International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Unveiling the Role of Social Media in Mental Health: A GAN-based Deep Learning Framework for Suicide Prevention

Rohini Kancharapu*[1], Sri Nagesh Ayyagari[2]

Submitted: 27/04/2023 **Revised**: 27/06/2023 **Accepted**: 07/07/2023

Abstract: In recent years, there has been a significant increase in user participation on social networking media sites. These platforms generate vast amounts of diverse data that have a substantial impact on the mental health of the general public. Suicide, being a leading cause of death globally, has drawn the attention of researchers. The World Health Organization estimates that around 800,000 people died by suicide in 2019, with a significant portion falling between the ages of 15 and 29. The COVID-19 pandemic has further exacerbated the issue due to social isolation. Traditionally, the study of suicide has focused on physiological aspects using questionnaires and in-person settings. However, the effectiveness of such approaches is hindered by societal stigma. To address this, we propose a method that combines Generative Adversarial Networks (GANs) with various deep learning algorithms, including Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, Bidirectional Long Short-Term Memory (BI-LSTM) networks, and Gated Recurrent Units (GRUs). We also employ multiple feature engineering techniques, such as GloVe, Word2Vec, Fasttext, and Weighted Average Fusion (GloVe+Word2Vec+Fasttext). Our results demonstrate that the combination of CNN with Fasttext yields impressive recall, precision, and accuracy measures, with caps of 95%, 95%, and 92%, respectively. This research contributes to the field of suicide prevention by utilizing deep learning models and feature engineering methods to analyze social media data. By leveraging these techniques, we aim to enhance suicide detection and prevention efforts in the context of the widespread use of social networking media.

Keywords: *Deep Learning, Generative Adversarial Networks, Machine Learning, Suicidal keywords, Suicide, Twitter*

1. Introduction

The Social Media sites play a ubiquitous role in the Modern Age. It has brought the world so close that physical and geographical boundaries are crumbling down and has resulted in increased engagement of the users. These websites generate an enormous and varied amount of data. As a result, it is now a crucial foundation for identifying and analysing the behavioural traits of many individuals, including suicidal ideation. Numerous more try suicide than the 703,000 suicides that occur annually [1, 2]. Every suicide is a tragedy that has an impact on the people left behind as well as families, towns, and entire countries. It is the fourth most common cause of death worldwide among people aged 15 to 29 according to the most current World Health Organization (WHO) survey [3]. Social media platforms have become an integral part of the modern age, connecting people worldwide and generating vast amounts of diverse data. With the increasing engagement of users on these platforms, it has become crucial to analyze and identify behavioral traits, including suicidal ideation. Suicidal attempts far exceed the reported 703,000 suicides each year, and the impact of each tragedy extends to families, communities, and nations [1, 2]. Notably, suicide ranks as the fourth leading cause of death among individuals aged 15 to 29 globally, as per the World Health Organization (WHO) [3].

The COVID-19 pandemic further exacerbated the crisis of social isolation, highlighting the need to address mental health issues effectively. Conventional approaches, such as questionnaires and in-person assessments, face challenges due to societal stigma. Therefore, the development of a robust classification system is imperative to accurately identify and prevent suicidal intent, while also distinguishing it from non-risky suicide-related information. Previous systems have focused on identifying and understanding social media posts related to mental health issues, such as depression and suicide. For example, the Durkheim project [4] examined the online behaviors and content shared by a group of combat veterans on social media platforms to identify defining characteristics of harmful activity. Linguistics-based prediction models were created, achieving an accuracy rate of 65.3% in determining suicide risk based on recorded behaviors indicative of fear, restlessness, and hallucinations.

To overcome the challenges and create a concise and accurate solution, this study employs deep learning techniques for predicting suicidal risk among social media users. Sentiment analysis, a natural language processing (NLP) technique widely used for analyzing social media

1 Gayatri Vidya Parishad College of Engineering for Women, Kommadhi, Visakhapatnam-530048, Andhra Pradesh, INDIA
ORCID ID : 0000-0001-9364-4021
2 Rayapati Venkata Rangarao & Jagarlamudi Chandramouli College of Engineering, Chowdavaram, Guntur-522019, Andhra Pradesh, INDIA
ORCID ID : 0000-0002-4197-5969
** Corresponding Author Email: rohinik3108@gmail.com*

posts, is employed [5-9]. The prediction is categorized into three levels of concern: Low, High, and Very High. The experimentation focuses on Twitter posts, with the primary features extracted from the tweet content scraped using the Tweepy API. Word embedding techniques [10-15] are utilized to process and represent the text, forming the basis for building the classification network. After predicting the risk level, appropriate preventive measures, such as encouragement and resource notification, are employed to assist users accordingly.

2. Literature Review

The literature review in this chapter explores existing research and studies related to suicide detection and risk assessment using social media data. It begins by examining the work of Coppersmith et al. [5], who developed an effective approach for categorizing user postings to detect signs of suicide. The subsequent sections discuss various studies that have employed different techniques, including natural language processing (NLP), machine learning (ML), and deep learning algorithms, to identify suicide risk in social media data.

Coppersmith et al. [5] developed an effective approach for categorizing user postings for suicide detection. They utilized data from users who voluntarily contributed it, combining publicly available information with data from OurDataHelps.org. The authors computed contextual information between words using a bidirectional LSTM layer and employed a Text Classification model to determine the likelihood of an author attempting suicide. This study primarily focused on females aged 18 to 24, showing effectiveness for women but potential limitations for men in the same age range.

Rabani et al. [6] proposed a system based on natural language processing (NLP) and multi-class classification techniques for suicide risk assessment in tweets. The authors gathered a dataset of 7852 tweets over a two-year period and developed a hybrid feature engineering approach using techniques such as Term Frequency Inverse Document Frequency (TFIDF) and Bag Of Words (BOW). Decision trees consistently yielded the best results among the different algorithms tested. However, the limitations of this strategy included human annotation of tweets and insufficient data on suicide.

Ambalavan et al. [7] utilized NLP and machine learning techniques to explore suicidal behavior through Reddit threads. They collected data from the Reddit thread titled "Suicide survivors of Reddit, what was your first conscious thought after you discovered that you hadn't succeeded?" using the Python Reddit API Wrapper. The authors employed trigrams, NLTK POS Tags, and Customized POS Tags for data pre-processing and feature extraction. Their results showed that Support Vector Machines (SVM) achieved higher accuracy compared to Logistic Regression, SGD, and Perceptron. However, this approach was constrained by the size of the dataset and the time-consuming annotation process.

Sawhney et al. [8] developed a supervised strategy for suicide risk identification in tweets, incorporating feature selection using the Firefly algorithm. The authors collected tweets using the Twitter REST API and applied TFIDF to extract commonly occurring terms. They created three datasets, labeled as Dataset H, Dataset UNI, and Dataset SCO, and evaluated various models. The results indicated that CNN-LSTM and RF + Binary Firefly Algorithm achieved the highest accuracy, precision, recall, and F1-scores across all three datasets.

Shahreen et al. [9] employed Support Vector Machines (SVM) and neural networks for Twitter text classification related to suicide risk. Real-time tweets containing specific keywords were extracted using the Twitter Streaming API. The authors utilized techniques such as CountVectorizer (CV) and TF-IDF for feature extraction and employed optimizers including Limited Memory BFGS (LBFGS), Stochastic Gradient Descent (SGD), and Adam. The study, however, faced limitations due to the availability of data. SVM achieved 95.2% accuracy, while neural networks achieved 97.6% accuracy.

Birjali et al. [10] presented an algorithm using WordNet for semantic analysis of suicide-related tweets collected from Twitter. The authors employed machine learning techniques including Support Vector Machines (SVM), Maximum Entropy, and Naive Bayes for tweet classification. They utilized the Weka tool and compared the semantic similarity between training and test tweets. The study had limitations in terms of the amount of data available and achieved relatively lower accuracy compared to other classification techniques.

Burnap et al. [11] developed a multi-class classification approach for suicide-related communication on Twitter. They collected anonymous data from user postings on four well-known websites, creating a glossary of phrases. TF-IDF was used for feature extraction, and Part-of-Speech (POS) labels were assigned to words using the Stanford POS Tagger. The authors also employed the Linguistic Inquiry and Word Count (LIWC) program to extract more precise labels for affective emotions and feelings. Classifiers including Support Vector Machines (SVM), decision trees, and Naive Bayes were used. SVM achieved the highest recall value, while Naive Bayes exhibited the top precision.

2.1. Summary and Gaps in Current Literature

In conclusion, the literature review highlights the advancements made in the field of suicide detection using social media data. It showcases studies such as Rabani et al. [6], Ambalavan et al. [7], Sawhney et al. [8], Shahreen et al. [9], Birjali et al. [10], and Burnap et al. [11], which have contributed valuable insights into the topic. These studies have utilized diverse methodologies, including feature engineering, semantic analysis, and multi-class classification, to assess suicide risk in tweets and online discussions. However, despite these advancements, there are still gaps and limitations in the existing literature, such as the need for larger and more diverse datasets, improved annotation processes, and further exploration of gender-specific factors in suicide detection.

2.2. Implications for Current Research

The findings from the literature survey suggest that there is a promising potential for leveraging social media data and advanced analytics techniques for suicide prevention and early intervention. The reviewed studies have demonstrated the effectiveness of various algorithms, including LSTM, SVM, decision trees, and neural networks, in identifying suicide risk. These findings can guide future research efforts towards developing more accurate and robust models for real-time suicide risk assessment and intervention.

In conclusion, the literature review provides a comprehensive overview of the current state of research on suicide detection and risk assessment using social media data. It highlights the advancements, limitations, and gaps in the existing literature and emphasizes the potential implications for future research in this field. The insights gained from these studies can contribute to the development of effective strategies and tools for suicide prevention and mental health support in the digital age.

3. Methodology

The suggested approach seeks to fix the drawbacks listed above. Figure 1 illustrates, there are five main steps: (A) Data collection, (B) Annotation, (C) Feature engineering, (D) Data balancing and (E) classification.

Algorithm: Suicidal Risk Identification from Social Media Posts

Inputs:

- Twitter dataset with collected tweets
- Annotation labels indicating the sentiment of each tweet (negative, neutral, positive)
- Word embedding models: word2vec, GloVe, fasttext

Outputs:

- Predicted risk levels for each tweet (low, high, very high)

Steps:

1. *Data Collection:*

- Collect tweets using the Twitter API based on predefined keywords and expressions.
- Retrieve various attributes of each tweet, such as content, user information, retweet count, etc.

2. *Annotation:*

- Apply the VADER Sentiment Analyzer algorithm to assign polarity scores to each tweet.
- Categorize and label tweets based on polarity scores: negative, neutral, positive.

3. *Word Embedding:*

- Preprocess the collected tweets for feature engineering.
- Apply word embedding techniques (word2vec,GloVe, fasttext) to represent words as real-valued vectors.

4. *Weighted Average Fusion:*

- Combine the word vectors obtained from the word embedding techniques using a weighted average approach.
- Assign equal weights to each embedding technique to ensure a balanced contribution.

5. *Data Balancing:*

- Address the sample imbalance issue by employing data augmentation through Generative Adversarial Networks (GANs).
- Train the GAN model using the original dataset and generate artificial samples to balance the positive and neutral classes.

6. *Classification:*

- Preprocess the vectorized data obtained from word embedding and data balancing.
- Utilize deep learning models, such as CNN, LSTM, BI-LSTM, and GRU, for risk classification.
- Train the classification models using the preprocessed data and annotation labels.
- Predict the risk levels (low, high, very high) for each tweet based on the trained models.

7. *Output Analysis:*

- Analyze the predicted risk levels and associated tweets to identify users at risk of suicidal behavior.
- Implement appropriate preventive measures, such as encouragement and resource notification, based on the risk levels.

8. *Evaluation and Optimization:*

- Evaluate the performance of the classification models using metrics like accuracy, precision, recall, and F1-score.

- Optimize the models and algorithms based on the evaluation results to improve the accuracy and reliability of risk identification.

9. Iterative Process:

- Iterate and refine the methodology based on feedback and additional data to enhance the accuracy and effectiveness of suicidal risk identification.

3.1. Data Collection

Different watchwords and expressions used in previous papers as well as from various locations and forums were gathered to collect the data for our research [11, 12]. Data access was made possible by the Twitter Application Programming Interface (API). Various fields of inferring are included in the extracted dataset. 87 keywords in all, which are presented in Table 1, were gathered overall.

Table 1. Suicidal Keywords

To end this nightmare	Want to die	Life is so meaningless
Want to die	Ready to jump	don't want to live
Want to dissapear	Want it to be over	put an end to this
Could just fall asleep	Take my own life	want to be dead
Stop the pain	Tired of living	thoughts of suicide
End it all	Suicide plan	to hurt myself
I'm drowning	Sleep forever	want to be gone
Life is too hard	Better off dead	take it anymore
My life consists of nothing	Depressed	hate myself
Don’t want to exist	Wanna die	end this pain
My life is miserable	Can’t go on	die in my sleep
Tired of being alone	To take my own life	really need to die
My suicide letter	Go to sleep forever	killing myself
Don’t want to be here	My suicide note	my death would
Thoughts of suicide	I am worthless	Don’t want to wake up
Suicidal	Nothing to live for	my life is pointless
Die alone	Hate my life	i am leaving now
Ready to die	Ready to die	better off without me
Never wake up	Slit my wrist	Suicide plan
Suicide ideation	Slash my wrist	Sleep forever
Thoughts of suicide	End my life	Tired of living alone
To hurt myself	Depressed	My life is miserable
Want to die	Be dead	Life is so meaningless
Want to be dead	Die alone	don't want to live
Put an end to this	Tired of living	put an end to this
My death would	Cut my wrist	want to be dead
Wanna suicide	Asleep and never wake	thoughts of suicide
Commit suicide	Take my own life	to hurt myself
want to be gone	Commit suicide	my life is pointless
take it anymore	Suicidal ideation	i am leaving now
hate myself	do not want to be here	better off without me
end this pain	why should i live	Suicide plan
die in my sleep	be dead	Sleep forever
really need to die	die now	Tired of living alone
killing myself	end my life	My life is miserable
my death would	suicide pact	-
Don’t want to wake up	not worthy living	-

The Twitter API can be used to programmatically retrieve and analyze Twitter data, as well as build for the conversation on Twitter. The API allows a maximum of 2 million tweets per month. Using these keys and the collected Reddit keywords, tweets were collected over a span of 4 years i.e., from 2018 to 2022 in sporadic intervals. A total of 1,48,768 tweets were collected along various columns like, Screen Name, User Name, Tweet Content, Tweet ID, User Description, User Location, User URL, Tweet Coordinates, Retweeted, Retweet Count, Source, Truncated, Reply to user ID, Favourited, Favorite Count bringing the total size of the dataset to 51MB shown in Figure 2.

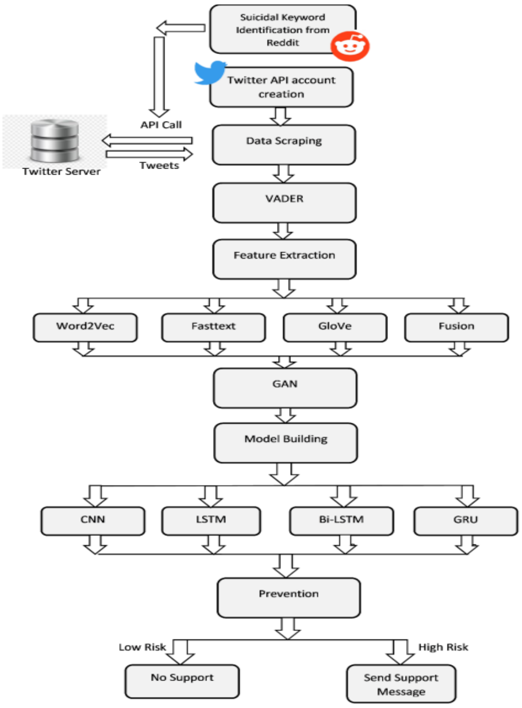

Fig. 1. Proposed framework

```
                                                 Tweet Created      Tweet id
0       RT @vicgerami: Maral can be your sister, daugh...  1.350000e+18
1       RT @AustinTSearle: I am running for a better f...  1.350000e+18
2       I am running for a better future. I do not and...  1.350000e+18
3       RT @KimFlorey2: @JolyonMaugham can @RishiSunak...  1.350000e+18
4       RT @vicgerami: Maral can be your sister, daugh...  1.350000e+18
...                                                   ...           ...
148764  Simon Peter said to him, "Let Mary leave us, f...  1.518632e+18
148765  Most relevant fact: Purposeless life is not TH...  1.518630e+18
148766  Dear #Asians\nThis is what Europeans Thinks ab...  1.518616e+18
148767  RT @Mercury_Prime: This is NOT an empty hype t...  1.518576e+18
148768  @TennisPodcast Do not suport the Wimbledon dec...  1.518571e+18
```

Fig. 2. Tweet Count

3.2. Annotation

The crucial step in training the machine learning (ML) model is annotation, where the emotional health of individuals is assessed. In this study, the VADER Sentiment Analyzer algorithm was employed for this purpose. VADER is a vocabulary and rule-based sentiment analysis tool specifically designed to analyze sentiments expressed in social media. Each tweet is assigned a polarity score by VADER, which falls into one of four categories: Positive, Negative, Neutral, and Compound. Based on these polarity scores, the tweets are categorized and labeled accordingly. Tweets that discuss suicide or express a desire to end one's life are classified as negative. Tweets exhibiting feelings of uneasiness, unhappiness, gloom, or other sentiments that may indirectly allude to suicide are categorized as Neutral. On the other hand, tweets that are sardonic, make references to the self-destruction of others, contain nerdy references, or feature unrelated content are assigned to the Positive category.

By utilizing the VADER Sentiment Analyzer algorithm and its polarity scores, the tweets are annotated and assigned appropriate labels, forming the basis for further analysis and classification in the subsequent stages of the ML model training process.

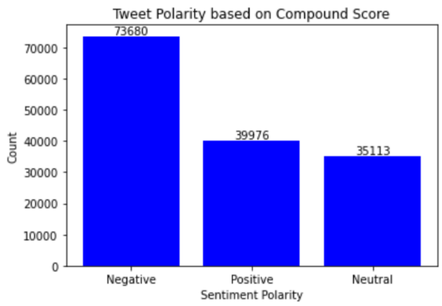

Fig. 3. Bar Graph of Label Counts

3.3. Word Embedding

The Feature engineering is a crucial step in preparing unprocessed data for supervised learning tasks. It involves selecting, modifying, and transforming the data into meaningful features that can be utilized by machine learning models. In the context of text analysis, one commonly used feature engineering technique is word embedding. Word embedding represents words as real-valued vectors, capturing the meaning and semantic relationships between words. The underlying idea is that words with similar meanings should have similar vector representations, and words that appear in similar contexts should be close in the vector space. Word embedding models fall into two primary categories: frequency-based models and prediction-based models.

Frequency-based models rely on the statistical properties of words in a corpus, such as term frequency-inverse document frequency (TF-IDF) or co-occurrence counts, to learn word representations. However, these models have limitations in terms of deterministic representations and vocabulary size. Prediction-based models, on the other hand, use machine learning algorithms to predict words based on their contexts. These models have been shown to be effective in capturing semantic relationships and are widely used for tasks like word analogies and similarities. Word2vec, GloVe, and Fasttext are examples of prediction-based word embedding models commonly employed in text analysis.

Word2vec utilizes shallow neural networks, such as CBOW and Skip-gram, to learn word vectors by predicting surrounding words given a target word or vice versa. GloVe (Global Vectors for Word Representation) constructs word embeddings by considering the co-occurrence statistics of words in a large corpus. Fasttext extends Word2vec by considering word subword n-grams, which is particularly useful for capturing meaning in rare or out-of-vocabulary words.

These prediction-based word embedding models enable the transformation of textual data into numerical representations, enhancing the effectiveness of machine learning algorithms in natural language processing tasks.

3.3.1. Word2vec

Word2vec is a popular word embedding technique that combines two algorithms: Continuous Bag of Words (CBOW) and Skip-gram. Both CBOW and Skip-gram are shallow neural network models used to map words to target variables, which are also words or sets of words. These algorithms aim to learn weights that represent word vectors.

3.3.1.1. Continuous Bag of Words (CBOW)

The CBOW (Continuous Bag of Words) operates by estimating the likelihood of a word given its context. The context can be a single word or a set of words, depending on the specific scenario. To prepare the corpus for training a CBOW model, one-hot encoding is applied to represent the words. This encoding transforms the corpus into a suitable training set.

In the CBOW architecture, a shallow neural network with three layers is employed. These layers include an input layer, a hidden layer, and an output layer. The input layer receives the encoded context words as input. The hidden layer processes the input and performs computations to capture the underlying patterns and relationships. Finally, the output layer utilizes a softmax function to generate a probability distribution over the vocabulary, indicating the likelihood of each word in the vocabulary being the target word given the provided context.The shallow neural network structure of CBOW allows for efficient training and prediction, with the softmax function providing a normalized probability distribution. This enables the model to predict the target word based on the given context. By training the CBOW model on a large corpus and optimizing the weights and biases through backpropagation, the model can learn to accurately estimate the target word given its context. The CBOW architecture is commonly used in natural language processing tasks and has demonstrated effectiveness in capturing semantic relationships between words.

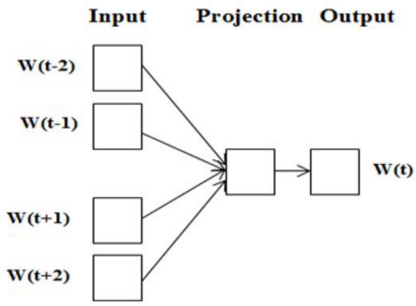

Fig. 4. Architecture of CBOW

3.3.1.2. Skip-gram

While CBOW (Continuous Bag of Words) and Skip-gram are two different architectures used in Word2Vec. While CBOW takes the context as input and predicts the target word, Skip-gram takes the target word as input and predicts the surrounding context words. The choice of architecture determines the direction of prediction. In the Skip-gram architecture, the input is a single word, and the goal is to predict its context words. The context window defines the range of neighboring words to consider on both sides of the target word. For example, with a context window of 1, we consider one word before and one word after the target word.

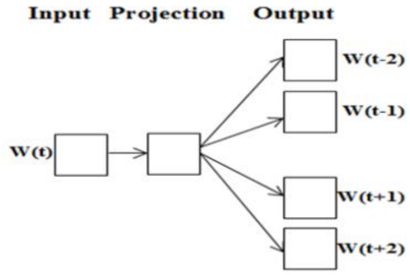

Fig. 5. Architecture of Skip-gram

Figure 5 illustrates the architecture of Skip-gram, where the target word is used as input, and the context words are predicted. In this architecture, there are two one-hot encoded target variables and two corresponding outputs, reflecting the context words on both sides of the target word. Word2Vec incorporates both CBOW and Skip-gram models and utilizes techniques such as softmax function and negative sampling. These techniques, combined with backpropagation, help represent words as vectors.

The loss function in Word2Vec is formulated as follows:

$$J(\theta) = \frac{-1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0}\log\big(p(wt+j|wt)\big) \quad (1)$$

where T represents the total number of training samples, m is the context window size, and p(wt+j | wt) represents the conditional probability of the context word given the target word.

Weight calculation in the model involves matrix multiplication, as shown by the equation:

$$h = W^{t}.x \quad (2)$$

where W^(t) represents the weight matrix and x represents the input vector.

These equations and calculations form the foundation of the Word2Vec model and enable the generation of word vectors based on the training data.

3.3.2. Fasttext

FastText is an extension of Word2Vec that introduces the concept of character n-grams in addition to word-level representations. Unlike Word2Vec, which treats each word as the smallest unit, FastText recognizes that a word can be composed of n-grams of characters. This is particularly useful for representing uncommon words or out-of-vocabulary (OOV) words.To illustrate this, let's consider the phrase "I want to kill myself." If we set n=3, we can generate character n-grams for the word "suicide." In this case, the possible n-gram outcomes would be sui, ici, ide, and icu. FastText defines a distributional representation of words by considering the frequency of occurrence for each word. The distribution is defined as follows:

$$P(w) = \sqrt{\frac{i}{f(w)}} + \frac{1}{f(w)} \qquad (3)$$

where $f(w) = \frac{count_w}{\text{total no. of tokens}}$

Here, f(w) represents the frequency of occurrence for word w, and i is a selected threshold (e.g., t = 10e-4). The distribution equation assigns higher weights to less frequent words, allowing them to contribute more to the word representation. In practice, FastText leverages both word-level and character n-gram information to generate more robust and informative word vectors, especially for rare or unseen words. This approach enhances the ability to capture meaningful semantic relationships within the text data.

3.3.3. Glove

GloVe, which stands for global vectors forpresentation, is an approach that utilizes statistics to establish the relationships between words. The key idea behind GloVe is to construct a co-occurrence matrix that captures the connections between words. To illustrate this concept, let's consider the two statements: "I literally wanna die" and "I literally don't wanna live." We extract all the distinct words from both sentences and represent them as rows and columns in a matrix. We then check if the combination of a column and a row exists in the statements and count how many times that combination occurs. For example, in our given example, the combination of "I" and "I" occurs 0 times, while the combination of "I" and "literally" occurs twice.

Table 2. Co-occurrence Matrix

	I	literally	wanna	die	don’t	live
I	0					
literally	2	0				
wanna	2	2	0			
die	1	1	1	0		
don’t	1	1	1	0	0	
live	1	1	1	0	1	0

Based on this co-occurrence matrix, we can calculate the co-occurrence probabilities using the formula:

$$F(w_i, w_j, w_k) = \frac{P_{ij}}{P_{jk}} \qquad (4)$$

where P_ij represents the co-occurrence count of words w_i and w_j, and P_jk represents the total count of word w_j. This probability measure provides valuable information about the relationships between words.

3.3.4. Weighted Average Fusion

To create fusion vectors, the word vectors obtained from the word embedding techniques, namely word2vec, fasttext, and GloVe, are combined element-wise. In order to leverage the strengths of each technique and address the limitations of individual methods, equal weights are assigned to each technique. Specifically, each method is given a weight of 33%, as depicted in Figure 7. It is important to note that the composition of the vectors should remain unchanged, and the sum of all weights must add up to 100%. The weighted average formula for fusion vectors is as follows:

$$W_{fusion} = \frac{(W_{word2vec} + W_{fasttext} + W_{glove})}{3} \qquad (5)$$

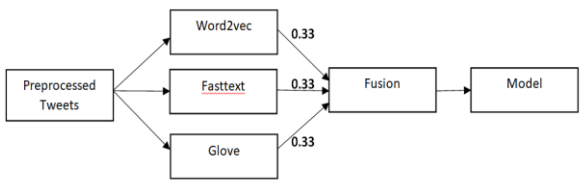

Fig. 7. Weighted Average of Word Embedding Techniques

The main data source for this study is Twitter data, which undergoes preprocessing using word embedding techniques such as word2Vec, fasttext, GloVe, and fusion (word2Vec + GloVe + fasttext). The output of these embedding techniques is transformed into vectorized data, which is then used to train models including CNN, BI-LSTM, LSTM, and GRU. The figures from 8 to 11 depict these models. The ultimate goal is to detect suicidal risk based on the trained models and their analysis.

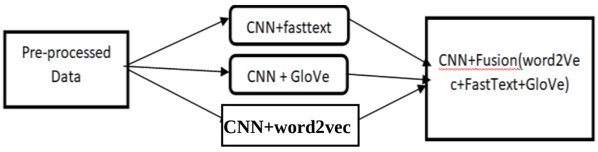

Fig. 8. Fusion of each embedding technique with CNN

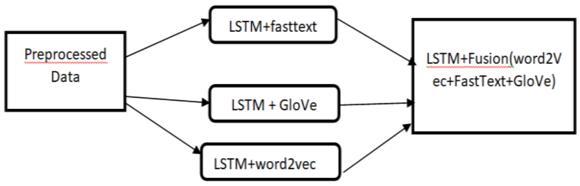

Fig. 9. Fusion of each embedding technique with LSTM

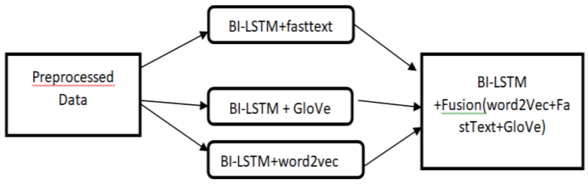

Fig. 10. Fusion of each embedding technique with Bi-LSTM

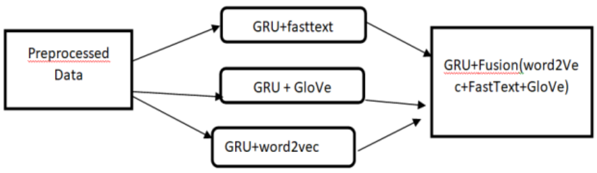

Fig. 11. Fusion of each embedding technique with GRU

3.4. Data Balancing

The dataset used in this research contains a significantly lower number of Positive and Neutral tweets compared to Negative tweets, resulting in a sample imbalance issue for the risk identification algorithms. To address this issue, data augmentation techniques can be employed. In this study, oversampling is chosen over undersampling to prevent information loss and maintain model performance.

Data augmentation involves using Generative Adversarial Networks (GANs) to generate artificial samples based on the original data. GANs consist of two sub-models: the generator and the discriminator. The generator is trained to create new instances by taking random samples as input, effectively generating synthetic data to balance the dataset. On the other hand, the discriminator is responsible for distinguishing between real and fake samples. Figure 12 illustrates the architecture of GANs, depicting the generator and discriminator sub-models. The generator generates plausible instances, while the discriminator classifies the data into two groups: Real and Fake. By iteratively training the GANs, the generator learns to create synthetic samples that closely resemble the real data, effectively addressing the sample imbalance issue.

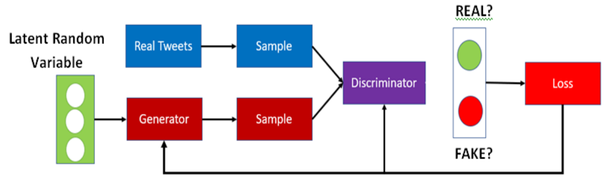

Fig. 12. GAN Architecture

3.5. Classification

In this study, we approach the task of identifying suicidal thoughts in tweets as a supervised multi-classification problem. After preprocessing the dataset to remove unnecessary data, we focus on two main columns: tweet labels and titles. Inspired by previous work [13], we formulate our multi-classification problem using a scheme where T represents the tweet and L represents the label. The labels are represented as [1,0,0] for the Negative class, [0,1,0] for the Neutral class, and [0,0,1] for the Positive class. Our machine learning model aims to predict the class with the least amount of error. To accomplish this, we employ deep learning models such as CNN, LSTM, BI-LSTM, and GRU in our research work

4. Results and Discussion

A total of 35113 neutral tweets, 39976 positive tweets (non-suicidal risk), and 73680 negative (suicidal risk) tweets were extracted. Using word2vec, GloVe, Fasttext, and a combination of all three algorithms, the tweets were pre-processed. The network was then built using four deep learning techniques: CNN, LSTM, BI-LSTM, and GRU. For the purposes of training and testing the model, the dataset was split 80:20. We utilised Python to carry out our work. The four deep learning methods offer varying degrees of accuracy. The outcomes produced by the different models were really impressive. Among all four deep learning models, best performance was given by CNN+ Fasttext with an F1-measure of 0.97, 0.94 for Low Risk (Positive and Negative tweets), and 0.96 for High Risk (Negative tweets) respectively. Furthermore, this research used balanced data. The accuracy of all predictions made using various models and word embedding strategies is shown in Table 3.

The combination of training loss and validation loss over time is a commonly used metric in evaluating model performance. The validation loss indicates how well the model generalizes to new data, while the training loss measures its performance on the training data. Accuracy is another widely used metric that provides a straightforward understanding of how well the model predicts. By analyzing these two observations together, we can gain more insights into the model's behavior. If the accuracy is low and the loss is high, it suggests that the model is making significant errors across the majority of the data. Conversely, if the model produces modest errors across the majority of the data, both the loss and accuracy would be low. However, if both the loss and accuracy are high, it indicates that the model is making significant mistakes on a subset of the data.

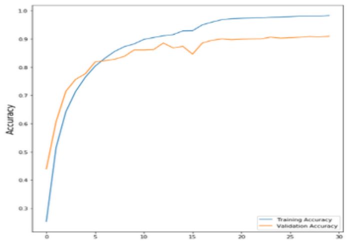

Fig. 13. CNN with Fasttext Accuracy

The ideal scenario is when the model makes small errors on only a portion of the data, which is reflected in high accuracy and low loss. This indicates that the model is performing well in terms of prediction accuracy. Figure 13 illustrates the model accuracy curve using training and testing datasets. The graph clearly shows that the validation accuracy surpasses the training accuracy, indicating that the model is effective in generalizing to new data and is well-suited for building a new predictive model.

The model loss curve with training and testing datasets is shown in Figure 14.

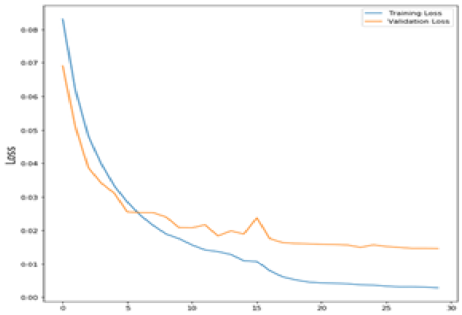

Fig. 14. CNN with Fasttext Loss

The performance of the model is examined using this graph utilising learning curves. The loss of the model is frequently lower on the training dataset than on the validation dataset, and it is seen that the training and validation curves are reducing to a point of stability with a minimum "generalisation gap." Thus, it is determined that the learning curve fits the situation well. Table 4 presents the accuracies of CNN for different word embedding strategies. Furthermore, we have conducted an analysis to evaluate the changes in accuracy over time by varying the number of epochs and batch sizes. This examination provides valuable insights into the performance of the CNN model under different training configurations. Table 5 showcases the accuracy of LSTM for various word embedding strategies. Additionally, we have conducted an analysis to explore how the accuracy of the LSTM model fluctuates with different batch sizes and epoch counts. This investigation provides insights into the impact of these factors on the model's performance.

Table 6 presents the accuracies of Bi-LSTM for different word embedding methods. Additionally, we have analyzed the impact of batch size and the number of epochs on the accuracy over time. By varying these parameters, we investigated how the accuracy of the Bi-LSTM model changes. Table 7 displays the accuracies of GRU for different word embedding strategies. Furthermore, we have examined the variations in accuracy over time by utilizing different numbers of epochs and batch sizes. This analysis allows us to observe how the accuracy of the GRU model has both increased and decreased sizes. Table 8 presents the accuracy comparison of different algorithms with and without GAN (Generative Adversarial Network) for various word embedding techniques. The table provides accuracy percentages for four deep learning models: CNN, LSTM, Bi-LSTM, and GRU.

Table 3. Performance Evaluation of Models Using Various Word Embedding Techniques

Deep Learning Models	Word2vec			Fasttext			Glove			Fusion		
	Precision (%)	Recall (%)	F1-score (%)	Precision (%)	Recall (%)	F1-score (%)	Precision (%)	Recall (%)	F1-score (%)	Precision (%)	Recall (%)	F1-score (%)
CNN	92	93	92	94	93	93	91	91	91	93	94	93
LSTM	93	93	92	94	94	94	92	92	92	95	95	95
Bi-LSTM	91	91	91	94	94	94	90	87	91	94	94	94
GRU	90	92	89	94	94	94	91	92	88	94	94	94

Table 4. CNN Accuracies

Model	Word2vec				Fasttext				GloVe				Fusion			
Epochs/ Batch Size	**3**	**5**	**7**	**10**	**3**	**5**	**7**	**10**	**3**	**5**	**7**	**10**	**3**	**5**	**7**	**10**
32	91.01	90.85	92.18	95.43	93.82	99.27	99.51	98.82	93.66	91.32	91.78	91.5	91.63	91.46	91.34	91.42
64	92.01	90.45	92.55	93.63	95.36	99.59	97.5	99.5	97.22	98.61	91.93	91.5	91.7	91.59	91.62	91.66
128	91.63	91.6	92.7	92.58	94.71	98.56	99.16	99.66	98.93	91.57	91.79	91.45	91.56	91.19	91.51	91.36

Table 5. LSTM Accuracies

Model	Word2vec				Fasttext				GloVe				Fusion			
Epochs/ Batch size	**3**	**5**	**7**	**10**	**3**	**5**	**7**	**10**	**3**	**5**	**7**	**10**	**3**	**5**	**7**	**10**
32	82.54	82.95	87.54	87.89	88.95	88.25	86.2	89.95	87.65	87.6	87.95	87.6	87.54	87.89	88.95	88.25
64	82.95	90.95	89.5	82.97	85.4	88.5	87.57	88.54	87.65	87.6	90.6	87.6	89.5	85.97	85.47	88.54
128	90.5	92.95	92.2	87.54	86.3	89.5	87.95	87.54	87.65	86.7	87.95	87.46	88.2	87.54	86.35	89.54

Table 6. BI-LSTM Accuracies

Models	Word2vec				Fast text				GloVe				Fusion			
Epoch/ Batch size	**3**	**5**	**7**	**10**	**3**	**5**	**7**	**10**	**3**	**5**	**7**	**10**	**3**	**5**	**7**	**10**
32	75.74	80.12	82.43	83.59	79.37	77.38	76.6	80.18	81.05	80.22	81.96	80.7	80.75	81.11	80.52	78.4
64	81.7	78.65	83.95	81.44	88.17	88.37	88.04	86.51	77.75	79.97	77.41	80.36	77.92	82.26	77.84	85.02
128	77.29	74.32	84.11	78.6	88.1	87.66	87.49	88.41	80.38	83.45	81.83	82.95	79.65	77.88	79.52	83.51

Table 7. GRU Accuracies

Models	Word2vec				Fasttext				GloVe				Fusion			
Epochs/ Batch Size	**3**	**5**	**7**	**10**	**3**	**5**	**7**	**10**	**3**	**5**	**7**	**10**	**3**	**5**	**7**	**10**
32	91.72	93.74	95.05	96.38	95.07	92.2	91.91	92	91.23	79.47	77.99	76.3	91.97	92.32	91.81	91.39

64	92.17	96.57	97.64	98.47	92.24	92.27	91.98	91.91	91.63	81.06	78.5	73.12	91.85	92.22	92.12	78.6
128	89.89	97.91	98.69	99.13	92.18	91.82	92.05	92.12	91.67	77.5	78.17	78..23	91.2	92.08	92.08	86.51

Table 8. Accuracy Comparison of Different Algorithms

DL Models	Without GAN				With GAN			
	Word2Vec	**Fasttext**	**GloVe**	**Fusion**	**Word2Vec**	**Fasttext**	**GloVe**	**Fusion**
CNN	88.75%	92.01%	89.1%	89%	68.67%	67.94%	74.3%	75.93%
LSTM	88.91%	90.66%	86.69%	87%	70.5%	67.5%	76.7%	76.05%
Bi-LSTM	89.9%	89.9%	88.7%	82.53%	65.4%	68.02%	71.2%	74.87%
GRU	86.5%	86.5%	88.7%	83.4%	73.4%	67.57%	67.8%	75.05%

Without GAN, the accuracy percentages vary across the different word embedding techniques. For CNN, the highest accuracy is achieved with the Fasttext embedding (92.01%), followed by Word2Vec (88.75%), GloVe (89.1%), and Fusion (89%). For LSTM, the highest accuracy is obtained with the Word2Vec embedding (88.91%), followed by Fasttext (90.66%), GloVe (86.69%), and Fusion (87%). Bi-LSTM achieves the highest accuracy with the Word2Vec embedding (89.9%), followed by Fasttext (89.9%), GloVe (88.7%), and Fusion (82.53%). Similarly, for GRU, the highest accuracy is achieved with the GloVe embedding (88.7%), followed by Word2Vec (86.5%), Fasttext (86.5%), and Fusion (83.4%).

With GAN, the accuracy percentages show variations compared to the models without GAN. For CNN, the accuracy decreases for all word embedding techniques. The highest accuracy is obtained with GloVe (74.3%) and Fusion (75.93%). For LSTM, the accuracy also decreases, with the highest accuracy achieved with Fusion (76.7%) and the lowest with Fasttext (67.5%). Bi-LSTM shows a mixed trend, with some improvements and some decreases in accuracy. The highest accuracy is obtained with Fasttext (74.87%). For GRU, the accuracy increases for some word embedding techniques, with the highest accuracy achieved with Fusion (75.05%).

Overall, the table provides insights into the impact of GAN and different word embedding techniques on the accuracy of the deep learning models. It highlights the variations in accuracy across the models and the importance of selecting appropriate word embedding techniques for achieving higher accuracy in suicide risk prediction**.**

5. Conclusion

In this research, we undertook a comprehensive analysis of a large dataset consisting of 148,768 tweets related to suicide. Our objective was to develop a deep learning model that could accurately identify tweets with a high likelihood of suicidal content. Through the construction of the deep learning network, we successfully identified 73,680 tweets that exhibited a significant possibility of individuals expressing suicidal thoughts or intentions. To facilitate the prediction of suicidal tendencies, we employed three distinct feature extraction approaches to effectively represent the tweets during the classification process. This multi-faceted approach allowed us to capture various aspects of the tweets, enhancing the accuracy of our predictions. By combining these feature extraction techniques with the power of deep learning, we established a robust prediction function capable of analyzing and assessing the likelihood of suicide in individual tweets.

The implications of this research extend beyond mere data analysis. The insights and predictions generated by our model can be utilized as a powerful suicide watch tool, enabling proactive intervention and support for individuals at risk. Particularly in the context of social media platforms like Twitter, where users often express their emotions and thoughts openly, our model serves as a valuable resource for identifying individuals who may be experiencing depression or other mental health issues. By leveraging the predictive capabilities of our model, online psychiatrists and mental health professionals can be alerted to individuals who are exhibiting signs of depression or suicidal ideation. This enables timely and targeted interventions, allowing for virtual consultations and support that can potentially save lives. Overall, our research highlights the potential of combining advanced machine learning techniques with social media data to improve mental health monitoring and provide timely assistance to those in need.

5.1. Future Scope

In the future, there are several potential enhancements to consider. Firstly, incorporating a notification system that alerts authorities, family members, or friends when a user's tweets indicate serious suicidal intent can provide immediate intervention and support. This can help ensure a timely response and potentially save lives. Secondly, expanding the scope of the research beyond Twitter to include other social media platforms such as Instagram and Tumblr can provide a more comprehensive understanding of suicidal behavior across different platforms. This broader analysis can uncover unique patterns and insights specific to each platform, contributing to more effective prevention strategies. Additionally, improving the support system by obtaining users' contact information beyond social media platforms can enable direct communication with individuals in crisis. This approach ensures that support is not solely reliant on the user granting permission to receive messages from anyone and allows for more proactive intervention.

Furthermore, exploring the integration of advanced embedding techniques like BERT (Bidirectional Encoder Representations from Transformers) and ELMo (Embeddings from Language Models) can enhance the performance of the prediction models. These state-of-the-art embedding methods have shown promising results in various natural language processing tasks and can potentially improve the accuracy and precision of suicide risk prediction. By addressing these future considerations, the research can expand its reach, improve the support system, and leverage advanced techniques to further enhance the effectiveness of suicide prevention efforts in the realm of social media.

Authors' contributions

Both authors collaborated in developing the research, analyzing the findings, and making significant contributions to the final manuscript. Both authors thoroughly reviewed and provided their approval for the final version of the manuscript.

Conflicts of interest

No conflicts of interest exist that we need to mention.

References

[1] Suicide." [Online]. Available: https://www.who.int/newsroom/fact-sheets/detail/suicide. [Accessed: 12-Aug-2020].

[2] V. Patel et al., "Suicide mortality in India: A nationally representative survey," *The Lancet*, vol. 379, no. 9834. Lancet Publishing Group, pp. 2343–2351, 2012.

[3] G. Copper smith, K. Ngo, R. Leary, and A.Wood, "Exploratory Analysis of Social Media Prior to a Suicide Attempt, "*Proc. ThirdWork. Comput. LinguisticsClin. Psychol.*, pp.106–117, 2016

[4] S. T. Rabani, Q. Rayees Khan and A. M .UdDin Khanday, "Multi-Class Suicide Risk Predictionon Twitter Using Machine Learning Techniques," 2020 2nd International Conference on Advances in Computing,CommunicationControlandNetworking(ICACCCN),2020,pp.128-134,doi:10.1109/ICACCCN51052.2020.9362979.

[5] D. O. Blessed et al., "Sentiment Analysis with Word Embedding", *Sentiment Analysis View project Fairness in Learning Analytics View project Sentiment Analysis with Word Embedding*, 2019. [Online]. Available: https://www.researchgate.net/publication/332091221

[6] A. Rohrbach et al., "Movie Description," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 94–120, May 2017, doi: 10.1007/s11263-016-0987-1.

[7] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Expert Systems with Applications*, vol. 72, pp. 221–230, Apr. 2017, doi: 10.1016/j.eswa.2016.10.065.

[8] J. Song, S. Tang, J. Xiao, F. Wu, and Z. (Mark) Zhang, "LSTM-in-LSTM for generating long descriptions of images," *Computational Visual Media*, vol. 2, no. 4, pp. 379–388, Dec. 2016, doi: 10.1007/s41095-016-0059-z.

[9] S. Zhang, E. Grave, E. Sklar, and N. Elhadad, "Longitudinal analysis of discussion topics in an online breast cancer community using convolutional neural networks," *Journal of Biomedical Informatics*, vol. 69, pp. 1–9, May 2017, doi: 10.1016/j.jbi.2017.03.012.

[10] S. Al-Azani and E. S. M. El-Alfy, "Using Word Embedding and Ensemble Learning for Highly Imbalanced Data Sentiment Analysis in Short Arabic Text," in *Procedia Computer Science*, 2017, vol. 109, pp. 359–366. doi: 10.1016/j.procs.2017.05.365.

[11] K. P. S. Sankar, P. C. R. Raj, and V. Jayan, "Unsupervised Approach to Word Sense Disambiguation in Malayalam," *Procedia Technology*, vol. 24, pp. 1507–1513, 2016, doi: 10.1016/j.protcy.2016.05.106.

[12] Subject Index," *Procedia Computer Science*, vol. 95, pp. 527–531, 2016, doi: 10.1016/j.procs.2016.09.336.

[13] A. G. Chifu and S. Fournier, "SegChainW2V:

Towards a Generic Automatic Video Segmentation Framework, Based on Lexical Chains of Audio Transcriptions and Word Embeddings," in *Procedia Computer Science*, 2016, vol. 96, pp. 1371–1380. doi: 10.1016/j.procs.2016.08.182.

[14] B. Tang, H. Cao, X. Wang, Q. Chen, and H. Xu, "Evaluating word representation features in biomedical named entity recognition tasks," *BioMed Research International*, vol. 2014, 2014, doi: 10.1155/2014/240403.

[15] M. M. Tadesse, H. Lin, B. Xu, and L. Yang, "Detection of suicide ideation in social media forums using deep learning," *Algorithms*, vol. 13, no. 1, Jan. 2020, doi: 10.3390/a13010007.

[16] Ambalavan AK, Moulahi B, Azé J, Bringay S. Unveiling Online Suicide Behavior: What Can We LearnAbout Mental Health from Suicide Survivors of Reddit? *Stud Health Technol Inform*. 2019 Aug 21;264:50-54.doi: 10.3233/SHTI190181. PMID: 31437883

[17] Sawhney, R., Shah, R.R., Bhatia, V., Lin, C.T., Aggarwal, S., and Prasad, M., Exploring the Impact of Evolutionary Computing based Feature Selection in Suicidal Ideation Detection. *IEEE International Conference on Fuzzy Systems*.2019.

[18] Shahreen,N.,Subhani,M.,andMahfuzurRahman,M.,S uicidal Trend Analysis of Twitter Using Machine Learning and Neural Network. 2018 *International Conference on Bangla Speech and Language Processing*,ICBSLP2018. https://doi.org/10.1109/ICBSLP.2018.8554733.

[19] Birjali, M., Beni-Hssane, A., and Erritali, M., Machine Learning and Semantic Sentiment Analysis based Algorithms for Suicide Sentiment Prediction in *Social Networks. The 8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks* (EUSPN2017). 113, 65–72, 2017. https://doi.org/10.1016/j.procs.2017.08.290.

[20] Burnap, P., Colombo, G., Amery, R., Hodorog, A., and Scourfield, J., Multi-class machine classification of suicide-related communication on Twitter. *Online Social Networks and Media*. 2,32–442017.https://doi.org/10.1016/j.osnem.2017.08.001.

[21] C. Poulin, B. Shiner, P. Thompson, L. Vepstas, Y. YoungXu, B. Goertzel, B. Watts, L. Flashman, and T.McAllister. Predicting the risk of suicide by analyzing the text of clinical notes. *PLoS ONE*,9 (2014).

[22] American Psychiatric Association, Diagnostic and Statistical Manual of Mental Disorders. *American Psychiatric Association*, 2013.

[23] B. O'Dea, S. Wan, P. J. Batterham, A. L. Calear, C. Paris, and H. Christensen, "Detecting suicidality on twitter," *Internet Interv.*, vol. 2, no. 2, pp. 183–188, 2015

[24] S. Ji, C. P. Yu, S.-F. Fung, S. Pan, and G. Long, "Supervised Learning for Suicidal Ideation Detection in Online User Content," *Hindawi Complex.*, vol. 2018, pp. 1–10, 2018.

[25] C. Sitaula, A. Basnet, and S. Aryal, "Vector representation based on a supervised codebook for Nepali documents classification," *PeerJ Computer Science*, vol. 7, Article ID e412, 2021.

[26] S. Boon-Itt and Y. Skunkan, "Public perception of the COVID-19 pandemic on twitter: sentiment analysis and topic modeling study," *JMIR Public Health and Surveillance*, vol. 6, no. 4, Article ID e21978, 2020.

[27] L. Nemes and A. Kiss, "Social media sentiment analysis based on COVID-19," *Journal of Information and Telecommuni- cation*, vol. 5, no. 1, pp. 1–15, 2021.

[28] B. N. Ramya, S. M. Shetty, A. M. Amaresh, and R. Rakshitha, "Smart simon bot with public sentiment analysis for novel COVID-19 tweets stratification," *SN Computer Science*, vol. 2, no. 3, pp. 1–11, 2021.

[29] U. Naseem, I. Razzak, M. Khushi, P. W. Eklund, and J. Kim, "COVIDsenti: A large-scale benchmark twitter data set for COVID-19 sentiment analysis," *IEEE Transactions on Com- putational Social Systems*, vol. 8, 2021.

[30] K. Garcia and L. Berton, "Topic detection and sentiment analysis in twitter content related to COVID-19 from Brazil and the USA," *Applied Soft Computing*, vol. 101, Article ID 107057, 2021.

[31] F. Rustam, M. Khalid, W. Aslam, V. Rupapara, A. Mehmood, and G. S. Choi, "A performance comparison of supervised machine learning models for COVID-19 tweets sentiment analysis," *PLoS One*, vol. 16, no. 2, Article ID e0245909, 2021.

[32] M. E. Basiri, S. Nemati, M. Abdar, S. Asadi, and U. R. Acharrya, "A novel fusion-based deep learning model for sentiment analysis of COVID-19 tweets," *Knowledge- Based Systems*, vol. 228, Article ID 107242, 2021.

[33] H. Kaur, S. Ul Ahsaan, B. Alankar, and V. Chang, "A proposed sentiment analysis deep learning algorithm for analyzing COVID-19 tweets," *Information Systems Frontiers*, pp. 1–13, 2021.

[34] M. S. Satu, M. I. Khan, and M. Mahmud, "TClustVID: a novel machine learning classification

model to investigate topics and sentiment in COVID-19 tweets," *Knowledge-Based Sys- tems*, vol. 226, Article ID 107126, 2021.

[35] A. S. Imran, S. Muhammad Daudpota, Z. Kastrati, and R. Batra, "Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on COVID-19 related tweets," *IEEE Access*, vol. 8, pp. 181074–181090, 2020.

[36] R. Chandrasekaran, V. Mehta, T. Valkunde, and E. Moustakas, "Topics, trends, and sentiments of tweets about the COVID-19 pandemic: temporal infoveillance study," *Journal of Medical Internet Research*, vol. 22, no. 10, Article ID e22624, 2020.

[37] J. Xue, J. Chen, R. Hu et al., "Twitter discussions and emotions about the COVID-19 pandemic: machine learning approach, "*Journal of Medical Internet Research*, vol. 22, no. 11, Article ID e20550, 2020.

[38] T. de Melo and C. M. S. Figueiredo, "Comparing news articles and tweets about COVID-19 in Brazil: sentiment analysis and topic modeling approach," *JMIR Public Health and Surveil- lance*, vol. 7, no. 2, Article ID e24585, 2021.

[39] T. B. Shahi and A. K. Pant, "Nepali news classification using Naive Bayes, support vector machines and neural networks," in Proceedings of the *International Conference on Commu- nication Information and Computing Technology (ICCICT)*, pp. 1–5, IEEE, Mumbai, India, February 2018.

[40] A. Basnet and A. K. Timalsina, "Improving Nepali news recommendation using classification based on LSTM recur- rent neural networks," in Proceedings of the *IEEE 3rd Inter- national Conference on Computing, Communication and Security (ICCCS)*, pp. 138–142, IEEE, Kathmandu, Nepal, October 2018.

[41] Ms. Mayuri Ingole. (2015). Modified Low Power Binary to Excess Code Converter. International Journal of New Practices in Management and Engineering, 4(03), 06 - 10. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/38

[42] Veeraiah, V., Pankajam, A., Vashishtha, E., Dhabliya, D., Karthikeyan, P., & Chandan, R. R. (2022). Efficient COVID-19 Identification Using Deep Learning for IoT. 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), 128–133. IEEE.

[43] George, N. ., & B. K., A. . (2023). Hypervolume Sen Task Scheduilng and Multi Objective Deep Auto Encoder based Resource Allocation in Cloud. International Journal on Recent and Innovation Trends in Computing and Communication, 11(4s), 16–27. https://doi.org/10.17762/ijritcc.v11i4s.6303