issue, the AP algorithm is employed, as shown in Table 13 and Table 14. By using this algorithm, it can be observed that the mean and standard deviation values for all emotions exhibit closer proximity.

Table 12: Clusters Generated by Affinity Propagation Algorithm

Clusters					
Emotions	**google-news-300**	**twitter-25**	**twitter-200**	**fastText**	**fastText in [14]**
Anger	267	243	269	681	444
Anticipation	260	229	250	625	393
Disgust	216	201	228	586	367

Fear	282	258	187	758	488
Joy	192	189	220	472	318
Sadness	260	229	250	624	395
Surprise	174	172	180	412	274
Trust	244	216	260	615	386
Positive	436	368	428	1170	740
Negative	493	420	455	1247	818

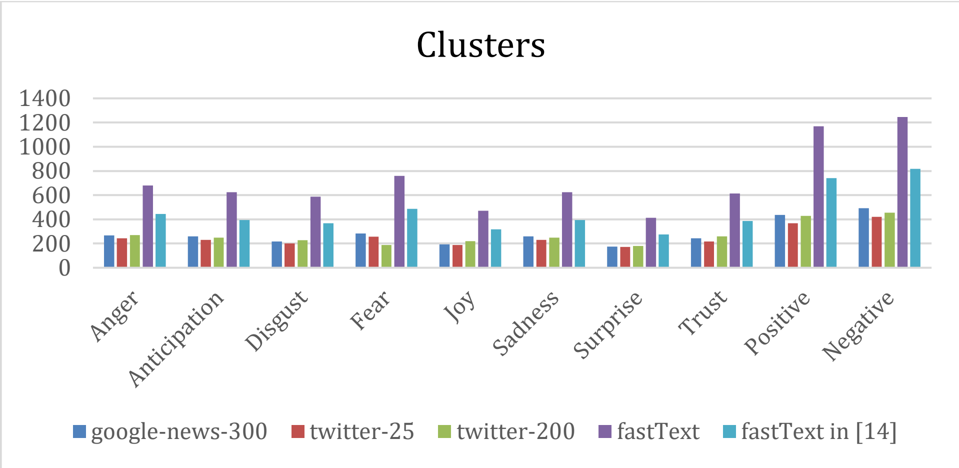


Fig 4: Clusters Generated by Affinity Propagation Algorithm

c) Mean **(μW) -** Average Word per Cluster

The average number of words in each cluster refers to the average count of words contained within individual clusters obtained from a clustering algorithm. This metric provides insights into the size or density of clusters in terms of the number of words they encompass. Table 13 and figure 5 show the closer proximity of mean values obtained for all emotions by both Word2Vec and GloVe models.

To calculate the average number of words per cluster following steps are performed:

Step 1: Apply AP algorithm to vocabulary with respect to its emotions.

Step 2: Assign each word to its corresponding cluster based on the clustering results.

Step 3: Calculate the number of words in each cluster.

Step 4: Compute the average by dividing the total number of words across all clusters by the number of clusters.

Table 13: Mean **(μW) -** Average Word per Cluster

Mean (μW)					
Emotions	**google-news-300**	**twitter-25**	**twitter-200**	**fastText**	**fastText in [14]**
Anger	20.52	19.78	17.88	8.27	13.6
Anticipation	20.33	20.09	18.41	8.71	14.77
Disgust	21.69	19.84	17.5	8.26	14.4
Fear	22.72	21.54	19.37	8.73	14.7
Joy	20.6	19	16.34	8.63	13.7
Sadness	20.33	20.09	18.41	8.71	14.78
Surprise	19.71	18.19	17.39	8.52	13.54
Trust	20.41	20.86	17.34	8.36	14.31
Positive	22.61	23.51	20.22	8.78	14.89
Negative	22.29	21.93	20.25	9.14	15.29

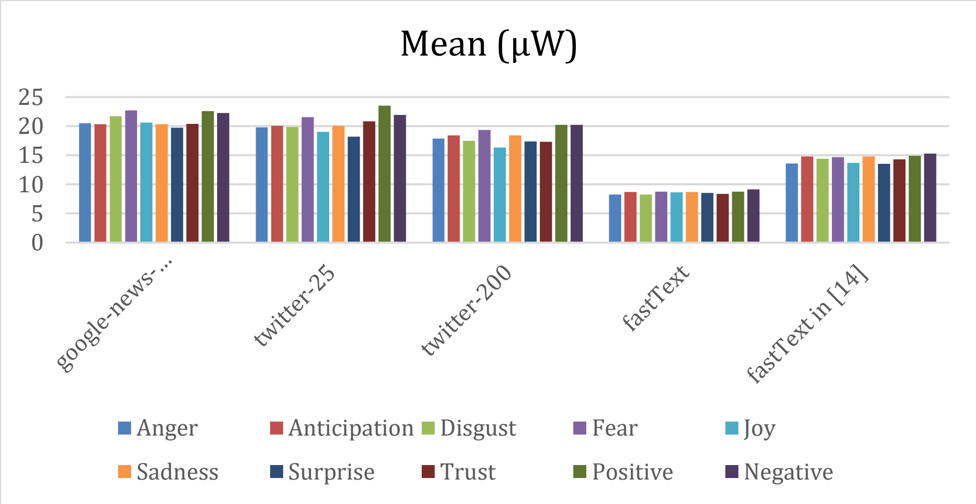


Fig 5: Mean **(µW) -** Average Word per Cluster

d) Standard Deviation (σW)

The standard deviation serves as a metric to quantify the extent of deviation of words within a cluster from its centroid. It aids in assessing the quality and compactness of clusters by evaluating the dispersion of words based on their vector values. The calculated values of standard deviation are tabulated in Table 14, and Figure 6 illustrates the comparison of the Word2Vec and GloVe models in terms of standard deviation.

Table 14: Standard Deviation (σW)

Standard Deviation (σW)					
Emotions	**google-news-300**	**twitter-25**	**twitter-200**	**fastText**	**fastText in [14]**
Anger	39.57	11.27	22.56	18.57	16.53
Anticipation	44.69	12.04	24.45	18.81	20.53
Disgust	40.7	12.47	23.93	18.64	21.29
Fear	47.76	11.68	26.08	19.2	23.36
Joy	42.91	10.01	18.82	20.49	21.25
Sadness	44.69	12.04	24.45	18.94	20.48
Surprise	40.7	10.56	21.89	19.58	28.68
Trust	45.32	11.41	21.42	20.37	21.59
Positive	47.17	12.78	37.16	19.35	24.53
Negative	40.79	13.18	29.45	19.37	23.75

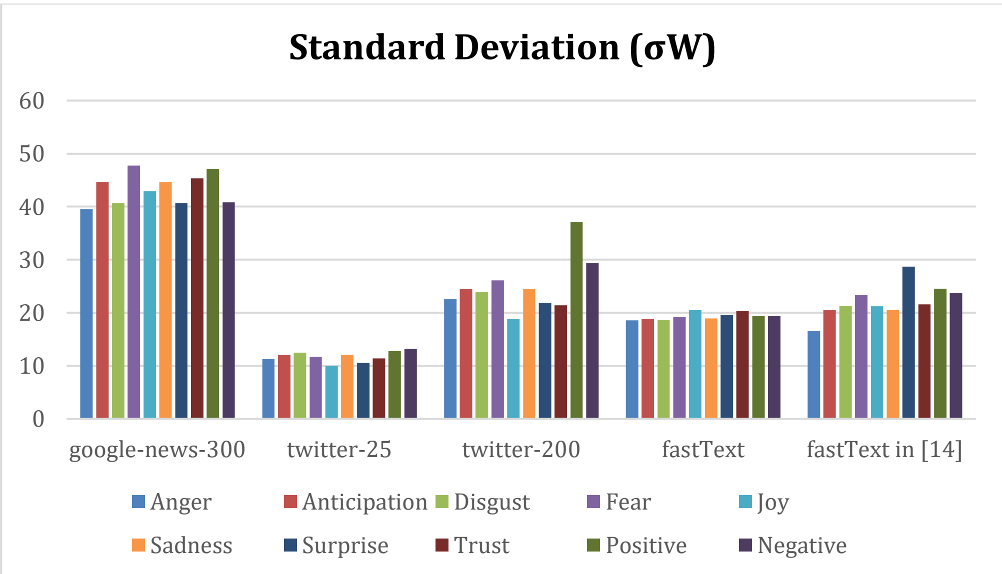


Fig 6: Standard Deviation **(σW)**

5. Discussion

This research work addresses a pressing issue in today's society: the rising prevalence of MI conditions, particularly depression, and the inadequate infrastructure to address them. With MI cases on the rise globally, there's a growing need for innovative solutions to detect and address these conditions.

The primary goal of this study is to explore the effectiveness of word embedding models, specifically Word2Vec and GloVe, in generating sub-emotions using the Affinity Propagation (AP) algorithm. The research also aimed to compare multiple pre-trained models, including google-news-300, GloVe models twitter-25 and twitter-200, and fastText, to assess their performance in converting words to vectors and generating sub-emotions.

The key findings of this study are as follows:

1. **Vocabulary and Clusters**: The analysis revealed that fastText generated the largest vocabulary among the models, while google-news-300 exhibited superior cluster quality and compactness. This discrepancy suggests that the choice of word embedding model has a significant impact on the resulting sub-emotions, with fastText capturing a broader range of words but potentially at the expense of cluster quality.
2. **Mean (μW) - Average Word per Cluster**: When examining the average number of words per cluster, it was observed that fastText tended to produce clusters with fewer words on average compared to the other models. This may indicate a higher granularity in sub-emotion differentiation by fastText, potentially leading to more fine-grained emotional analysis.
3. **Standard Deviation (σW)**: The standard deviation analysis demonstrated that different models exhibited varying levels of dispersion within their clusters. Notably, fastText displayed lower standard deviations in comparison to other models, indicating tighter clusters and potentially more consistent sub-emotion assignments.

5.1 Limitations

The methodology used in this research work is a crucial contribution, offering a structured approach to extracting nuanced emotional content from text data. The limitations identified in this work emphasize the complexity of sub-emotion generation and the challenges inherent in evaluating the performance of word embedding models and clustering algorithms. Some potential limitations of this work are a) Limited focus on Sub-Emotions: The paper focuses primarily on the generation of sub-emotions using Word2Vec and GloVe models. However, it does not provide a comprehensive discussion of the theoretical background or the psychological basis for sub-emotions. This lack of context could limit the broader understanding of the research. b) Emotion Lexicon: Emotion lexicons can vary in terms of accuracy and cultural relevance, which could affect the results.

6. Conclusions and Future Work

In conclusion, this research draws inspiration from [27] and [14] and focuses on implementing a novel model for generating a bag of sub-emotions from an emotion lexicon. Word2Vec and GloVe models, including google-news-300, Twitter 25, Twitter 200, and FastText, are utilized to convert words into vectors. For optimal clustering of emotions, the Affinity Propagation algorithm is employed instead of the k-means clustering algorithm. Comparative analysis is conducted among the four models themselves and with the fastText model [27] & [14], examining factors such as the number of vocabularies, sub-emotion clusters, average words per cluster (μW), standard deviation per cluster (σW), and masked users' sample text.

From the analysis, it is observed that the fastText model generates a larger vocabulary, while the google-news-300 model demonstrates superior cluster quality and compactness. Regular updates to the libraries of fastText, google-news-300, Twitter 25, and Twitter 200 differentiate them from [27] & [14]. Notably, Twitter serves as a valuable source for recent trends, usable words, and slangs. This analysis provides a foundation for further research to determine which model is better suited for specific use case.

In our ongoing work, the bag of sub-emotions generated from the four models is combined with machine learning and deep learning algorithms to detect mental illnesses such as depression, anxiety, and stress in social media posts, with the aim of achieving enhanced accuracy. In the future, this model can be leveraged to identify the emotional behavior of social media users' posts, offering a technological platform for doctors and suicide prevention teams to analyze and address mental disorders.

Author Contributions

For this research work all authors' have equally contributed in Conceptualization, methodology, validation, resources, writing—original draft preparation, writing—review and editing.

Conflicts of interest

We declare "There is no conflict of interest showed by authors while conducting this research work".

References

[1] S. Ghosh and T. Anwar, "Depression Intensity Estimation via Social Media: A Deep Learning Approach," in IEEE Transactions on Computational

Social Systems, vol. 8, no. 6, pp. 1465-1474, Dec. 2021, doi: 10.1109/TCSS. 2021.3084154.
[2] "DSM-5 Update: Supplement to Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition" (PDF). *PsychiatryOnline*. American Psychiatric Association Publishing. September 2016.
[3] Al-Ezzi, N. Yahya, N. Kamel, I. Faye, K. Alsaih and E. Gunaseli, "Severity Assessment of Social Anxiety Disorder Using Deep Learning Models on Brain Effective Connectivity," in IEEE Access, vol. 9, pp. 86899-86913, 2021, doi: 10.1109/ACCESS.2021.3089358.
[4] [online] https://datareportal.com/reports/digital-2022-global-overview-report.
[5] [online] https://www.hootsuite.com/
[6] Paul Ed Ekman and Richard J Davidson. 1994. The nature of emotion: Fundamental questions. New York, NY, US: Oxford University Press.
[7] K. A. Govindasamy and N. Palanichamy, "Depression Detection Using Machine Learning Techniques on Twitter Data," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 960-966, doi: 10.1109/ICICCS51141.2021.9432203.
[8] S. Tariq et al., "A Novel Co-Training-Based Approach for the Classification of Mental Illnesses Using Social Media Posts," in IEEE Access, vol. 7, pp. 166165-166172, 2019, doi: 10.1109/ACCESS. 2019.2953087.
[9] Uddin, M.Z., Dysthe, K.K., Følstad, A. et al. Deep learning for prediction of depressive symptoms in a large textual dataset. Neural Comput & Applic 34, 721–744 (2022). https://doi.org/10.1007/ s00521-021-06426-4.
[10] M. M. Tadesse, H. Lin, B. Xu and L. Yang, "Detection of Depression-Related Posts in Reddit Social Media Forum," in IEEE Access, vol. 7, pp. 44883-44893, 2019, doi: 10.1109/ACCESS. 2019.2909180.
[11] K. Hemmatirad, H. Bagherzadeh, E. Fazl-Ersi and A. Vahedian, "Detection of Mental Illness Risk on Social Media through Multi-level SVMs," 2020 8th Iranian Joint Congress on Fuzzy and intelligent Systems (CFIS), 2020, pp. 116-120, doi: 10.1109/CFIS49607.2020.9238692.
[12] Olteanu, Alexandra & Kiciman, Emre & Castillo, Carlos. (2018). A Critical Review of Online Social Data: Biases, Methodological Pitfalls, and Ethical Boundaries. WSDM '18: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. 785-786. 10.1145/ 3159652.3162004.
[13] M. Rizwan, M. F. Mushtaq, U. Akram, A. Mehmood, I. Ashraf and B. Sahelices, "Depression Classification From Tweets Using Small Deep Transfer Learning Language Models," in IEEE Access, vol. 10, pp. 129176-129189, 2022, doi: 10.1109/ACCESS.2022.3223049.
[14] Aragon, Mario Ezra,Lopez-Monroy, Adrian Pastor, Gonzalez-Gurrola, Luis Carlos and Montes-y-Gomez, Manue Montes-y-Gomez, "Detecting Depression in Social Media using Fine-Grained Emotions, "Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)", pages 1481-1486, jun- 2019, doi: 10.18653/v1/ N19-1151.
[15] Mohammad Saif M. and Turney Peter D, "Crowdsourcing a Word-Emotion Association Lexicon ", Computational Intelligence, vol. 29, pp. 436-465, 2013
[16] Mohammad Saif M. and Turney Peter D, "Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon", Computational Linguistics, pp. 26-34, 2010.
[17] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.
[18] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532-1543.
[19] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics, 5, 135-146.
[20] Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. Science, 315(5814), 972-976.
[21] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.
[22] Woolf P., Keating A., Burge C., and Michael Y, "Statistics and Probability Primer for Computational Biologists". Massachusetts Institute of Technology, BE 490/ Bio7.91, Spring 2004.
[23] Taylor, J. "An Introduction to Error Analysis". Sausalito, CA: University Science Books, 1982.
[24] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.
[25] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society: Series C (Applied Statistics), 28(1), 100-108.

[26] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. ACM Computing Surveys, 31(3), 264-323.

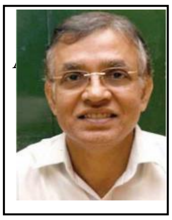
[27] Aragon, Mario Ezra,Lopez-Monroy, Adrian Pastor, Gonzalez-Gurrola, Luis Carlos and Montes-y-Gomez, Manue Montes-y-Gomez, "Detecting Depression in Social Media using Fine-Grained Emotions, "Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)", pages 1481-1486, jun- 2019.

Authors Biography

Srinath K S received the B.E. degree in Information science and Engineering and M. Tech degree in Networks and Internet Engineering from Visvesvaraya Technological University, India. Currently working as Research Scholar in department of Computer Science and Engineering in University Visvesvaraya College of Engineering, Bangalore University, Bengaluru, India. He can be contacted at email: solansrinath@gmail.com.

Kiran K is currently working as an Associate professor in the Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering, Bangalore. He completed his Ph.D in 2015 from Bangalore University. His areas of research include machine learning, natural language processing, cyber security, and wireless networks. He has published 25 papers in refereed International conferences and journals. He is also a senior member of IEEE and currently serving as execom member of IEEE Bangalore section. He can be contacted at email: kirank@uvce.ac.in.

P Deepa Shenoy is currently working as a Professor in the Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering, Bangalore. She did her doctorate in the area of Data Mining from Bangalore University in the year 2005. Her areas of research include data mining, soft computing, biometrics and social media analysis. She has published more than 200 papers in refereed International conferences and journals. She is also a senior member of IEEE and currently serving as chair of, IEEE Bangalore section. She can be contacted at email: shenoypd1@gmail.com.



Venugopal K R is a former Vice Chancellor, Bangalore University, Bangalore. He obtained his Bachelor of Engineering from University Visvesvaraya College of Engineering. He received his Masters degree in Computer Science and Automation from Indian Institute of Science Bangalore. He was awarded Ph.D. in Economics from Bangalore University and Ph.D. in Computer Science from Indian Institute of Technology, Madras. He has authored and edited 79 books on Computer Science and Economics and has published over 1000 research papers. His research interests include Computer Networks, Wireless Sensor Networks, Parallel and Distributed Systems, Digital Signal Processing and Data Mining. He can be contacted at email: venugopalkr@gmail.com