Suicidal Ideation Detection Based on Social Media Data in the Context of Bangladesh

By

Name: Suraia Jabeen and ID: 012211090

Submitted in partial fulfilment of the requirements
of the degree of Masters of Science in Computer Science and Engineering

September 9, 2024

Department of Computer Science and Engineering
United International University

Abstract

The increasing use of social media paves the way for solving problems like suicide and urge the need to detect and analyze suicidality. People are suffering from suicidal ideation(SI) across the world. The urgency of early intervention and suicide prevention programs cannot be overstated. Social media platforms are known for self-expression and emotional sharing which offer an opportunity for early detection of SI. Our study aims to use data from social media platforms like Facebook, Twitter, and Reddit. By using machine learning (ML) to the data, we can automatically identify at-risk individuals. We employ two ML based techniques: one topic modeling based deep learning (DL) and one graph neural network (GNN) based models. Our deep learning based model achieves an outstanding accuracy of 87%, while the graph neural network based model achieves a remarkable accuracy of 78%. The deep learning based model surpasses traditional machine learning methods, such as SVM, random forest, and Naive Bayes. Our study may contribute to policy-making approaches which can facilitate early intervention, suicide prevention, and developing a comprehensive decision support system.

Keywords: *Machine Learning, Deep Learning, Graph Neural Network, Suicidal Ideation, and Social Media.*

12pt

Acknowledgements

I am grateful to the almighty to provide me with strength to start my thesis journey and come at this stage. I would like to express my deepest appreciation to the individuals and my institution to play pivotal roles in the successful completion of this research endeavor.

At first, I would like to express my sincere gratitude to my supervisor, Prof. Dr. Md. Saddam Hossain Mukta, for his unwavering guidance, invaluable insights, and unwavering support throughout this research. His mentorship and commitment to excellence have been a constant source of inspiration.

I would also like to extend my gratitude to the thesis committee of United International University, United City, Madani Avenue, Satarkul, Badda, Dhaka, Dhaka 1212, Bangladesh, as well as the esteemed university authorities, including Prof. Dr. Md. Abul Kashem Mia, the Vice-Chancellor, Prof. Dr. Hasan Sarwar, Dean, Prof. Dr. Md. Nurul Huda, Department Head, and Prof. Dr. Md. Motaharul Islam, MSCSE Director. Their vision, leadership, and the resources provided by the university have been instrumental in facilitating my thesis through this research process.

Secondly, I am profoundly grateful to my research collaborators for their collaboration, dedication, and invaluable contributions to this research. Their ideas and support have enriched the project and made it a truly collaborative effort.

I would also like to acknowledge the support and encouragement I have received from my friends, family, and well-wishers. Their unwavering belief in my abilities and their encouragement during the research journey have been a source of strength and motivation.

Finally, I would like to express my heartfelt appreciation to all the individuals who supported and believed in this research, directly or indirectly. Your contributions, no matter how small, have been meaningful and appreciated.

This research would not have been possible without the collective efforts and support of all these individuals and instructions. Thank you for being an essential part of this journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Suicide is the act of taking one's own life intentionally which is a growing problem globally and the act becomes prevalent among the individuals of different culture, geographic, religious, social and economic boundaries [1], [2], [3]. Suicide is the eighth leading cause of death in Bangladesh [4]. Those who have no control over themselves may choose the wrong path like suicide. Suicidal ideation (SI) represents thoughts, plans, or intentions related to self-harm or suicide of an individual. People with SI prevalence may reveal signs and symptoms that indicate a person may be at risk of harming themselves [6, 7, 8, 38]. These people are highly in need of appropriate steps to intervene and provide support [9]. Traditional SI detection approaches are either interviews, questionnaire or survey based. Those approaches are self explanatory and people with SI might not be interested to provide actual information of themselves about their suicidal thoughts and mental health condition [8, 10, 11]. In recent times, social media has become integral part of our daily lives where we share our thoughts, ideas, and opinion [8, 10]. These media become important resource for the researchers to investigate human behavior and mental health status [11]. However, majority of the people with SI likely express their thoughts through the posts on different social media platforms. Thus, social media platforms can be a great choice for making early interventions of SI. The number of the users of the social media platforms like Facebook, Twitter and Reddit are increasing in Bangladesh day by day. According to the data published in Meta's advertising resources in early 2023, Facebook has 43.25 million users and Twitter has 1.05 million users in Bangladesh [12]. Several studies [13],[14],[15],[16],[17],[18] and [19] have been conducted using traditional machine learning (ML) techniques to detect SI from the social media posts. However, to find SI prevalence users, a suitable annotation guidelines from social media posts and cutting edge hybrid ML and deep learning (DL) based methods with higher accuracy are yet to explored.

1.1 Motivation

Several studies [13, 16, 18, 19] have been appeared in the literature to detect SI from different content such as text, image, etc. by using ML and DL techniques. Vioules et al. [13] uses concatenated features of N-gram, term frequency-inverse document frequency (TF-IDF) [14], and linguistic inquiry and word count (LIWC) [15] to detect SI from Reddit posts. Different natural language processing, computational linguistics, and deep learning techniques are used in [16] to detect SI from Facebook, Twitter and Instagram posts. Model fusion is used to analyze and detect suicidal data. An LSTM-CNN [17] combined model is used in [18] to evaluate SI and compare to other classification models. Murad et al. [19] finds out that a deep learning based model with CNN+BiLSTM has outperformed with 0.61 F1-Score in Fasttext word embedding methods. Majority of the existing techniques do not use ensemble of traditional ML or DL techniques. In particular, Graph neural network (GNN) [20] is one of the latest deep learning techniques which was not utilized before over social media dataset and not obtained higher accuracy. In addition, these studies largely suffer of creating a high quality dataset and they do not reveal any clear guidelines about how do they prepare and annotate dataset. To address this gap, we endeavor to develop a high quality dataset with clear annotation guidelines by the domain expert and develop a strong SI prediction model by using advanced DL based hybrid approach from social media posts.

In our study, we use Facebook, Twitter and Reddit posts to analyze and detect SI. We collect a total of 785 social media posts from these social media sites by using data scrawling and scrapping techniques with search keywords ”Suicide”, ”suicidal”, ”self injury”, ”self harm”, and many more related words. The collected posts are annotated by using annotation rules instructed by one behavioral scientist. A total of 405 posts are annotated as suicidal and the rest 380 posts are annotated as non-suicidal. We use two approaches to perform the experiments. First, we develop a semi-supervised deep learning model or Long Short Term Memory (LSTM) [17] to detect SI. We convert the social media posts into vectors by using TF-IDF. The converted vectors are used in Latent Semantic Indexing (LSI) [21] for topic modeling to extract the semantic relationship among the words in the posts. Then, we utilized LSI to identify contextual topics which correlate with SI. After topic modeling, we use word to vector (word2vec) [22] to create features. We follow the above steps for extracting vectors of correlated topics with SI. However, the above methods is used for sentence level vectorization. Again, we find the vector of complete sense of the post, we use a pre-trained language model like Bidirectional Encoder Representation of Transformer (BERT) [23]. Later, to retain the maximum information, we merge both word2vec and BERT feature vector for LSTM [17] model training. Second, we develop a Graph Neural Network [20] model. In the GNN model, we use psycho-linguistic features from a dictionary like LIWC [15] to understand users' psychological condition well. We create Graph using K-nearest neighbour from the features and apply GNN on the graph for suicidal text classification. GNNs [20] are well-suited for data with a graph structure,

which is often the case with social media networks. We also make a comparison with baseline models to find out the best one in terms of performance metrics. The comparison helps to fine tune our model performance to identify SI correctly.

1.2 Objectives

The main objective of our research is to suggest an automated system which can perform the detection of suicidal ideation from social media posts and can take part in suicide prevention policy making approaches. We target to implement a machine learning based model to integrate into an automated decision support system. To find out the best performant model with robustness, our goal is to implement and experiment with the model through different data size and type. We aim to perform hyper parameter tuning to those models for optimization and generalization. We want to discuss the usability and strength of different natural language processing and machine learning techniques that we opt to use in our model to specify the techniques for developing the objectives.

1.3 Contributions

- To construct a dataset from Facebook, Twitter and Reddit posts and to implement a novel DL and GNN based models for detecting user's SI from their social media posts.
- To propose a hybrid model which uses LSI output to LSTM model as input for improving word semantics over text classification.
- To propose a psycoliguistic lexical features based GNN model.
- To make a comparison of our built model with the conventional ML based approaches to understand the performance of our model.

To ensure it's real life implications, implementation of ethical guidelines and ensuring of compliance with privacy regulations is needed. Real-time data processing, model prediction latency, and prompt intervention strategies are required to provide timely support to individuals at risk overcoming challenges related to it.

Chapter 2

Background and Related Works

In this chapter, first, we describe relevant background studies of our research work. Later, we present related research of our work.

2.1 Background

We have used different procedures and technologies of artificial intelligence in the way of development in our research work. We have used different natural language processing techniques such as Word to Vector (Word2Vec)[75], Term Frequency-Inverse Document Frequency (TF-IDF)[76], Linguistic Inquiry and Word Count (LIWC)[82], Latent Semantic Indexing (LSI)[81], Bidirectional Encoder Representations from Transformers (Bert)[79] etc. to process the social media data and make it machine readable. Those techniques are used in the feature extraction process. In our feature selection and ranking, we have used different statistical methods such as Analysis of Variance (ANOVA)[84], Boruta[85], Fisher's Score[87]. We have used different machine learning techniques such as Long Short Term Memory (LSTM)[88], Graph Neural Network (GNN)[83] etc. We are collectively representing all of the techniques as a background study of our work.

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

TF-IDF transforms the text into meaningful representation of integers or numbers [76]. This transformed data is used to fit machine learning algorithm for predictions. TF-IDF transforms a word by comparing the number of times a word appears in a document of words with the number of documents the word appears in [76]. Term Frequency and Inverse Document Frequency can be described as follows :

Term Frequency: In a document d, the term frequency represents the number of instances of a given word t [76]. A word becomes more relevant when it appears in the text, which is rational. In a document, the weight of a term that occurs is proportional to the term frequency [76]. Term frequency of a word t in a document d can be described as follows:

tf = count of t in d / number of words in d.

Inverse Document Frequency: Inverse document frequency tests the relevant word where tf considers all terms equally significant [76]. First, it finds the document frequency of a term t by counting the number of documents containing the term [76]:

$$df(t) = N(t) \tag{2.1}$$

where *df(t)* = Document frequency of a term t and *N(t)* = Number of documents containing the term t

Inverse document frequency of a word t in a document d can be described as follows:

$$idf(t) = N/df(t) = N/N(t) \tag{2.2}$$

The more common word is considered as less significant. The logarithm (with base 2) of the inverse frequency of the paper can be considered [76]. In that case, the inverse document frequency of the term t becomes:

$$idf(t) = log(N/df(t)) \tag{2.3}$$

TF-IDF is more efficient than Count Vectorizer [76]. It focuses both on the frequency of words present in the corpus and provides the importance of the words. Less important words can be removed for analysis. Thus, it makes the model building less complex by reducing the input dimensions [76].

2.1.2 Word to Vector (word2vec)

Word2vec is a two-layer neural net that is used to process text by "vectoring" words. Its input is a corpus of text. Its output is a set of vectors known as feature vectors. Feature vectors are used to represent words the corpus. Word2vec is not a deep neural network but it turns text into a numerical form so that deep neural networks can understand[75].

The main purpose of word2vec is to group the vectors of similar words together in the vector space. It mathematically detects the word similarities[75]. If enough data, usage and contexts are given, word2vec can make highly accurate prediction about a word's meaning based on its past appearances. This prediction can be used to establish association between two words. For example: "man" is associated with "boy" where "woman" is associated with "girl"[75]. Also, it is used to cluster documents and classify them by topic. It works by measuring cosine similarity between words.

2.1.3 Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers is a language model which is introduced by researchers at Google in 2018 [77, 78]. BERT is used to pre-train deep bidirectional representations from unlabeled texts. Through the pre-training, BERT model can be used with just one additional output layer to fine-tune with the state-of-the-art

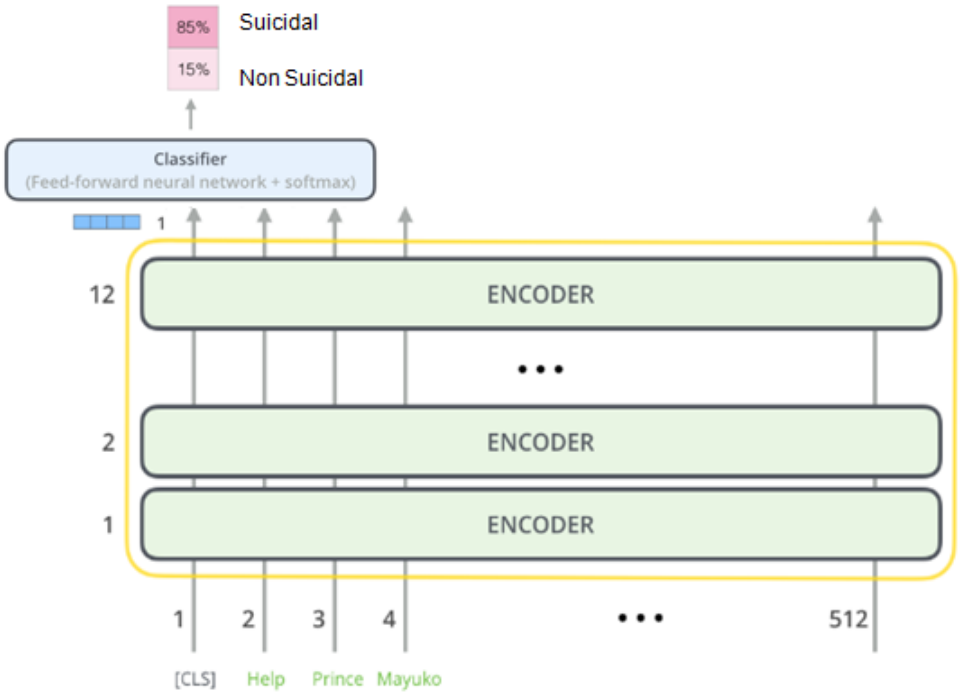

Figure 2.1: BERT base architecture for suicidal text classification.

models for different tasks. BERT can be used in question answering and language inference without modifications of basic task-specific architecture [79].

BERT considers all the words of the input sentence simultaneously. It uses attention mechanism to develop contextual meaning of the given words. BERT architecture evolved from transformers.There are several stacked encoder cells inside it just like transformers but only the encoder part of the transformer is used in the case of BERT. BERT has self-attention heads and a feed-forward neural network just like the encoder cells in transformer. BERT has two architectures known as BERT base and BERT large. Both architectures take an input of 512-dimensions. BERT base has 12 layers (transformer blocks), 12 attention heads, 110 million parameters, and has an output size of 768-dimensions. BERT Large has 24 layers (transformer blocks), 16 attention heads, 340 million parameters, and has an output size of 1024-dimensions. BERT was pre-trained on a large corpus of unlabelled text along with the entire Wikipedia (2,500 million words) and book corpus (800 million words). It has two training methods. The methods are Masked Language Model (MLM) Next Sentence Prediction (NSP) [80].

2.1.4 Latent Semantic Indexing (LSI)

LSI is one of the processes of Natural Language Processing (NLP) which is a subset of linguistics and information engineering. NLP focuses on the interpretation of human language into machine language. LSI understands and classifies words with similar contextual meanings within large data sets [81].

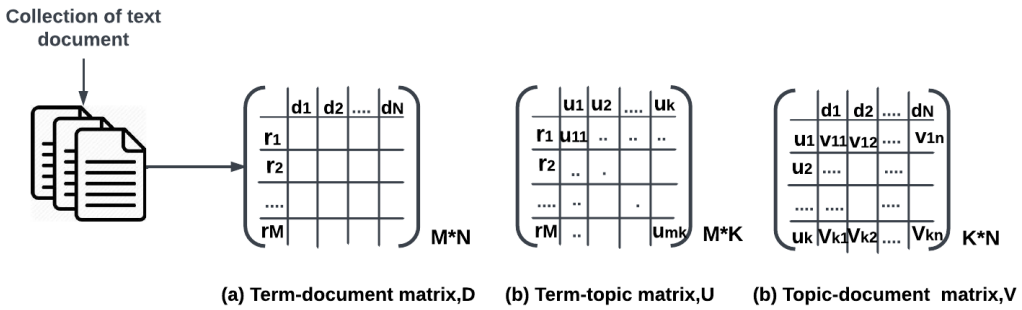

Figure 2.2: Matrix decomposition mechanism in SVD.

LSI works using a mathematical operation named as Singular Value Decomposition (SVD) as a partial application. SVD reduces a matrix into its constituent parts for simple and efficient calculations. The pre-processed words of a string are placed into a Term Document Matrix (TDM). A TDM is a 2D grid that counts the frequency of each specific word (or term) occurs in the data set of documents which is decomposed into Term Topic Matrix (TTM) and Topic Document Matrix (TDM) [81].

Weighing functions are used in the TDM. Classification of the documents is performed that contain the word with a value of 1 and with a value of 0. Words can be occurred with the same general frequency in these documents which is called word co-occurrence [81].

LSI uses Singular Value Decomposition (SVD) which produces vectors by predicting the meaning more accurately than analysing the individual terms [81]. The working procedure of SVD through the matrix decomposition mechanism is shown on figure 2.2.

2.1.5 Long Short Term Memory (LSTM)

LSTM is a special type of neural network which is designed to overcome the long-term dependency problem faced by Recurrent Neural Networks (RNN). Recurrent neural networks have vanishing gradient problem which is responsible for the long-term dependency. LSTM has feedback connections which makes it different from the feed forward neural network. This feedback connections enable LSTM to process the entire sequences of data. (e.g. time series). Thus, LSTM is handy to process sequential data such as text, speech and general time-series [88].

LSTM consists of four neural networks and different cells known as memory blocks. Information is stored in the cells and the memory is manipulated in the gates [88]. In an LSTM, there are three gates –

- Forget Gate: The unimportant information in the cell state is removed with the forget gate. Two inputs x_t (input at the certain state of time) and h_{t-1} (previous cell output) are fed to the gate and multiplied with weight matrices and then it is added to the bias. The result is then passed through an activation function which

gives a binary value as an output. The output is 0 for a particular cell state when the piece of information is forgotten and 1 when the information is retained for future use [88]. The forget gate has the following equation :

$$f_t = (W_f \cdot [h_t - 1, x_t] + b_f) \tag{2.4}$$

where, W_f denotes the weight matrix associated with the forget gate,$[h_t - 1, x_t]$ denotes the concatenation of the current input and the previous hidden state, b_f denotes the bias with the forget gate and is the sigmoid activation function.

- Input gate: The additional useful information is stored in the cell state by the input gate.The information is first regulated by the sigmoid activation function and filtered using inputs h_{t-1} and x_t similar to the forget gate. Then, *tanh* function is used to create a vector that gives an output from -1 to +1 that contains all the possible values from h_{t-1} and x_t. After that, the vector values and the regulated values are multiplied to obtain useful information [88]. The input gate has the following equation :

$$i_t = (W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.5}$$

$$_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{2.6}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot _t \tag{2.7}$$

where $\odot$ denotes element-wise multiplication and *tanh* denotes tanh activation function.

- Output gate: The output gate decides if useful information from the current cell state to be extracted and presented as output. For that, *tanh* generates a vector on the cell. Then, the information is regulated by the sigmoid activation function and filtered using inputs h_{t-1} and x_t. After that, the values of the vector and the regulated values are multiplied to send as an output and input to the next cell. The output gate has the following equation :

$$o_t = (W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.8}$$

The output of an LSTM at a certain time is basically dependant on three measures. The current long-term memory of the network which is known as the cell state. The output at the previous iteration of that time which is known as the previous hidden state and the input data at the current time state [88]. Cell structure of an LSTM along with gates is shown on figure 2.3.

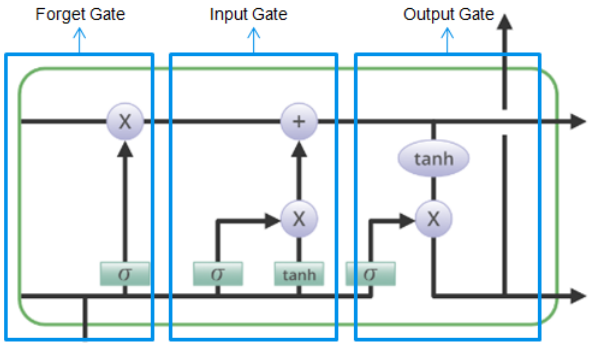

Figure 2.3: LSTM cell architecture

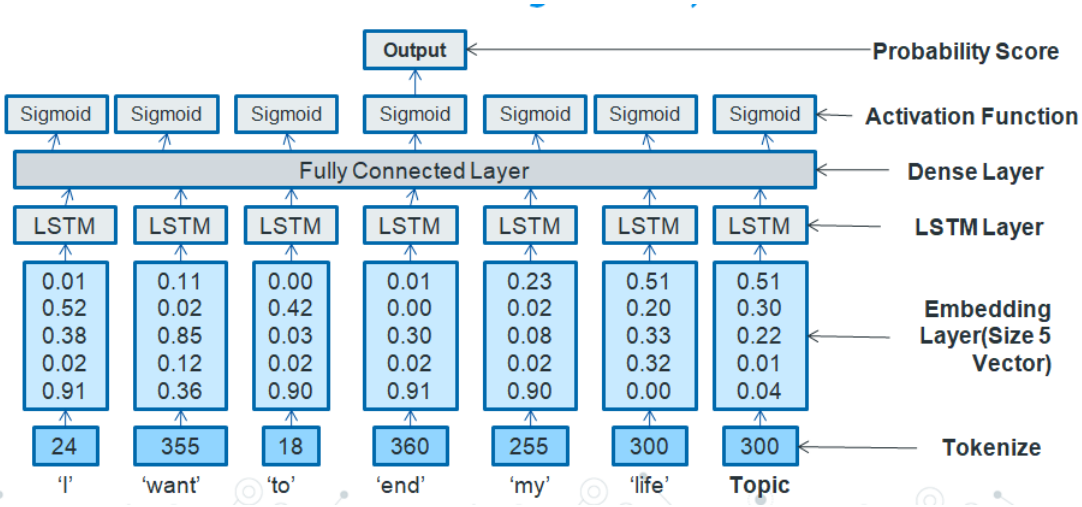

Figure 2.4: LSTM network architecture for suicidal text classification

LSTM is widely used in sentiment analysis. For a given sentence, LSTM first converts the words into tokens (integers). The tokens are then passed onto the embedding layer where the word tokens are converted into embedding of specific size. The next one is the LSTM Layer which is defined by hidden state dimensions and number of layers. The fully connected layer is responsible for mapping the output of LSTM layer into a desired output size. Sigmoid activation function is used in the Sigmoid activation layer where it turns all output values in between 0 and The final output of this network is the sigmoid output from the last time step [88]. The layer-wise working mechanism of LSTM in sentiment analysis is shown on Figure 2.4.

2.1.6 Linguistic Inquiry and Word Count (LIWC)

Linguistic Inquiry and Word Count (LIWC) is a lexicon used in Natural Language Processing to extract features from text. Basically, it helps to extract the emotions, and sentiments behind texts and helps to understand a wide variety of psycho-linguistic char-

acteristics from the text [82].

LIWC includes a dictionary with more than 90 categories which covers linguistic, psychological, and emotional dimensions. These categories include linguistic elements like pronouns, articles, and prepositions, as well as psychological categories such as emotions, cognitive processes, and social dimensions[102]. LIWC is often employed for sentiment analysis to help determine the emotional tone of text. Researchers use LIWC to identify positive, negative or neutral sentiment in text data. LIWC is also used to analyze text data related to mental health and well-being by identifying the linguistic patterns associated with conditions such as depression, anxiety, or suicidal ideation[103]. LIWC uses dictionary to calculate the word occurrences and compares with the words from the dictionary.It calculates the percentage of total words in the text that match each of the dictionary categories [82].

2.1.7 Analysis of Variance (ANOVA)

Analysis of Variance (ANOVA) is a statistical method in which the means of two or more groups is checked that are significantly different from each other [84]. Its also called the Fisher analysis of variance and the extension of the t- and z-tests [84]. It assumes Hypothesis as :

1. H0: Means of all groups are equal [84].
2. H1: At least one mean of the groups are different [84].

Its an analytical tool that splits an aggregate variability which is found inside a data set into two parts: systematic factors and random factors. The systematic factors have a statistical influence but the random factors do not in a dataset. ANOVA test is used to find out the influence that independent variables have on the dependent variable [84].

A one-way ANOVA is used in a three or more groups of data to find out the relationship between the dependent and independent variables. In case of finding no true variance between the groups, the ANOVA's F-ratio should be close to 1 [84]. ANOVA uses the following formula to calculate the ANOVA coefficient:

$$F = MSE/MST \tag{2.9}$$

where F=ANOVA coefficient, MST=Mean sum of squares due to treatment and MSE=Mean sum of squares due to error.

2.1.8 Boruta

Boruta acquires the top of the Random Forest algorithm as the name Boruta originates from the name of the spirit of the forest in Slavic mythology [85]. The traditional Random Forest mechanism works by estimating feature importance. In Random Forest, each decision tree creates a prediction and stores it. Then, the values of the features are permuted

randomly through the training samples and the previous step is followed and tracing of the result of the predictions is repeated. Thus, the importance of a feature is calculated by considering the average of the measurements across all trees for that feature. But, the calculation of the Z-scores for each feature is not done here. That's where, Boruta comes into play [85].

At first, a copy of the training set features is created and merged them with the original features. Random permutations on these synthetic features is done to remove correlation between those features and the target variable y. Synthetic features can be found randomly at each new iteration. Z-score is calculated for all the original and synthetic features [85]. The Z Score follows the following formula for one sample :

$$Z = (x–)/ \tag{2.10}$$

For multiple samples, the formula is as follows :

$$Z = (x–)/(/\sqrt{n}) \tag{2.11}$$

Relevant features is considered based on the importance of the features only if those are higher important than the maximum important of all synthetic features. Applies a statistical test on all original features and keeps memory of its results. The null hypothesis is that the importance of a feature is equal to the maximal importance of synthetic features. Statistical test is performed to measure the equality between the original and synthetic features. If the importance of a feature is significantly higher or lower than one of those of synthetic features, the null hypothesis is rejected. Unimportant features are removed from both the original and synthetic dataset. The iteration continues for nth time until all of the features are removed or considered important [85].

2.1.9 Fisher's Score

Fisher score is a supervised feature selection method which selects each feature independently based on the scores under the Fisher criterion. Thus,it leads to a suboptimal subset of features [87]. The main purpose of Fisher score is to find a subset of features and the data space can be spanned by the selected features. Data points in different classes are made to be distant and the distances between them as large as possible where the data points in same classes are made to be closer and the distances between them as small as possible. If the selected m features is given, the input data matrix is $X \epsilon R^{d \times n}$ reduced to $Z \epsilon R^{m \times n}$. The Fisher Score is computed as follows,

$$F(Z) = tr\{(\bar{S_b})(\bar{S_t} \dot{+} \gamma I)^{-1}\} \tag{2.12}$$

where γ is a positive regularization parameter, S^b is called between-class scatter matrix, and $S^t$is called total scatter matrix, which are defined as

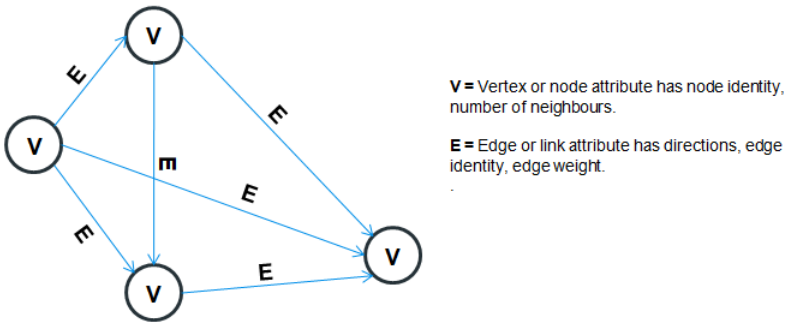

Figure 2.5: Basic graph structure

$$\bar{S}_b = \sum_{k=1}^{c} n_k(\bar{\mu_k} - \bar{\mu})(\bar{\mu_k} - \bar{\mu})^T \bar{S}_t = \sum_{i=1}^{n}(Z_i - \bar{\mu})((Z_i - \bar{\mu})^T, \tag{2.13}$$

where $\bar{\mu_k}$ and n_k are the mean vector and size of the k-th class respectively in the reduced data space.

Let μ^j and σ^j denote the mean and standard deviation of the whole data set corresponding to the j-th feature. Then the Fisher score of the j-th feature is computed below,

$$F(x^j) = \frac{\sum_{k=1}^{c} n_k(\mu_k^j - \mu^j)^2}{(^j)^2} \tag{2.14}$$

where $(^j)^2 = \sum_{k=1}^{c} n_k(^j_k)^2$. After computation of the Fisher score for each feature, the top-m ranked features are selected with large scores. The score of each feature is computed independently and the heuristic algorithm is used to select features which is suboptimal.

2.1.10 Graph Neural Network (GNN)

Graph consists of a set of objects, and the connections between objects. The objects are named as nodes/vertex and their corresponding connections are named as edges. The nodes have information about node identity and the number of neighbors. The edges have information about edge identity and edge weight. Based on the direction of edge connections, graphs can be of two types such as directed graph and undirected graph. A basic graph structure is illustrated in figure 2.5.

Images can be represented as graphs where each pixel represents a node and is connected via an edge to its adjacent pixels. We can digitize a text by creating a simple directed graph, where each character or index can be represented as a node and is connected via an edge to its adjacent node. Molecules can be described as a graph, where nodes can be represented as atoms and edges as covalent bonds. Graph can be used to solve different level tasks such as node level task, edge level task and graph level task [83].

The main goal of a graph-level task is to predict the property of an entire graph. For

example, a molecule an be represented as a graph and using the graph-level task, we can predict how the molecule smells like. In node-level task, the identity or role of each node within a graph is predicted and for edge-level task, the connection of the edges within a graph is predicted [83].

Social networks can be represented as graph data. Social networks can be used as tools to study patterns in collective behaviour of people, institutions and organizations. Graph can be used to represent groups of people by modelling individuals as nodes, and their relationships as edges [83].

A Graph Neural Network (GNN) is a type of neural network that operates on graph. Its an optimizable transformation on all attributes of the graph (nodes, edges, global-context) that preserves graph symmetries (permutation invariances). GNN makes predictions by pooling information [83].

In this research work, we have used graph neural network to perform sentiment classification and node-level task is performed. We have used the suicidal text data collected from social media to calculate sentiment score of suicidal thoughts and tasks using the graph neural network.

2.2 Related Works

Researchers find out patterns on suicidal behaviour from social media posts using different machine learning techniques. Rahim et. al.[24] proposes a solution to detect depression/suicidal ideation using NLP and DL techniques. They use transformers and a unique model to train the proposed model and apply it to three different data sets: SuicideDetection, CEASEv2.0, and SWMH. The proposed model outperforms the state-of-the-art in the SuicideDetection and CEASEv2.0 data sets with F1 score of 97% and 75% respectively.

Mehar et. al.[25] presents an approach that analyses social media platform Twitter to identify suicide warning signs for individuals. DL and ML based classification models are applied to tweets to detect SI. A stacked CNN - 2 Layer LSTM model is used to evaluate and compare with other classification models. Stacked CNN - 2 Layer LSTM architecture with word embedding technique achieves 93.92% accuracy compared to previous CNN - LSTM approach. Jere et al.[26] uses a relational Graph Attention Network (RGAT) with a DNN classifier to detect suicidal ideation and twitter posts are used for the ideation detection. They use BERT for word embedding of the tweeter posts to find relation between topics and sentimental words. To find the topics, they use Latent Dirichlet Allocation (LDA). To extract sentimental words, they use sentiment-based Lexicon approach. They employ RGAT to relate the features from LDA and BERT as well as the lexicon-based sentiment approach. Both of the RGAT outcomes are concatenated and classified using the DNN algorithm. The accuracy of classification of suicidal ideation of the proposed model is 88.21%.

Renjith et al.[27] proposes a combination of LSTM and CNN models to detect the emotions from posts of the users. In order to improve the accuracy, they suggest some

approaches like use of more data for training and use of attention model to improve the efficiency of existing models. They propose a LSTM-Attention-CNN combined model to analyze social media submissions to detect any underlying suicidal intentions. This model demonstrates an accuracy of 90.3% and an F1-score of 92.6%, which is greater than the baseline models. Islam et al.[19] creates a noble Bangla suicidal posts dataset named BanglaSPD and compares with the performance of various machine learning and deep learning models for suicide attempt prediction by training and evaluating with the dataset. They are able to find out that a deep learning-based model with CNN+BiLSTM has outperformed with 0.61 F1-Score in Fasttext word embedding methods. The comparative analysis of the existing studies which are closely related to our work is illustrated Table 2.1.

Table 2.1: Comparison of related literature

Literature and Year	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Renjith et al. [27], 2022	N	N	N	N	Y	N	N
Islam et al. [19], 2022	Y	Y	Y	N	Y	Y	N
Jere et al. [26], 2023	N	Y	Y	N	Y	Y	Y
Rahim et al. [24], 2024	Y	N	N	N	Y	Y	N
Mehar et al. [25], 2023	N	N	N	N	Y	N	N
Our Paper	Y	Y	Y	Y	Y	Y	Y

Note, Data coverage from diverse sources = 1: Hybrid feature usage (LSI+LSTM) = 2: Lexicon usage = 3: Word semantics coverage = 4: Deep learning mechanisms usage = 5: Pre-training of Language Model = 6: Graph Neural Network usage = 7: . Y = denotes 'yes'; N = denotes 'no'.

In the light of above discussion, our approaches address several limitations present in the existing studies by customizing the model for context, reducing dimensionality, providing better interpretability, and offering a more tailored solution. The above mentioned existing studies like [26], [25] etc. are based on data from popular social media platform like Twitter. This platform only may not fully represent the dynamics and language used on more region-specific platforms. Our approaches utilizes word embeddings from both LSI, BERT, and GNN which is potentially capturing more semantic and contextual information. Some of the existing studies suffer from high dimensionality due to the use of a large number of features or words, which can lead to computational challenges and reduce the model's interpretability. Our suggested approach addresses this issue by using LSI for topic modeling and GNN for feature extraction, potentially reducing dimensionality. Our proposed approaches uses the strengths of deep learning, which can provide a more robust and accurate solution. Some of the existing models may lack interpretability in their predictions. In our approach, especially the GNN model, can potentially offer better insights into why a certain prediction is made, which can be crucial for interventions and support.

Chapter 3

Methodology

We develop two pipelines in which we follow different techniques of machine learning and natural language processing for feature extraction (TF-IDF, word2vec, BERT), feature selection and model creation (LSI, LSTM, GNN) in the classification task. One is deep learning based model and another one is graph neural network model.

We have developed our project and made outline based on the working procedure in a sequential fashion. In the SI detection task, the methodology for first approach includes data pre-processing, LSI topic modeling, feature extraction, feature selection, data fed to LSTM model and output generation for SI detection. For the second approach, the methodology includes data pre-processing, feature extraction, feature selection, data fed to GNN model and output generation for SI detection. We are describing those steps for both models.

3.1 Dataset Collection and Annotation

3.1.1 Social Media Data Collection and Annotation

We collect a total of 785 posts where 386, 321 and 78 posts are from reddit, Facebook and Twitter, respectively. We scrawl and scrap data from those platforms. The collected data is annotated as 'YES' and 'NO' for suicidal and non suicidal labels, respectively by one behavioural scientist. 405 posts are annotated as 'YES' and 380 posts are annotated as 'NO'. We describe the complete data annotation guidelines in Section.

Table 3.1: Statistics of collected data.

Category	**Value**
Social media posts	Reddit=386, Twitter=78, Facebook=321)
Total number of labeled data	785
Demography	Bangladesh
Time period	1st January, 2020 to 13th March, 2022
Word statistics	Words=26,946, Sentences=1,183, Avg. Sentence (words)=35, Paragraphs=736, Keyword Density (Suicide=388 (6%), Bangladesh=164 (3%))

3.2 Data Pre-processing and Representation

We collected both qualitative and quantitative data from three social media sites (Facebook, Twitter and Reddit). However, long lists of data points can be difficult to interpret and it takes more time in data modeling. Data representation is needed to display and summarize data and it helps model to understand the meaning. Before making data in a model representable format, it needs to be pre-processed and cleaned. We are discussing about data pre-processing and different types of data representation methods that we adopted for both approaches in this chapter.

3.2.1 Pre-processing

We carefully clean the textual data before executing them into SI detection task since the data can be noisy. We pre-process the data for both approaches. Data pre-processing steps include removing irrelevant characters [28], stemming and lemmatization [28] and stop words removal [28] etc. Nonsensical characters are not recognizable to the machine learning models which make the text noisy. It must be deleted from text to ease the classification task. Emojis, URLs, punctuation, white space, numerals, and user references are deducted from the text using regular expressions [28]. We apply porter stemmer and wordnet lemmatizer of nltk to perform stemming and lemmatization to improve text categorization accuracy [28]. Unimportant and frequently occurring words which has little or no grammatical responsibility for text classification is identified as stop words. We use nltk stop words corpus to eliminate stop words to concentrate more on the relevant information [28].

3.2.2 Feature Extraction and Training

In our first approach, We have employed different feature engineering approaches for creating textual features such as TF-IDF[76], Word2Vec[75] and BERT [77, 78]. We have employed Latent Semantic Indexing to get contextual features and reduce dimension of

features.

In our second approach, we have employed Linguistic Inquiry and Word Count to create categorical features from the collected data. We are discussing about different types of data representation methods that we adopted for both approaches of classification in this chapter.

Term Frequency-Inverse Document Frequency (TF-IDF): We have rendered TF-IDF[76] to convert the data into numerical format and to find out important words that occur a few times with high weights. So, we get only the significant words to a record in a collection or corpus. We have used Scikit-learn, a free software machine learning library of Python to implement tf-idf. Scikit-learn has Tf-idf Vectorizer class that combines all the options of Count Vectorizer and Tf-idf Transformer. The numerical formatted data we get from the Tf-idf is further used in Latent Semantic Indexing as input data to generate topic from it and categorize the data based on topic.

Latent Semantic Indexing (LSI): Latent Semantic Indexing has been employed to create features from data and reduce the dimension of the data. Singular value decomposition (SVD)[81] which is a mathematical technique is used to identify patterns in the relationships between the terms and concepts contained in a corpus. SVD is also used to reduce the dimension of the data as it decomposes the term-document matrix into term-topic matrix and topic-document matrix. The final of LSI is a topic-document matrix which is a reduced dimension of matrix.

Word to Vector (word2vec): Word to Vector[75] is a word embedding method. The basic idea of Word to Vector is to find out words that occur in similar context tend to be closer to each other in vector space. We have used pre-trained Word2Vec model from google and trimmed the word list to 50,0000 compared to 300,000 in the original Google pre-trained model of 'GoogleNews-vectors-negative300.bin.gz'. For implementing word to vector in Python, we have used modules such as nltk and gensim. We have developed Skip Gram architecture in our implementation. Skip Gram model predicts the context words from given current word within a specific window in which the input layer stores the current word and the output layer stores the context words. The hidden layer stores the number of dimensions in which the current word has to represented at the output layer. Output of Word to Vector indicates the cosine similarities between word vectors. The Tf-idf output data is passed through Latent Semantic Indexing model to categorize the data based on topic. The collected and pre-processed data is then concatenated with its corresponding topic and used again in word to vector model to perform word embedding.

Bidirectional Encoder Representations from Transformers (BERT): We choosed BERT [77, 78] for the word embedding too. It offers an advantage over Word2Vec as it produces word representations that are dynamically informed by the words around them. But each word has a fixed representation under Word2Vec regardless of the context. BERT ensures polysemy through the context-informed word embedding. BERT has the best use by capturing other forms of information for accurate feature representations. Transformers in BERT provides a number of classes for supporting different tasks like token classifica-

tion, text classification etc. We used Bert Model with no specific output task. BERT is employed just to extract word embedding from corpus.

Table 3.2: LIWC categories based on suicidal ideation detection.

LIWC Category	**Relevance to Collected Data**
Word Count	WC (Word Count)
Syntax	WPS (Words per Sentence), Qmarks (Question Marks)
Lexical Diversity	Unique, Dic, Sixltr (Six-Letter Words)
Pronouns	pronoun, ppron (Personal Pronouns), i, we, you, shehe, they, ipron (Impersonal Pronouns)
Articles and Verbs	article, verb, auxverb (Auxiliary Verbs), past, present, future
Adverbs and Prepositions	adverb, preps (Prepositions), conj (Conjunctions), negate (Negations), quant (Quantifiers), number
Swear Words	swear (Swear Words)
Social Words	social, family, friend, humans
Emotions	affect, posemo (Positive Emotion Words), negemo (Negative Emotion Words), anx (Anxiety Words), anger, sad
Cognitive Mechanisms	cogmech (Cognitive Mechanism Words), insight, cause, discrep (Words Referring to Discrepancies), tentat (Tentative Words), certain, inhib (Inhibitory Words)
Inclusivity	incl (Inclusive Words), excl (Exclusive Words)
Perception	percept (Perceptual Words), see, hear, feel
Biological Terms	bio, body, health, sexual, ingest (Words Related to Ingestion and Consumption)
Relativity	relativ (Words Related to Relativity), motion, space, time
Work and Employment	work, achieve, leisure, home, money
Religion and Spirituality	relig (Words Related to Religion and Spirituality), death, assent (Words Indicating Agreement or Affirmation), nonfl (Non-Fluencies Indicating Speech Disfluencies)

Linguistic Inquiry and Word Count (LIWC): To extract features from the pre-processed data, we have used Linguistic Inquiry and Word Count (LIWC)[82] in the second approach of methodologies. We have used LIWCLite7.EXE software which performs linguistic analysis on the input text. This analysis involves breaking down the text into its constituent elements, such as words, phrases, and sentences. It identifies linguistic patterns, structures, and features within the text. LIWCLite7.EXE uses a pre-defined dictionary, often referred to as the LIWC dictionary. This dictionary contains a list of words and phrases that are associated with specific linguistic and psychological categories.

Each word or phrase in the dictionary is tagged with one or more categories, such as emotions, cognitive processes, social processes, and more. LIWC uses 69 features/categories and calculates the value of word proportion among those categories. In the context of our collected data, we got the proportion value of our 528 data as a CSV format. As a result, we got output of calculated value from LIWC.

3.3 Feature Selection

For deep learning based suicidal text classification task, we have selected all of the features that we got from the output of word embedding models such as word to vector, TF-IDF, latent semantic indexing and Bert. In the second approach of suicidal text classification task where we have used graph neural network, the output data from LIWC underwent feature ranking and statistical tests to extract the best features to create graph and to use for modeling in the graph neural network. We have used Analysis of Variance (Anova), Boruta, and Fisher's score to rank and select the relevant features to find out the significance of features. In this chapter, we are discussing in detail each one of them.

3.3.1 Pipeline 1: Deep learning based semantic topic modeling approach

The final feature vectors are the concatenated features of size 1,205,760 from Bert and word2vec where Bert is used and only word2vec features of size 602,880 where Bert is not used. The concatenated feature vectors are illustrated on table 3.3.

Table 3.3: Feature vectors of deep neural network based suicidal text classification model with concatenated Bert and word2vec features

	col_0	**col_1**	**col_2**	**col_3**	.....	**col_767**
0	-0.215415	0.369104	0.143434	-0.152095	.....	0.00000
1	-0.251024	0.404868	0.133691	-0.226219	.....	0.00000
2	-0.228476	0.362848	0.118640	-0.221778	.....	0.00000
.....	.....	.....	.....	.....	.....	.....
1569	-0.897630	-0.413052	-0.891758	0.724002	.....	0.927102

The suicidal text classification model where only word2vec features are used in a deep neural network is illustrated on table 3.4.

Table 3.4: Feature vectors of deep neural network based suicidal text classification model with word2vec features

	col_0	**col_1**	**col_2**	**col_3**	.....	**col_767**
0	-0.215415	0.369104	0.143434	-0.152095	.....	0.00000
1	-0.251024	0.404868	0.133691	-0.226219	.....	0.00000
2	-0.228476	0.362848	0.118640	-0.221778	.....	0.00000
.....	.....	.....	.....	.....	.....	.....
784	-0.578900	-0.423022	-0.807458	0.953012	.....	0.988421

3.3.2 Pipeline 2: Psycho-linguistic lexical feature based graph neural network driven approach

For graph neural network based suicidal text classification task, we have taken 72 LIWC features at first. Then, we applied different feature selection methods (ANOVA, Boruta, Fisher Score) for selecting significant features. The LIWC feature vectors of size 38,088 are shown on table 3.5.

Table 3.5: Feature vectors of graph neural network based suicidal text classification model

	ID	**WC**	**WPS**	**Qmarks**	.....	**Class**
0	1	144	24.00	0.00	.....	1
1	2	121	20.17	0.00	.....	1
2	3	125	15.62	0.00	.....	1
.....	.....	.....	.....	.....	.....	.....
528	529	7	7.00	0.00	.....	0

Analysis of Variance (ANOVA): Analysis of variance (ANOVA) is a statistical technique used to check if the means of two or more groups are significantly different from each other [84]. We have used pandas library and scipy.stats module to perform ANOVA on the data. For that, we took the LIWC output features and read the data to select relevant features. We started by creating a list D of dictionaries and a list column based on columns in a DataFrame df. The iteration goes through the columns of df except columns 'ID' and 'Class' and creates dictionaries in D. The dictionary contains group-specific data from the columns of df (excluding 'ID' and 'Class'). The length of the D list will be equal to the number of columns in df that meet the condition. The calculation of analysis of variance (ANOVA) scores for each pair of groups in the list D is performed and results are stored in a dictionary called anova_score. Significant $p - values$ are filtered from a DataFrame named Anova and create a new DataFrame P with those significant

$p - values$. If the value is less than or equal to 0.05, it updates the dictionary P with the corresponding value. Also, the values are appended to a list L. From initial 69 features, we got 15 features finally. In table 6.3, the selected features of ANOVA along with their correlation (Pearson) value is shown. Theb selected features are shown on table 3.6

Table 3.6: Correlation scores for ANOVA-selected features.

Selected Features	**Correlation (Pearson)**
WC	0.069501
WPS	-0.067655
Qmarks	-0.008608
Unique	0.049743
Dic	0.339795
...	...
relig	0.020536
death	0.343349
assent	0.068330
nonfl	-0.037389
filler	0.103920

Boruta: Boruta works on the "all-relevant" principle as it provides you with ALL the features that are relevant to your Machine Learning problem[85]. To perform Boruta, we use boruta package. For that, we took the LIWC output in a dataframe to select relevant features. We have defined the feature matrix X by dropping the 'ID' and 'Class' columns from the DataFrame. We also defined the target variable y using the 'Class' column and created a random forest classifier. The we created the Boruta feature selector and fit the Boruta selector to our data. We got total 9 features out of initial 69 features after performing Boruta. The selected features are shown on table 3.7.

Table 3.7: Correlation scores for Boruta-selected features.

Selected Features	**Correlation (Pearson)**
WC	0.069501
WPS	-0.067655
Dic	0.339795
funct	0.128490
preps	0.167909
affect	0.215996
negemo	0.208045
cogmech	0.146448
death	0.343349

Fisher's score: For each feature/variable, it computes Fisher score, a ratio of between-class variance to within-class variance. The algorithm selects variables with largest Fisher scores and returns an indicator projection matrix. We start by specifying the number of features to select and create an instance of SelectKBest with the Fisher score function (f_classif). After that, the feature matrix is fitted and transformed. The top 20 features are selected using the Fisher score (f_classif) as the scoring function. The SelectKBest transformer is used to the feature matrix X and target variable y to select the top 20 features and print the indices of the selected features. Table 3.8captures the selected features using Fisher's Score feature selection method and their corresponding correlational value calculated with Pearson correlation method.

Table 3.8: Correlation scores for Fisher's Score-selected features.

Selected Features	**Correlation (Pearson)**
WC	0.035354
WPS	0.083854
Qmarks	0.0751975
Unique	0.054630
Dic	0.080996
Sixltr	-0.045805
funct	0.093868
pronoun	-0.047607
ppron	0.131946
i	0.080134
we	-0.016379
you	0.084493
shehe	-0.020033
they	0.103252
ipron	-0.016707
article	0.020536
verb	0.343349
auxverb	0.068330
past	-0.037389
present	0.103920

3.4 Model Building

Model building is a crucial part in any machine learning based research work. We employed machine learning models to find out semantic relationship of words in text, topic wise text categorization, convert words to vector format and suicidal text classification. We implemented all of the models on google colaboratory platform. As we developed two different pipeline in our suicidal text classification task, we are describing step wise model building approach for the pipelines separately. The proposed method is implemented to detect suicidal ideation based on a DNN classifier. We have integrated an LSI to generate topics and label data with topics. For the implementation, Python 3.8.8, is installed with an Intel(R) i5-3330S processor running at 2.70GHz, 8GB of memory (RAM). A 64-bit operating system is used to evaluate the designed model.

From data collection to suicidal text classification in a model, we divided the model building procedure into layers. We selected the layered approach in model building which can enhance modularity, understandability, flexibility, and collaboration. Efficient exper-

imentation, interpretation, and maintenance can be easier which can ultimately improve the model's robustness and adaptability to different scenarios and datasets. Breaking down the model into layers provides a modular and abstract structure and modularity allows to easily swap or update components within a layer without affecting the entire model. This is important because different techniques like TF-IDF, BERT, and LSI may require different pre-processing or data formats. Separating them into layers simplifies maintenance and upgrades. By providing a detailed description of each layer, the documentation process can be easier for other researchers or developers to comprehend, replicate, or build upon the existing work. Different techniques have varying strengths and may work differently on specific datasets. By keeping the layers separate, we can easily experiment with new algorithms or improvements within a specific layer without affecting the others. Moreover, when any issue is encountered or need to understand why our model made a specific prediction, separated layer helps to pinpoint where the problem might lie. We can trace back issues to a particular layer or component, which simplifies debugging and interpretation. Figure 3.1 and 3.2 describe the flow diagram of pipeline 1 and pipeline 2 respectively.

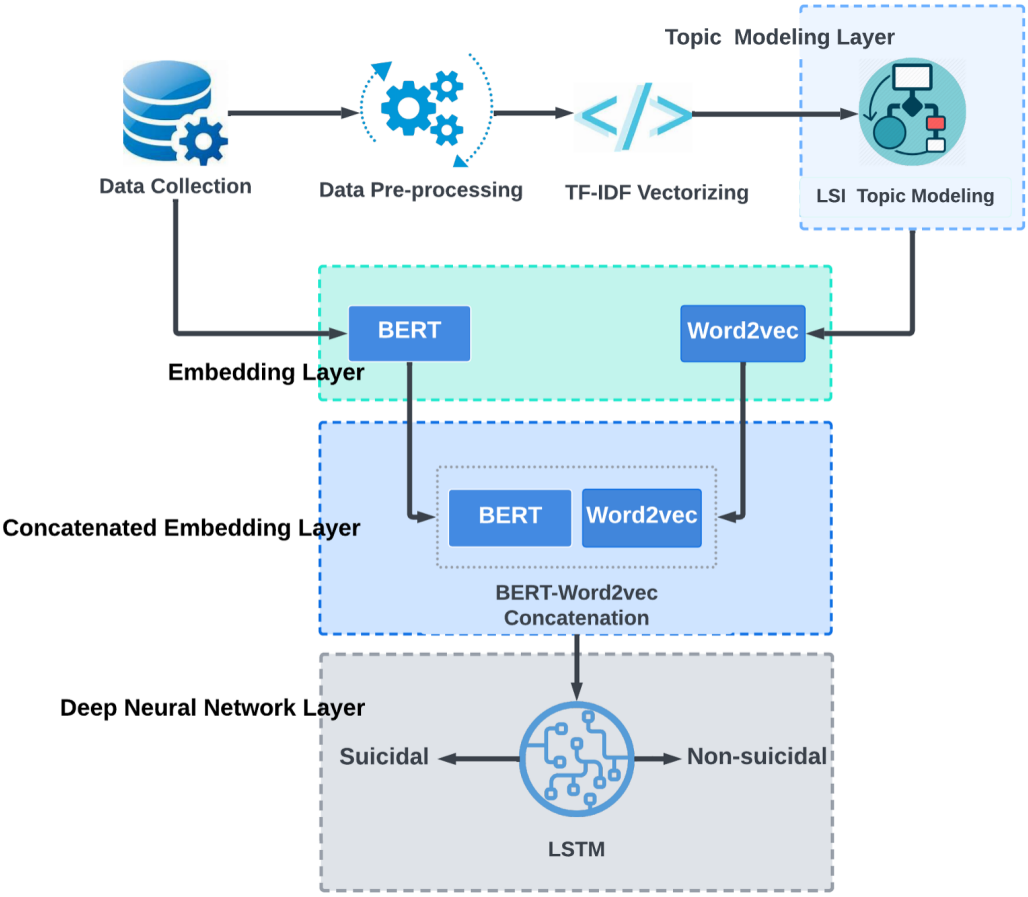

Figure 3.1: Pipeline 1: Deep learning based SI detection with semantic topic modeling

3.4.1 Pipeline 1: Deep learning based semantic topic modeling approach

System Model: We incorporate the power of word semantics (LSI) [21] and preserving long text (LSTM) [17] and produce an integrated LSI-LSTM model for SI detection. We employ TF-IDF [14] for converting the text data into vectors. Before performing LSI, it is important to ensure term document matrix to be filled with important words by TF-IDF vectors [29]. The TF-IDF vectors are passed through LSI for topic modeling. The output from LSI are embedded with word2vec [22]. The original text data are embedded with BERT [23] embedding. The concatenated embeddings from word2vec and BERT enter into the LSTM model as input for SI detection. By incorporating BERT, we include the power of contextual word embedding through the pre-trained language model.
Architecture: We divide the pipeline 1 into layers which can enhance modularity, understandability and flexibility. The layers for pipeline 1 are topic modeling layer, embedding layer, concatenated embedding layer and deep neural network layer.

a) Topic modeling layer

The collected and pre-processed data retrieved from social media are passed through TF-IDF vectorizer. TF-IDF ensures retrieval of less frequent but significant features [14]. TF-IDF transforms the text into meaningful representation of integers or numbers [14]. This transformed data is used to fit to machine learning algorithm for predictions. TF-IDF transforms a word by comparing the number of times a word appears in a document of words with the number of documents the word appears in [14].

We import tf-idf [1] vectorizer from scikit-learn [2], a free machine learning library. we initialise the vectorizer and then call fit and transform functions over it to calculate the tf-idf score for the text. The tf-idf vectors converted from pre-processed data is used as input in latent semantic indexing model to generate topics and categorize the input data based on the topics. We consider LSI as it finds semantic similarities between texts and outperforms in dimensional reduction using singular value decomposition (SVD) mechanism of mathematics [21]. LSI understands and classifies words with similar contextual meanings within large data sets [21]. LSI works using a mathematical operation named as SVD [21] as a partial application. SVD [21] reduces a matrix into its constituent parts for simple and efficient calculations. The pre-processed words of a string are placed into a Term Document Matrix (TDM). A TDM is a 2D grid that counts the frequency of each specific word (or term) occurs in the data set of documents which is decomposed into Term Topic Matrix (TTM) and TDM [21]. Weighing functions are used in the TDM. Classification of the documents is performed that contain the word with a value of 1 and with a value of 0. Words can be occurred with the same general frequency in these documents which is called word co-occurrence [21]. LSI uses SVD which produces vectors by predicting the meaning more accurately than analysing the individual terms [21].

[1] https://scikit-learn.org/stable/modules/generated/sklearn.feature$_e xtraction.text.TfidfVectorizer.html$
[2] https://scikit-learn.org/

We used gensim[3] python library for topic modeling, document indexing, and similarity retrieval performance. We import corpora from gensim to get collections of texts used for topic modeling and "Coherence Model" class for computing coherence scores for topic models.

b) Embedding layer

The original data is labeled with corresponding topic generated by LSI. Word2vec is used to embed the topic labeled data. The original data is embedded with BERT. BERT [23] ensures contextualized features. We use BERT for context based pre-trained embedding. We use pandas [4], numpy [5] and torch [6] packages and pre-trained BERT model and tokenizer to implement BERT. After that, we convert the tokenized BERT input to the tensor format to create contextual embedding for a set of contexts. Word2vec and BERT embedding is grouped into the embedding layer.

i) Word to Vector: The topic categorized text are further converted into vectors using word2vec. Word2vec word embedding is used to measure syntactic and semantic similarities between words. There are two layers in a word to vector model [22]. The word embedding layer and dense layer [22]. The word embedding layer is a matrix which is defined with number of unique words in the corpus and word embedding size [22]. Each row of the matrix represent a word in the corpus. "word embedding size" is a hyper-parameter to be decided and can be thought as how many features that we would like to use to represent each word. The latter part of the model is simply a logistic regression in a neural network form [22]. We use pre-trained word2vec model from google but trim the word list to 3,00,000 compared to 5,00,000 in the original Google pre-trained model. We use gensim [7] open-source Python library for representing text as semantic vectors and Tokenizer" class of keras to convert words to integers. texts_to_sequences" method helps to convert tokens of text corpus into a sequence of integers. Maximum length of tweet is considered as maximum length of vector. "pad_sequences" pads sequences to the same length considering the maximum length.

ii) BERT: BERT considers all the words of the input sentence simultaneously [23]. BERT uses attention mechanism to develop contextual meaning of the given words [23]. BERT architecture evolved from transformers. There are several stacked encoder cells inside it just like transformers but only the encoder part of the transformer is used in the case of BERT. BERT has self-attention heads and a feed-forward neural network just like the encoder cells in transformer. BERT has two architectures

[3]https://radimrehurek.com/gensim/
[4]https://pandas.pydata.org/pandas-docs/stable/getting$_started/overview.html$
[5]https://numpy.org/
[6]https://pypi.org/project/torch/
[7]https://radimrehurek.com/gensim/

known as BERT base and BERT large. Both architectures take an input of 512-dimensions. BERT base has 12 layers (transformer blocks), 12 attention heads, 110 million parameters, and has an output size of 768-dimensions. BERT was pre-trained on a large corpus of unlabelled text along with the entire Wikipedia (2,500 million words) and book corpus (800 million words). It has two training methods. The methods are Masked Language Model (MLM) Next Sentence Prediction (NSP) [23]. We use BERT-base[23] in our model as we are designing the model for real-time intervention and BERT-base has faster data processing time to cope up with the environment. We start by installing TensorFlow [8] which provides a rich collection of libraries to perform the pre-processing regularly required by text based models, and includes other features useful for sequence modeling. "hub.KerasLayer" is used to load "bert pre-process" and "bert encoder" module from the local directory to use it offline. We define "get_sentence_embeding" function which returns encoded text from the pre-processed text. For 529 data set the "embedded shape" becomes 768.

c) Concatenated embedding layer

Two embedding is concatenated together to set as input data in deep neural network. Topic labeled text is embedded using word to vector and initial pre-processed data is embedded using BERT. Topic labeled text ensures semantic information gaining [33] where BERT ensures contextual information gaining and shows impact on large scale training dataset [34]. As we are going to use this model for real-time SI detection, BERT can make it's way to retrieve contextual information from large scale real-time data from social media. The BERT and word2vec embedding is concatenated together to get the combined embeddings for both semantic and contextual information retrieval [35].

For the concatenation of the embeddings, we start by importing the NumPy [9] library, which is commonly used for numerical operations and array manipulation. "wor2vec_embeddings" is a numpy array representing word2vec embeddings with a shape of (n_samples, 100). Here, "n_samples" represents the number of samples or data points, and each embedding vector has a dimension of 100.

"bert_embed" is numpy array representing BERT embeddings with a shape of (n_samples, 768). Similar to word2vec, "n_samples" represents the number of samples, and each BERT embedding vector has a dimension of 768. We extend the word2vec embeddings to match the same dimensionality (768) as of the BERT embeddings. The "np.pad" function is then used to add these zero columns to the word2vec embeddings. The padding is applied along the second axis (axis 1), effectively increasing the dimensionality of each word2vec embedding from 100 to 768. The shape of "wor2vec_embeddings_extended" becomes (n_samples, 768). Finally, "np.concatenate" is used to concatenate the extended word2vec embeddings and the BERT embeddings along the first axis (axis 0). This results in the "merged_embed" variable, which is a combined array of embeddings with

[8]https://www.tensorflow.org/
[9]https://numpy.org/

a shape of (2 * n_samples, 768). The variable stacks the word2vec embeddings on top of the BERT embeddings which is essentially merging the two sets of embeddings. The numpy array "merged_embed" contains both the extended word2vec embeddings and the original BERT embeddings from BERT with matching dimensions for further analysis or processing.

d) Deep neural network layer

We use LSTM [17] in deep neural network Layer to solve the vanishing gradient problem in back propagation. LSTM has feedback connections which makes it different from the feed forward neural network. This feedback connections enable LSTM to process the entire sequences of data (e.g. time series). Thus, LSTM is handy to process sequential data such as text, speech and general time-series [17]. LSTM consists of four neural networks and different cells known as memory blocks. Information is stored in the cells and the memory is manipulated in the gates [17].

We import "Sequential" from "keras.models" and "Dense", "LSTM", "Embedding" from "keras.layers". l"lstm_out" is set to 80 which contains information about the entire sequence and transforms the vector sequence into a single vector of size 80. We use a Sequential model to build it layer by layer. We start adding with an embedding layer. Next, we add an LSTM layer with 80 units. A dense output layer with a sigmoid activation function is added which outputs the probability that a given text is suicidal. We split the data into 80:10, where 80% of the data is used for training and 10% of the data is used for testing. After defining the model, we compile it using the binary cross entropy loss function, the Adam optimizer, and accuracy as our metric. We train our model with a batch size of 32 and train for 6 epochs. After the training, we can evaluate our model for the test data and we get 83% accuracy with a loss of 17% for combining the features from word2vec and BERT.

We train our DL based model for classification. Fine-tuning with 10-fold cross-validation is performed. We apply a pre-trained word2vec model trained on 100 billion words from Google News for features classification. The neural network models are initialized with a 300-dimensional pre-trained word2vec [30, 52]. Table 3.9 presents the parameter setting for the DL based model. The experiment is conducted using different parameters such as embedding dimensions, hidden units of LSTM, dense layer, total parameter, trainable parameter, non-trainable parameter, activation function, batch size, loss, optimizer and epochs.

Table 3.9: Parameter table of pipeline 1: Deep learning based SI detection with semantic topic modeling

Hyper-parameter	**Value**
Embedding Dim	768
Hidden Units of LSTM	128
Dense	1
Total parameter	213313 (833.25 KB)
Trainable parameter	213313 (833.25 KB)
The activation function of LSTM	Sigmoid
Batch Size	32
Loss	Binary crossentropy
Optimizer	Adam'
Epochs	10

3.4.2 Pipeline 2: Psycholinguistic lexical feature based graph neural network driven approach

System model: We incorporate the power of lexical features and cutting edge technique to our pipeline 2 for constructing a lexical psycholinguistic knowledge-guided graph neural network [36] based model for SI detection. We employ LIWC [15] to extract psycholinguistic features from the collected and pre-processed text data. The LIWC features are used to create graph using k-nearest neighbour [37]. Later, we apply graph neural network [20] on the graph for SI detection.

Architecture: In pipeline 2, the layers are LIWC feature creation and selection layer, graph creation layer and graph neural network layer.

a) LIWC feature creation and selection layer

LIWC-15 [15] is used to perform LIWC analysis to extract psycho-linguistic features. LIWC is a lexicon used in Natural Language Processing (NLP) to extract features from text [15]. LIWC includes a dictionary with more than 90 categories which covers linguistic, psychological, and emotional dimensions. These categories include linguistic elements like pronouns, articles, and prepositions, as well as psychological categories such as emotions, cognitive processes, and social dimensions [15]. We use "Home" icon labeled "LIWC Analysis" to get started with. We store the data set in CSV format to speed up the process. We use the newer LIWC-15 dictionary format (.dicx).The Categories tab lists all of the categories which are contained within the selected dictionary. The "segmentation" tab is selected to divide texts into segments. By clicking on the "Run LIWC" button, LIWC analysis is performed to get features based on categories and save the data in a CSV format. The collected LIWC features are then used for feature ranking and selection. We get total of 69 features. We use three feature selection techniques (ANOVA [38], Boruta

[39] and Fisher Score [40]) to rank and select features. We get a total of 32 features in ANOVA, 9 features in Boruta and 20 features in fisher score. The features are normalized using the "Min-Max" scalar of "sklearn" so that all features can be transformed into the range 0 to 1. We use "fit_transform" to compute mean and standard deviation and then use it to scale the train dataset.

b) Graph creation layer

We start building k-nearest neighbour graph [37] by using a Python package networkX [10]. In kneighbors_graph function, we set 2 for the parameter k to find 2 nearest neighbour of each data point by calculating distance using cosine similarity. With the selected k-nearest neighbors, the k-NN graph is constructed by connecting each data point to its k-nearest neighbors and edge weight is assigned based on the calculation of cosine similarity. For 529 number of data entry, we get 529 nodes and 885 edges in the undirected graph.

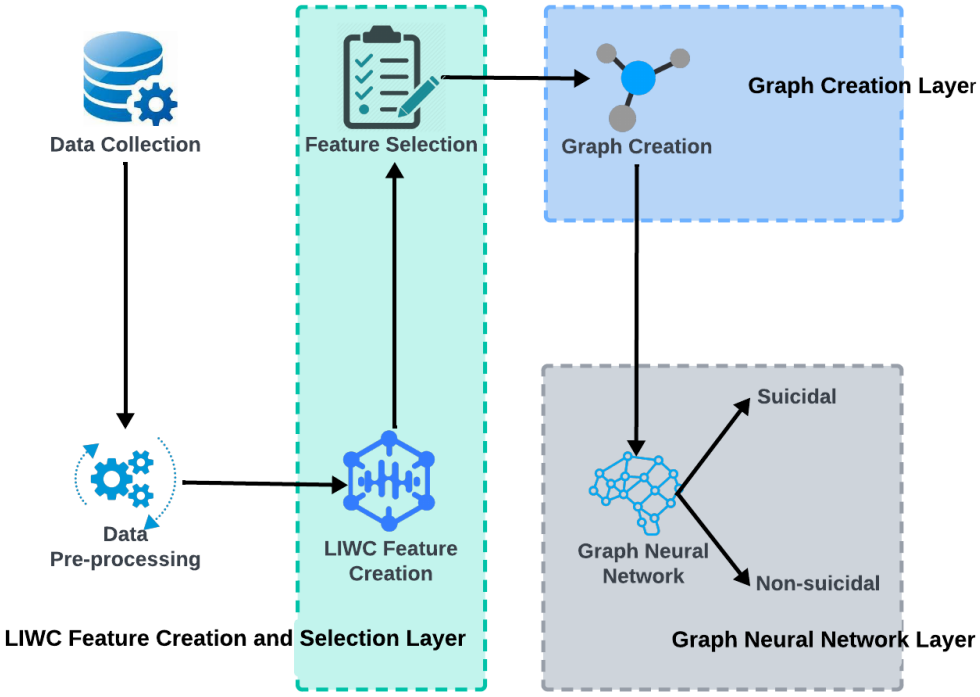

Figure 3.2: Pipeline 2: Psycho-linguistic lexicon based Graph neural network driven SI detection

c) Graph neural network layer

For implementing graph neural network [20], we use PyTorch [11] which is available through torch.nn module. we define the classification neural network. We declare Softmax() function as activation function for our last layer. The last layer has 2 output units as it is

[10] https://networkx.org/
[11] https://pytorch.org/

a binary classification task. There are two probabilities present per sample of the data and probabilities for a single sample will sum up to 1 as the output of the last layer is from softmax. We use two defined method "___init___" and "forward". In "___init___", we define two layers through which three channels are passed. The channels are input channel, hidden channel and output channel. We use adam optimizer to optimize the model. Before casting the model parameters, we cast inputs to double. The input, hidden and output parameters for two graph convolution layer is 71, 64 and 530. We train the model for 200 epochs to train for our data. We get 98% of accuracy for the test data.

Table 3.10 presents the parameter setting for the GNN based model. The experiment is conducted using different parameters such as number of parameter in Linear1 Layer, number of parameter in Sum Aggregation2 Layer, number of parameter in GCN Conv3 Layer, number of parameter in Linear4 Layer, number of parameter in Sum Aggregation5 Layer, number of parameter in GCN Conv6 Layer, activation function, batch size, loss, optimizer and epochs. We used Python with the NLTK natural language toolkit.

Table 3.10: Parameter table of pipeline 2: Psycho-linguistic lexicon based graph neural network driven SI detection

Hyper-parameter	**Value**
# of parameter in Linear1 Layer	4,544
# of parameter in GCN Conv3 Layer	64
# of parameter in Linear4 Layer	128
# of parameter in GCN Conv6 Layer	2
Total parameter	4,738
Trainable parameter	4,672
Non-trainable parameter	66
Input size	0.38MB
Parameter size	0.02 MB
Estimated Total Size	0.40 MB

Chapter 4

Result and Analysis

Model performance can be evaluated in various ways. We have considered accuracy, precision, recall, and F-measure as performance metrics. In this chapter, we will discuss each of them in detail to precisely evaluate the performance of our suggested model.

4.1 Performance Metrics

As mentioned above, we are considering accuracy, precision, recall, and F-measure as performance metrics. Before discussing the evaluation of the metrics, let's define each metric in the context of our work. We will also discuss terms such as True Positive (TP), True Negative (TN) , False Positive (FP), and False Negative (FN) [97]. True Positive (TP) is the number of data points from the test set that are correctly classified (positive/negative) by the classifier as belonging to a particular class [97]. True Negative (TN) is the number of data points from the test set that are correctly classified (positive/negative) by the classifier as not belonging to a particular class [97]. False Positive (FP) is the number of data points from the test set that are incorrectly classified (positive/negative) by the classifier as belonging to a particular class [97]. False Negative (FN) is the number of data points from the test set that are incorrectly classified (positive/negative) by the classifier as belonging to a particular class but should not have been [97]. Precision (P) is the number of true positives divided by the total number of positive predictions [97].

$$\text{Precision } (P) = \frac{TP}{TP + FP} \tag{4.1}$$

Recall (R) is the number of correct positive predictions made out of all positive predictions that could have been made [97].

$$\text{Recall } (R) = \frac{TP}{TP + FN} \tag{4.2}$$

F-measure is the weighted harmonic mean of precision and recall for a particular class [97].

$$\text{F-measure } (F1 - score) = \frac{2 \cdot P \cdot R}{P + R} \tag{4.3}$$

Accuracy is the number of true positives and true negatives divided by the number of true positives, true negatives, false positives, and false negatives [97].

$$\text{Accuracy } (Acc) = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.4}$$

4.2 Pipeline 1: Deep learning based semantic topic modeling approach's performance evaluation

Combined BERT and Word2vec Feature with LSTM: The accuracy, precision, recall and F-measure calculated from this classification is as follows :

Table 4.1: Classification report for a model combining BERT and word2Vec features with LSTM

Metric	**0 (Non-suicidal)**	**1 (Suicidal)**	**Accuracy**
Precision	0.88	0.74	0.87
Recall	0.86	0.77	0.87
F1-score	0.87	0.75	0.87
Support	47	22	79
Macro Avg.	0.81	0.81	0.81
Weighted Avg.	0.87	0.87	0.87

The precision is 0.88 for class 0, so it can be said that 88% of the instances predicted as class 0 are correct. The precision is 0.74 for class 1, indicating that 74% of the instances predicted as class 1 were correct. For class 0, the recall is 0.86, meaning 86% of the actual class 0 instances are correctly predicted. For class 1, the recall is 0.77, indicating that 77% of the actual class 1 instances are correctly predicted. For class 0, the F1-score is 0.87, and for class 1, the F1-score is 0.75, where the F1-score is the harmonic mean of precision and recall. The support for class 0 is 50, meaning there are 50 instances of class 0 in the test dataset. For class 1, the support is 26, and there are 26 instances of class 1 in the test dataset. The accuracy is 0.83, indicating that 87% of the test instances are classified correctly. As the macro-average is the unweighted average of precision, recall, and F1-score across both classes, it provides equal weight to both classes. The macro-average in this model for F1-score is 0.81. In this model, the weighted average F1-score is 0.87, so we can say that the weighted average of precision, recall, and F1-score, where each class's score is weighted by its support, is 0.87.

Overall, this classification model performs well in terms of F1-score for both classes. For class 1 (recall: 0.77), it has higher recall than class 0 (recall: 0.86). The accuracy of 0.83 indicates that the model correctly classifies 83% of the test instances. However, the precision for class 1 is relatively lower (0.74), indicating a moderate number of false positives for class 1 predictions.

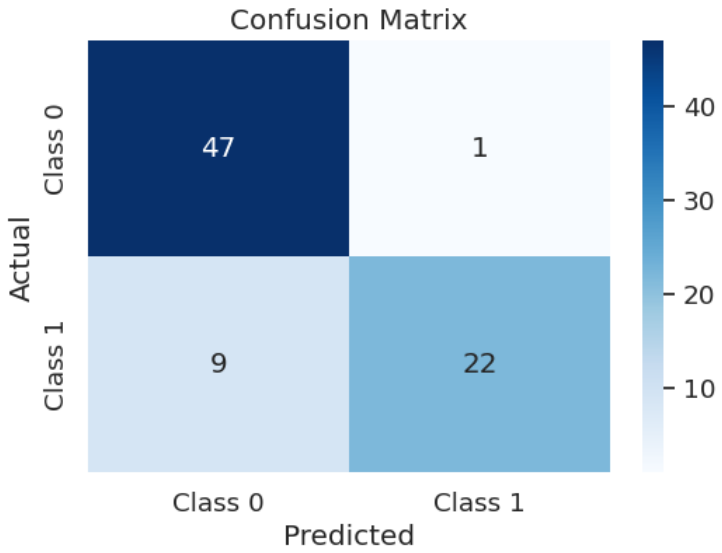

Figure 4.1: Confusion matrix of joined BERT and word2vec feature with LSTM classification performance

Figure 4.1 illustrates the confusion matrix of the deep neural network based suicidal sentiment classification model where joined BERT and Word2vec feature with LSTM is used for classification. The confusion matrix is used to measure the effectiveness of a classification model. It is used to classify correct and incorrect predictions. Confusion matrix provides a matrix layout for different outcomes of the predicted results and helps to visualize the outcomes. It plots a tabular form of all classifier's predicted and actual values. As we know, a good method has high true positive (TP) and true negative (TN) rates and low false positive (FP) and false negative (FN) rates. The column represents the predicted label, and the row represents the true label of the target variable as positive and negative. The confusion metric analyses the actual and predicted data from a given dataset. The prediction of data for 0 and 1 classes are 47 and 22. The total data used for testing is 79 in that 69 are predicted according to the actual class, and the rest of the 10 are predicted wrongly.

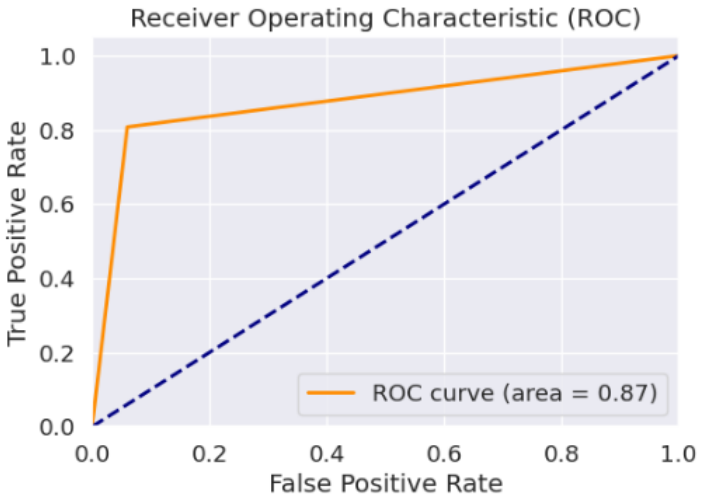

Figure 4.2: ROC of joined BERT and word2vec feature with LSTM classification performance

Figure 4.2 depicts the Receiver Operating Characteristic Curve (ROC) for the joined BERT and Word2vec feature with LSTM suicidal ideation prediction model. ROC (receiver operating characteristic curve) graph represents the classification model's performance across all thresholds. AUC stands for Area under the ROC Curve, which measures the total two-dimensional Area beneath the entire ROC curve. Two variables, such as the true and false positive rates, are used to plot the curve. For the best classifier, the range under ROC and AUC should be close to 1.

Combined BERT-Bangla BERT and Word2vec Feature with LSTM: The accuracy, precision, recall and F-measure calculated from this classification is as follows :

Table 4.2: Classification report for a model using joined BERT-Bangla BERT and word2Vec features with LSTM

Metric	**0 (Non-suicidal)**	**1 (Suicidal)**	**Accuracy**
Precision	0.76	0.53	0.68
Recall	0.76	0.53	0.68
F1-score	0.76	0.53	0.68
Support	37	19	56
Macro Avg.	0.64	0.64	0.64
Weighted Avg.	0.68	0.68	0.68

The precision for class 0 (Non-suicidal) is 0.76, suggesting that 76% of the instances predicted as non-suicidal are correct. Similarly, for class 1 (Suicidal), the precision is 0.53, indicating that 53% of the instances predicted as suicidal are correct. Precision is

a measure of the model's accuracy in predicting positive instances. The recall for both classes is 0.76, meaning that 76% of the actual non-suicidal instances and 76% of the actual suicidal instances are correctly predicted by the model. Recall, also known as sensitivity, quantifies the model's ability to capture positive instances. The F1-score, which is the harmonic mean of precision and recall, is 0.76 for class 0 and 0.53 for class 1. The F1-score provides a balanced measure of a model's overall performance by considering both false positives and false negatives. The support column indicates the number of instances in the test dataset for each class. There are 37 instances of non-suicidal (class 0) and 19 instances of suicidal (class 1) posts. The macro-average, which is the unweighted average of precision, recall, and F1-score across both classes, is 0.64. This metric treats both classes equally and provides an overall performance evaluation. The weighted average considers the imbalance in class distribution. In this case, the weighted average precision, recall, and F1-score are all 0.68, indicating a balanced performance considering the class distribution in the test dataset. The confusion matrix of this model is depicted below:

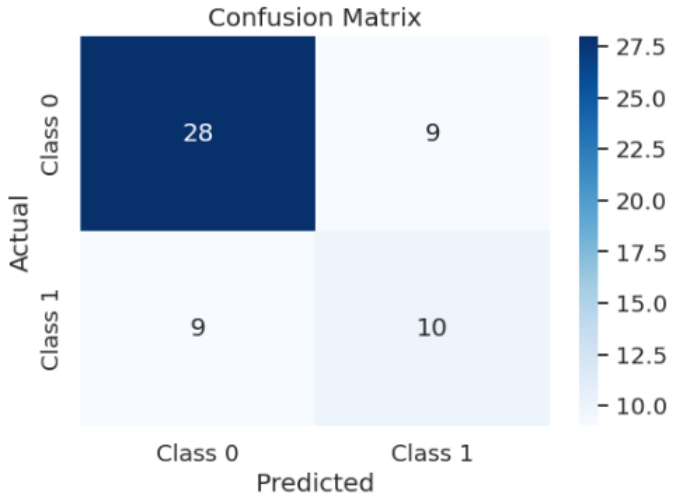

Figure 4.3: Confusion matrix of joined BERT-Bangla BERT and word2vec feature with LSTM classification performance

This confusion matrix interprets that the model correctly predicted 28 instances of non-suicidal posts and 10 instances of suicidal posts. There were 9 instances where non-suicidal posts were wrongly predicted as suicidal. Additionally, 9 instances of suicidal posts were incorrectly predicted as non-suicidal.

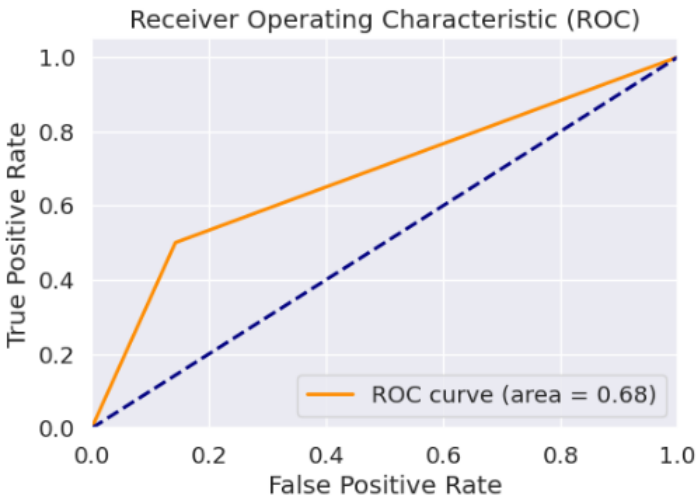

Figure 4.4: ROC of joined BERT-Bangla BERT and word2vec feature with LSTM classification performance

The ROC curve with an area of 0.68 indicates a reasonable ability of the model to distinguish between suicidal and non-suicidal posts.

Word2vec Feature with LSTM The accuracy, precision, recall and F-measure calculated from this classification is as follows :

Table 4.3: Classification report for a model using word2Vec features with LSTM

Metric	**0 (Non-suicidal)**	**1 (Suicidal)**	**Accuracy**
Precision	0.90	0.88	0.87
Recall	0.94	0.81	0.87
F1-score	0.92	0.84	0.87
Support	43	25	79
Macro Avg.	0.87	0.87	0.88
Weighted Avg.	0.87	0.87	0.87

The precision is 0.90 for class 0, so it can be said that 90% of the instances predicted as class 0 are correct. The precision is 0.88 for class 1, indicating that 88% of the instances predicted as class 1 were correct. For class 0, the recall is 0.94, meaning 94% of the actual class 0 instances are correctly predicted. For class 1, the recall is 0.81, indicating that 81% of the actual class 1 instances are correctly predicted. For class 0, the F1-score is 0.92, and for class 1, the F1-score is 0.84, where F1-score is the harmonic mean of precision and recall. The support for class 0 is 43, meaning there are 43 instances of class 0 in the test dataset. For class 1, the support is 25, and there are 25 instances of class 1 in the test dataset. The accuracy is 0.87, indicating that 89% of the test instances are classified

correctly. As the macro-average is the unweighted average of precision, recall, and F1-score across both classes, it provides equal weight to both classes. The macro-average in this model for F1-score is 0.88. In this model, the weighted average F1-score is 0.87, so we can say that the weighted average of precision, recall, and F1-score, where each class's score is weighted by its support, is 0.87.

Overall, this classification model performs exceptionally well in terms of F1-score for both classes. For class 1 (recall: 0.81), it has higher recall than class 0 (recall: 0.94). The accuracy of 0.89 indicates that the model correctly classifies 89% of the test instances. The precision for class 1 is relatively high (0.88), indicating a low number of false positives for class 1 predictions.

Figure 4.5 illustrates the confusion matrix of the deep neural network based suicidal sentiment classification model where word2vec feature with LSTM is used for classification. The prediction of data for 0 and 1 classes are 43 and 25. The total data used for testing is 79 in that 68 are predicted according to the actual class, and the rest of the 8 are predicted wrongly.

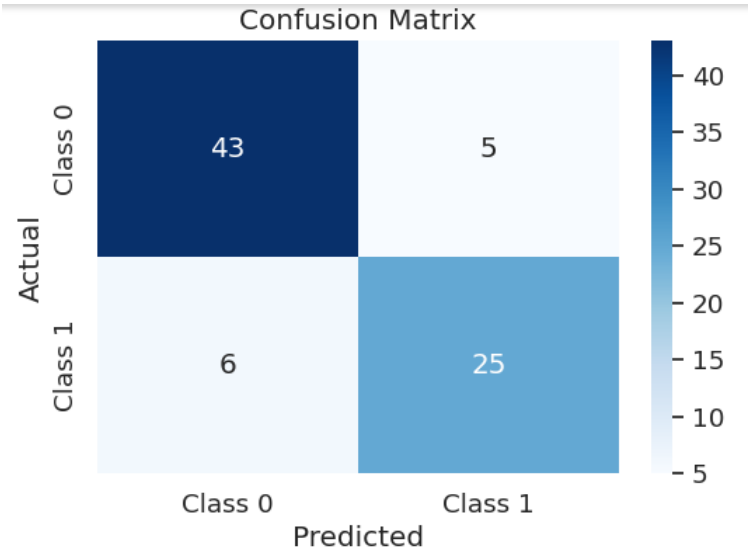

Figure 4.5: Confusion matrix of word2vec feature with LSTM classification performance

Figure 4.6 depicts the Receiver Operating Characteristic Curve (ROC) for the Word2vec feature with LSTM suicidal ideation prediction model. ROC (receiver operating characteristic curve) graph represents the classification model's performance across all thresholds. AUC stands for Area under the ROC Curve, which measures the total two-dimensional Area beneath the entire ROC curve. Two variables, such as the true and false positive rates, are used to plot the curve. For the best classifier, the range under ROC and AUC should be close to 1.

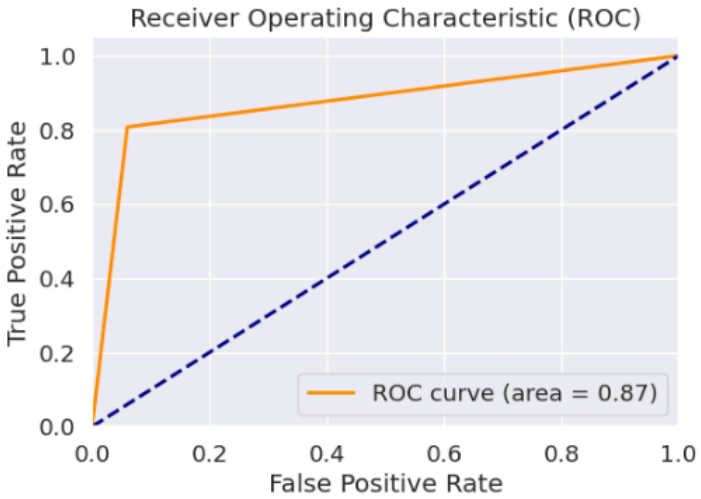

Figure 4.6: ROC of word2vec feature with LSTM classification performance

4.3 Pipeline 2: LIWC feature based GNN approach

The accuracy, precision, recall and F-measure calculated from this classification is as follows :

Table 4.4: LIWC feature with graph neural network based classification report

Metric	**0(Non-suicidal)**	**1(Suicidal)**	**Accuracy**
Precision	0.98	0.98	0.98
Recall	0.98	0.98	0.98
F1-score	0.98	0.98	0.98
Support	280	249	529
Macro Avg.	0.98	0.98	0.98
Weighted Avg.	0.98	0.98	0.98

For both classes 0 and 1, the precision is 1.00. We can say that 100% of the instances predicted as class 0 and class 1 were correct. For both classes 0 and 1, the recall is 1.00 and 100% of the actual instances of both class 0 and class 1 are correctly predicted. The F1-score is 1.00 for both classes 0 and 1. There are 280 instances of class 0, and 249 instances of class 1. 100% of the test instances are classified correctly as the accuracy is 1. In this model, the macro average and weighted average of precision, recall, and F1-score are all 1.00.

Overall, this classification report suggests that the model performs exceptionally well, achieving perfect precision, recall, and F1-scores for both classes 0 and 1, with an overall accuracy of 1.00. This level of performance indicates that the model is making correct

predictions for all instances in the dataset. However, it's important to validate the model on an independent test dataset to ensure that it generalizes well to new and unseen data.

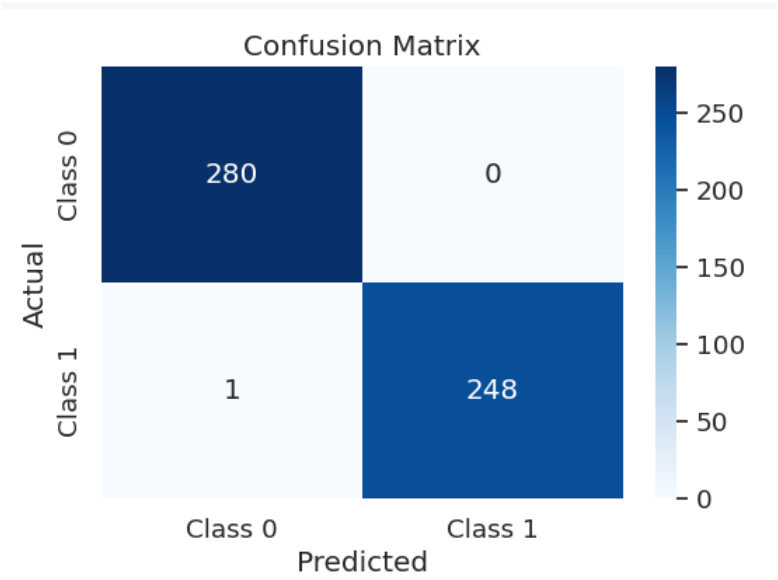

Figure 4.7: Confusion matrix of LIWC feature with graph neural network based classification performance

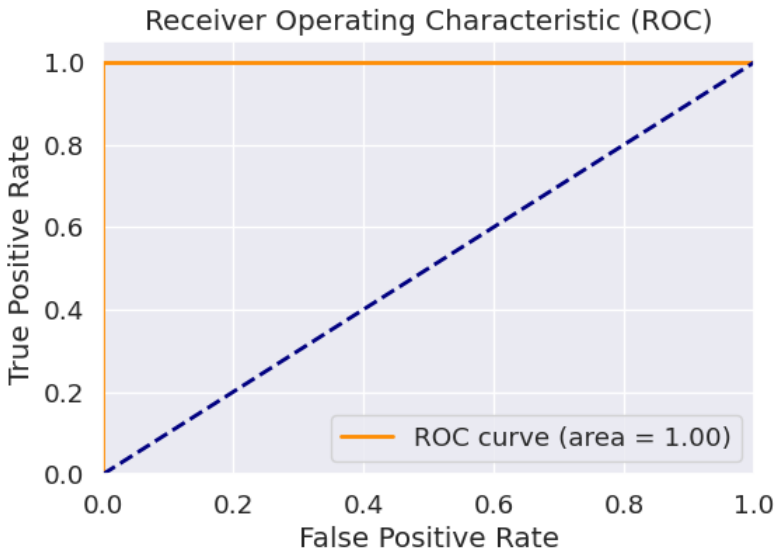

Figure 4.8: ROC of LIWC feature with graph neural network based classification performance

The confusion matrix of the classification performance is represented on figure 4.7 The prediction of data for 0 and 1 classes are 280 and 248. The total data used for testing is 529 in that 528 are predicted according to the actual class, and the rest of the 1 are predicted wrongly.

Figure 4.8 depicts the Receiver Operating Characteristic Curve (ROC) for the LIWC feature with graph neural network based suicidal ideation prediction model. ROC (receiver operating characteristic curve) graph represents the classification model's performance across all thresholds. Two variables, such as the true and false positive rates, are used to plot the curve. This is the best classifier as the range under ROC and AUC us close to 1.

4.4 Ablation Study

4.4.1 Pipeline 1: Deep learning based semantic topic modeling approach

We conducted ablation study to understand the impact of different components or features on the overall performance of our model so that the ablation study with different configurations can provide valuable insights into the contribution and effectiveness of each component. We evaluated the model's performance using only word2Vec features without incorporating BERT embeddings which determines the baseline performance of the model with traditional word embeddings. This helps understand how well the model performs without the additional contextual information provided by BERT embeddings. We then assessed the impact of incorporating BERT embeddings, specifically for English text to understand how much improvement BERT embeddings bring to the model's performance when focusing only on English text. This helps evaluate whether the contextual information from BERT is beneficial for suicidal ideation detection in English social media posts. To evaluate this, we have used total 785 social media posts of English language with human annotation. We finally investigated the combined impact of BERT and Bangla BERT embeddings along with word2Vec features, considering a mix of English, Bangla, and Banglish texts. We explored the model's ability to handle multilingual and code-switched content to evaluate whether the combination of BERT and Bangla BERT embeddings enhances the model's performance across different languages and language-mixed posts. In this regard we first used 700 social media posts of English language, 80 social media posts of Bangla language and 5 social media posts of Banglish language. The posts were all human annotated following the annotation strategy described on the "Data Collection and Annotation" chapter. For the combined approach of BERT and Bangla BERT embeddings along with word2Vec features, the performance was very poor at first. This was because of the Bangla and Banglish data we used was not sufficient to generalize well to the characteristics of Banglish and Bangla text. Limited data may resulted in less robust learning for these language types. We know that transliteration process introduces an additional step that might have errors, impacting the performance. If the transliteration accuracy is not high, it may affect the model's ability to understand the true meaning of the text. To resolve this, we augmented the Bangla and Banglish datasets through data synthesis or translation to artificially increase the size of these datasets. We used Gamitisa, an online Phonetic Transcription Tool, to convert Banglish text to Bangla. We observed a gradual improvement over the performance of our model using the combined approach of BERT

and Bangla BERT embeddings along with word2Vec features after data augmentation. The comparison among different components of our suggested model is shown on table 4.5.

Table 4.5: Performance evaluation of ablation study in deep neural network based suicidal text classification

Model Component	**Language Coverage**	**Data Size**	**Accuracy**	**Precision**	**Recall**	**F1-score**
word2vec + LSTM	785 (English posts)	380 (Suicidal posts), 405 (Non-suicidal posts)	0.87	0.87	0.87	0.87
word2vec + BERT + LSTM	785 (English posts)	380 (Suicidal posts), 405 (Non-suicidal posts)	0.87	0.87	0.87	0.87
word2vec + BERT + Bangla BERT + LSTM	785 (700 English posts, 80 Bangla posts and 5 Banglish posts)	380 (Suicidal posts), 405 (Non-suicidal posts)	0.68	0.68	0.68	0.68

4.4.2 Pipeline 2: Psycho-linguistic lexical feature based graph neural network driven approach:

In the graph neural network based suicidal text classification model, we made an ablation study on the feature selection. At first, we have evaluated the model performance using the most significant features. We used ANOVA, boruta and Fisher's Score for selecting the significant features. We observed a decline in the model's performance using those feature selection methods. For ANOVA, the model accuracy becomes 73.72%. As ANOVA assumes a linear relationship between individual features and the target variable, ANOVA might not capture the true impact of the features with non-linear relationship leading to suboptimal feature selection[100]. Also, ANOVA doesn't account for multicollinearity when features are highly correlated with each other[100]. In such cases, ANOVA might select one correlated feature discarding others which results in a loss of valuable informa-

tion. ANOVA is typically used for numerical features and assumes normally distributed data[100]. We have categorical features or non-normally distributed data. So, ANOVA might not be suitable and it may lead to feature selection errors. For these reasons, ANOVA feature selection method doesn't show remarkable improvement of our graph neural network based model performance. The accuracy becomes 69.84% using Boruta feature selection method and 78.90% using Fisher score feature selection method. Graph data can be highly complex having intricate relationships between nodes. Boruta is an extension of Random Forest. So, it may not be well-suited to capture these complex dependencies. It may be difficult to distinguish relevant graph nodes and edges from noisy or irrelevant ones.

Table 4.6: Performance evaluation of ablation study in graph neural network based suicidal text classification

Model Component	**Language Coverage**	**Data Size**	**Feature**	**Accuracy**
LIWC + GNN	785 (English posts)	380 (Suicidal posts), 405 (Non-suicidal posts)	ANOVA selected features	73.72%
LIWC + GNN	785 (English posts)	380 (Suicidal posts), 405 (Non-suicidal posts)	Boruta selected features	69.84%
LIWC + GNN	785 (English posts)	380 (Suicidal posts), 405 (Non-suicidal posts)	Fisher Score selected features	78.90%
LIWC + GNN	785 (English posts)	380 (Suicidal posts), 405 (Non-suicidal posts)	All features	98.00%

Moreover, the graph representation in Boruta may not effectively capture the structural information present in the graph data. The standard Random Forest approach is designed for tabular data. In terms of graph data, this representation may not be informative enough. Boruta can potentially select too many graph nodes or edges which can occur overfitting issues. The excessive features can result in noise and complexity into the GNN model and can make it less capable of generalizing to new data. Fisher's Score is most effective when a binary classification problem is chosen with distinct class labels[101]. We tried Fisher's Score feature selection method for our imbalanced dataset and class distributions. In that case, Fisher's Score might prioritize features that distinguish the majority class while neglecting the minority class and this may lead to reduced model performance

when the minority class is important. Fisher's Score assumes that the data follows a normal distribution. If this assumption doesn't hold, the method may not accurately reflect the feature's importance[101]. We are dealing with imbalanced data. Also, the data do not have normal distribution. So, we are not able to consider a remarkable improvement over the data for this method.

4.5 Model Performance Analysis on True positive and False Negative

True Positive (TP) represents correctly identified instances of suicidal ideation. High TP indicates effective identification, aiding in providing timely support or intervention for individuals expressing suicidal thoughts. False Negative (FN) represents instances of suicidal ideation that were not detected. High FN values may lead to missed cases, potentially delaying support for individuals in distress, posing a serious concern for their well-being. Balancing these values is crucial. False negatives can have severe consequences. So, efforts to minimize them are important. However, a good balance is needed to avoid excessive false positives, which could lead to unnecessary interventions and strain on resources. The goal is to maximize true positives while minimizing false negatives. Considering this, we evaluated our model performance. The best performant model among these is the graph neural network based model using all LIWC features. The model has 280 as true positive and 0 as false negative with selecting all of the features. As the main goal is to minimize the false negative and to maximise true positive, this model performed best. After that, deep learning based model with word2vec features and BERT features to LSTM performs well which has true positive of 47 and false negative of 5. We are considering two models, one is from deep learning based model and another one is from graph neural network based model. WE can consider deep neural network based model for proper detection of suicidal ideation to a decision support system for providing support to the majority of suicidal people so that less number of people miss it.

4.6 Model Performance Analysis on False positive and True Negative

False Positive (FP) represents instances incorrectly identified as suicidal ideation. High FP values may result in false alarms and it may cause unnecessary interventions or alerts. This can lead to a waste of resources and potential distress for individuals who do not actually express suicidal thoughts. On the other hand, True Negative (TN) represents correctly identified instances as not indicating suicidal ideation. High TN values are generally desirable as they indicate accurate identification of non-suicidal content. This helps in avoiding unnecessary interventions and conserving resources for cases that truly require attention. This time too, our deep neural network based model with word2vec and

BERT features to LSTM (21 as true negative and 3 as false positive) performs best and surpasses some traditional methods. After that, graph neural network based model with all LIWC features performs well. . The ensemble of these two model ensures unnecessary interventions to detect suicidal ideation from social media data.

Chapter 5

Comparison with Baseline Models

We evaluate the performance of baseline models and our proposed model based on different performance metrics. The prime purpose of this research is to showcase and find out the best model of evolving work on automatic recognition of suicidal content. In the baseline models, we use single handcrafted features, such as TF-IDF [14] and word2vec [22] which are applied on SVM [41], LR [43], NB [44], and RF [42]. In the proposed model, in pipeline 1, we incorporate BERT [45] features with word2vec [22] to get contextualized features for modeling. The major aim of amalgamating different NLP techniques is to investigate on the features that best suits the performance accuracy for SI detection. We observe that the performance is same for both the combine features of BERT and word2vec and only word2vec. In pipeline 2, we use psycho-linguistic lexical features [15] for modeling on graph neural network for SI detection. We try to show on two different pipelines that how the DL based model perform on a pre-trained language modeling features in pipeline 1 and how the GNN model perform on LIWC [15] features. Table 5.1 shows the performance of different ML models used as the baseline models along with our proposed models. By performance assessment of ML methods in the baseline we notice that the performance of SVM where TF-IDF features are used is the best in terms of scoring. Accuracy achieved with SVM is 92%. After that, NB with TF-IDF features performs best with 87% of accuracy among all other ML techniques except SVM with TF-IDF features.

Contrasting the proposed model with the baseline, we can conclude that the best classification result is attained with pipeline 1 of our proposed models with 87% of accuracy. Pipeline 1 performs better than SVM with word2vec features, LR with word2vec features and RF with word2vec features. Pipeline 1 has similar performance with NB where TF-IDF features are used.

Table 5.1: Comparison with baseline models based on performance metrics

Classification Model	**Feature**	**Accuracy**	**Precision**	**Recall**	**F1-score**
Baseline Model					
SVM	TF-IDF	0.92	0.92	0.92	0.92
SVM	Word2vector	0.68	0.75	0.68	0.58
LR	Word2vector	0.65	0.50	0.65	0.55
NB	TF-IDF	0.87	0.87	0.87	0.87
RF	Word2vector	0.54	0.00	0.00	0.00
Our suggested Model					
LSTM (pipeline 1)	Word2vector + Bert	0.87	0.87	0.87	0.87
GNN (pipeline 2)	LIWC	0.78	0.78	0.78	0.78

TF-IDF is used to convert the text data into numerical feature vectors where each document is represented as a vector of TF-IDF scores. SVM uses this TF-IDF vectors to find the optimal hyperplane that separated the classes (e.g., suicidal vs. non-suicidal). SVM is effective with high-dimensional sparse data, such as TF-IDF features. TF-IDF captures the importance of words in a document relative to the entire corpus which is useful for text classification tasks [46]. Word2vec captures semantic relationships between words but it may not perform as well with SVM due to the less discriminatory nature of these features compared to TF-IDF. SVM performs better with sparse and high-dimensional data [41]. For this, the performance has degraded for SVM with word2vec. NB uses the TF-IDF vectors to calculate the posterior probabilities of each class and make predictions. NB assumes feature independence which works well with TF-IDF features in text classification. The probabilistic nature of NB makes it suitable for handling the variability in suicidal text data [47]. Thus, NB shows better performance for classification of suicidal data. Linear Regression aims to find the best-fit line that predicts the dependent variable (target) as a linear combination of the independent variables (features). For binary classification, the decision boundary is a hyperplane that separates the data into two classes. Word2vec is used to convert text data into numerical feature vectors. Each word in a document is represented as a dense vector, and these vectors can be aggregated to form a document-level feature vector (e.g., averaging the word vectors. LR uses these feature vectors to find the decision boundary that best separates the classes (e.g., suicidal vs. non-suicidal) [52]. The combination of word2vec and LR may struggle with the task of suicidal text classification due to the limitations of capturing context, handling non-linear decision boundaries, and dealing with the nuanced data. LR performs better with features that have clear separability [48]. RF uses word2vec feature vectors to train multiple decision trees and make predictions. RF can capture feature interactions within its decision

trees. If the features (word vectors) do not adequately represent the underlying semantic and syntactic relationships in the text, the model's performance will suffer [49]. RF is less effective with dense vector representations like word2vec as they rely on feature importance and splitting which doesn't align well with the continuous nature of word2vec embeddings [50]. RF has the least performance for classification of suicidal text. LSTM networks are effective for sequence data and can capture temporal dependencies in text [51]. Combining word2vec with BERT leverages both semantic relationships and contextual understanding which is responsible for improving performance [52]. GNN effectively captures the structural relationships in data [53]. LIWC provides psycho-linguistic features which are highly informative for understanding the psychological state of users [54]. GNN with LIWC features has better performance than SVM with word2vec features, LR with word2vec features and PF with word2vec features.

Chapter 6

Discussion and Findings

Our cultivated models have the potential to offer several benefits for suicidal ideation detection in social media platform. After studying the focused area for identification of suicidality, we have narrowed down the scope of the models to meet our research goals and objectives.

Sequential Data Handling with Contextualization: LSTM (Long Short-Term Memory) networks are well-suited for analyzing sequential text data. But, it is critical to deal with social media posts or conversations where context is essential. TF-IDF features help to normalize the issues.

Dimensionality Reduction and Contextual Labeling: Latent Semantic Indexing is beneficial for identifying topics related to mental health and suicidal ideation which can effectively capture latent topics within textual data. LSI helps to reduce the dimensionality of the TF-IDF matrix preserving semantic information. The feature space can be more manageable. Also, labeling text data with relevant topics which are derived from LSI provides contextual information. It helps in categorizing posts and conversations based on the content and promotes more targeted analysis.

Combination of Semantic and Contextual Information for Comprehensive Understanding: Combination of Word2Vec and BERT embeddings in the model helps to gain access to both semantic and contextual information. This embeddings can improve the model ability to recognize subtle cues and expressions associated with suicidal ideation. By combining LIWC features with GNNs, it's possible to create a model with interpretable features. This can help clinicians and mental health professionals understand which linguistic and contextual factors contribute to the detection of suicidal ideation.

Holistic Approach to Gain Comprehensive Understanding: Our multi-step outlined deep neural network based holistic approach incorporates both traditional natural language processing (NLP) techniques (TF-IDF, LSI) and state-of-the-art deep learning (LSTM, BERT) to gain a comprehensive understanding of the text data.

False Positives Reduction: A model that integrates both semantic and contextual information is likely to be more accurate in identifying true signals of suicidal ideation and reduces the likelihood of false positives. GNNs can be a potential use to reduce

false positive rates by considering the context and social connections. As it works by understanding the relationships between individuals. It can help differentiate between general discussions about mental health and situations where intervention is urgently needed.

Scalability and Real-Time Monitoring: The concatenated Word2Vec and BERT embeddings provide a more compact and informative input for downstream machine learning models. This compact form of data reduces the computational complexity and improves model efficiency. GNN is used to handle large-scale data in real-time. This is essential for monitoring online platforms and providing timely interventions which can be life-saving.

6.1 Findings

The LSI, in the deep learning based model can't extract topics explicitly. Our analysis of suicide related social media posts in Bangladesh have identified six major topics based on the terms in each topic and the context. Table 6.1 shows these topics, top 20 keywords and social media posts related to the topics. We now examine each topic closely and analyze social, cultural, geographical, and Socio-economic aspects of suicide in Bangladesh.

Suicidality and Mental Health: All of our collected data from different social media is related to suicidal incidences. For example: from the terms "suicide," "dazai," "explained", this topic might focus on discussions related to suicidality, including references to Osamu Dazai and explanations or insights into the topic. Phrases like "I don't feel like this is a life worth living" and "I wish this would just end" directly indicate the presence of suicidal thoughts. These expressions reflect a deep sense of hopelessness and despair. Multiple expressions in the text, such as "Grades affecting my mental health" and "I am tired of being weak and destroyed," reveal a strong connection between mental health struggles and suicidal ideation. Expressions of frustration with society and a feeling of not being able to live life "for me myself and I" may indicate a sense of isolation and alienation, which can be closely tied to mental health struggles.

Cases of Suicide: "6 dead in Texas in apparent murder-suicide after brothers made pact, police say" - This text reports a specific incident of murder-suicide involving two brothers in Texas. It provides details about a tragic event related to suicide. The statement "The chance of a 15-year-old boy dying by the age of 50 is now higher in America than in Bangladesh" highlights a comparative perspective on suicide rates in different countries. It raises questions about the factors contributing to these differences. The risk factors may intersect with suicide cases, although the terms themselves don't provide specific case details. "The Death of the Family Unit" - This text discusses the concept of the family unit, and while it may not directly discuss suicide rates or cases, disruptions in family structures can be linked to mental health issues and potentially suicidal risk. "6 dead in Texas in apparent murder-suicide after brothers made pact, police say" - This text reports a specific incident of murder-suicide involving two brothers in Texas. It provides

details about a tragic event related to suicide.

Suicidal Risk Factors: Our collected data expresses different suicide reasons which we can identify as risk factors of suicide. The risk factors are academic stress, online experiences and relationships, identity struggles, legal issues and human rights violations, intersecting factors, addiction, domestic violence, false accusation, family issues, mental, physical issues, poverty etc. The text touches on factors such as cultural, religious, and social issues. "I'm suffering from serious depression & life risk."- This text mentions serious depression and life risk, which is an explicit reference to suicidal risk.

Different suicide methods can also be identified as risk factor. The identified methods are, drugs, poisons, cutting instruments, firearms, jumping over, cutting neck, electric shock, firing, poison, railway track, taking injection, taking pills etc. We made an analysis on those risk factors to find out the level of risk among them. Our statistical analysis of the suicidal reasons and risk factors can be used to recognize the most severe risk factors. We made analysis on the suicidal cases based on the risk factors.

Social Media on Suicide: Approximately 4,450 teens/children commit suicide due to cyber bullying[69]. Social Media based websites and applications are platforms that enable users to create and share content. Cyber bullying is the use of those social networking sites to bully another. Social media negatively affects teens with an overdose of technology. Many people are negatively impacted by social media. Teens with suicidal risk can be influenced by the suicidal news on social media. Also, teens can be in extreme depression with an overdose of technological use.

Though social media has so many positive sides. For example, it can be used to educate young people. Platforms like YouTube can be used for educational videos and more at their disposal. Social media can also be a source of reason in society. Creating an Instagram account can be of good use to post about today's political issues around the world to inform peers. Social media can give teens the creative freedom and can be a good platform to share creative thoughts and ideas that they need.

Unfortunately, social media acts as a driving force of suicidal tendency among teens and adolescents. As most of the suicidal reasons we found are either of relationship issues, emotional arrogance, harassment, mental issue, family issue, depression etc. These reasons are directly or indirectly related to the negative effects of social media. The phrase "Had a falling out with my online ex last year" suggests that online interactions and experiences might also be related to mental health struggles, although not necessarily suicide cases. Reasons like harassment and depression are directly related to the use of social media. Sexual assault, embarrassment and shame, spreading doctored photo are the outcome of using social media. Phrases like "not manipulated by corrupt mods" imply that online communities may have some influence on individuals' mental health, potentially related to bullying or harassment, and could be related to cases of mental distress.

Society and Culture on Suicide: The social and cultural factors correlated with suicide at different levels. Critical life events or circumstances are responsible for suicides. Individuals who face divorce, economic strain, or political repression are often character-

ized as suicide risks.

Individuals living in areas of low social integration have higher risk of suicide. Less family attachments influence suicide probability. Some researchers maintain that the family unit is the single most important factor in understanding suicide. Whatever the societal context, living alone increases the risk of suicide.

Family and other social support are protective factors for suicide. Those who enjoy close relationships with others can cope better with various stresses, including bereavement, job loss, and physical illness, and enjoy better psychological and physical health. Studies have documented that social support can attenuate severity of depression and can speed remission of depression in at-risk groups such as immigrants and the physically ill. "it's a rant since I have no one to talk to and I suppose strangers could be helpful". It reflects an individual's personal experience with loneliness and social isolation which may lead to suicidal thoughts.

As mentioned above, completed suicide occurs more often in those who are socially isolated and lack supportive family and friendships. Evidence suggests different mechanisms of support's influence. Social support sometimes represents part of a protective process that increases self-efficacy and thereby reduces suicidal behaviour. At other times social support more directly reduces suicidality via reducing psychic distress. Online supporting communities are also contributing to reduce suicidal risks. The terms "Helpline numbers: Bangladesh" and "Suicide prevention hotlines" indicate the presence of resources and support systems aimed at preventing suicides. These are measures designed to offer assistance to those in crisis.

Furthermore, family and friendship support appear to play somewhat different roles in protecting against suicidality; men and women may differ in use and types of social support. Effective treatment for suicidality, whether medical or psychosocial, involves human contact and support. Recent suicide prevention programming to in- crease social support and other positive variables builds on emerging evidence suggesting a greater ameliorative effect of increasing protective factors than reducing risk.

Table 6.1: Top 20 keywords, top 5 topics and related posts.

Keywords	**Topics**	**Post**
life(99), mental(41), depression(29), health(28), feel(28), anxiety(7).	Suicidality and mental health.	"Grades affecting my mental health".
suicide (365), bangladesh (141), commit(44), die(37), country(26), attempt(11).	Cases of suicide	"6 dead in Texas in apparent murder-suicide after brothers made pact, police say"
parents(46), family(29), social(24), money(20), exam(20), break(11), teacher(11), reason(11), grade(5), abused(5), issues (4).	Suicidal risk factors	"I'm suffering from serious depression & life risk."
media(45), social(24), post(14), websites(12), break(11), video(8), emotion(5), you-tube(4), facebook(4), teenagers (4).	Social media on suicide	"Had a falling out with my online ex last year"
friend(30), prevention(25), social(24), squad(11), protect(7), community(4).	Society and culture on suicide	"it's a rant since I have no one to talk to and I suppose strangers could be helpful."

6.2 Open Issues and Challenges

This section presents the main open issues, challenges and the future perspectives of suicidal ideation detection based on social media data in the context of developing countries like Bangladesh.

Data Privacy and Ethics: Maintaining user privacy along with balancing the need for mental health support is a critical challenge. Implementing ethical guidelines and ensuring compliance with privacy regulations is an ongoing concern.

Cross-Cultural Sensitivity: Adapting the model to diverse cultural contexts within Bangladesh and accounting for variations in language use and expressions poses a challenge. One-size-fits-all approaches may not be effective across different cultural groups.

Real-Time Intervention: Real-time data processing, model prediction latency, and prompt intervention strategies are required to provide timely support to individuals at risk overcoming challenges related to it.

Handling Code-Switching: Dealing with code-switching between languages in social media posts, especially prevalent in Banglish, remains a challenge. Models need to effectively handle this linguistic complexity for accurate detection.

Limited Labeled Data: The scarcity of accurately labeled data for suicidal ideation in the context of Bangladesh poses challenges for model training and evaluation. Obtaining high-quality, diverse labeled data is essential for model performance.

Algorithmic Bias and Fairness: Ensuring fairness and mitigating biases in the model's predictions, especially regarding demographic groups, is an ongoing challenge. Regular audits and fairness-aware training techniques can help address this issue.

Long-Term Impact Assessment: Assessing the long-term impact of the model on individuals' mental health and well-being requires careful longitudinal studies and collaboration with mental health experts.

Dynamic Nature of Social Media: Adapting to the rapidly evolving language and behaviors on social media platforms presents challenges. Regular model updates and staying informed about emerging trends are necessary.

Scalability and Deployment in Resource-Limited Settings: Ensuring that the model is scalable and can be effectively deployed in resource-limited settings, which may be common in some areas of Bangladesh, poses a challenge.

Chapter 7

7.1 Practical Implications

Our proposed pipeline and study findings offer various practical implications for stakeholders. This pipeline is able to detect social media users with SI by monitoring their social media posts with suicidal ideation and making timely intervention possible.

- Our system can enable early intervention by accurately identifying posts indicative of suicidal ideation. Mental health professionals or support systems can be alerted promptly which can potentially prevent suicide attempts.
- Our research and findings can provide mental health professionals and organizations with an alternative means to the traditional way of direct consultation.
- Most of the people with mental health issues do not visit healthcare professionals for treatment because their mental illness remains heavily stigmatized. Therefore, their engagement with social media platforms keeps great promise for the detection and intervention of SI so that their family members can understand their mental health issues.
- Social media platforms can use our system to identify and flag potentially harmful content ensuring timely moderation and providing support resources to users who may be in distress.

7.2 Limitations

One critical challenge that demands attention is the prevalence of imbalanced data in the training set. Imbalances between classes, particularly in the context of suicidal and non-suicidal posts, introduce complexities that necessitate careful consideration. Imbalanced data, where one class (e.g., non-suicidal posts) significantly outnumbers the other (suicidal posts), can lead the model to be biased towards the majority class. As a result, the model may perform well on the majority class but poorly on the minority class. Models trained on imbalanced data may have difficulty generalizing to new, unseen data, particularly for the underrepresented class. The model might exhibit poor performance when

faced with real world imbalances. The model's performance is sensitive to the choice of hyperparameters in various components such as TF-IDF, LSI, word2vec, and LSTM in pipeline 1. Suboptimal hyperparameter tuning has impact on the overall efficacy of the model. The performance of word2vec and BERT embeddings depends on pre-trained models. Changes in linguistic trends or the emergence of new expressions may pose challenges if these embeddings do not adequately capture evolving language patterns. In pipeline 2, the way of graph construction for GNN is crucial. If the connections between nodes are not well-defined or do not accurately reflect the relationships relevant to suicidal ideation, the GNN's performance might be compromised. Suicidal language can be subtle, context dependent and influenced by cultural factors and LIWC alone might not capture.

7.3 Future Research

We target some potential avenues for future work on our suggested model. Incorporation of additional modalities such as images, videos, or audio data from social media posts can create a more comprehensive understanding of user sentiment and mental health. Degree of severity of suicidal ideation can be calculated to provide the community based support properly. We plan to implement mechanisms for continual learning to adapt the model over time as language and online behaviors evolve.

7.4 Conclusion

Our research sheds light on the alarming rise of suicide rates particularly among the youth both globally and in Bangladesh. Leveraging the expressive nature of social media platforms, our study targets Facebook, Twitter, and Reddit for employing advanced machine learning models to detect individuals at risk of suicidal ideation. The integration of two models (deep neural network based model with 87% of accuracy and graph neural network based model with 78% of accuracy) with their unique strengths yields comprehensive insights that surpass conventional techniques in terms of recall. The identification of five contextual topics adds depth to our understanding and provides a foundation for proactive policy approaches. In essence, our mission is to combat the insidious parasite of suicide through cutting-edge technology by offering a message of hope and assistance to individuals struggling with suicidal thoughts. By empowering policy-making approaches, we aspire to create a positive impact on mental health and fostering a society that prioritizes early intervention, prevention, and support for those in crisis.

References

[1] Sharma, S., Parolia, A., Kanagasingam, S.: A review on covid-19 mediated impacts and risk mitigation strategies for dental health professionals. European journal of dentistry 14(S 01), 159–164 (2020) 22.

[2] Bridge, J.A., Ruch, D.A., Sheftall, A.H., Hahm, H.C., O'Keefe, V.M., Fontanella, C.A., Brock, G., Campo, J.V., Horowitz, L.M.: Youth suicide during the first year of the covid-19 pandemic. Pediatrics 151(3) (2023).

[3] La Sala, L., Pirkis, J., Cooper, C., Hill, N.T., Lamblin, M., Rajaram, G., Rice, S., Teh, Z., Thorn, P., Zahan, R., et al.: Acceptability and potential impact of the chatsafe suicide postvention response among young people who have been exposed to suicide: pilot study. JMIR human factors 10, 44535 (2023).

[4] Beghi, M., Butera, E., Cerri, C.G., Cornaggia, C.M., Febbo, F., Mollica, A., Berardino, G., Piscitelli, D., Resta, E., Logroscino, G., et al.: Suicidal behaviour in older age: A systematic review of risk factors associated to suicide attempts and completed suicides. Neuroscience Biobehavioral Reviews 127, 193–211 (2021).

[38] Kim, J., Lee, D., Park, E., et al.: Machine learning for mental health in social media: bibliometric study. Journal of Medical Internet Research 23(3), 24870 (2021).

[6] Pretorius, C., Chambers, D., Coyle, D.: Young people's online help-seeking and mental health difficulties: Systematic narrative review. Journal of medical Internet research 21(11), 13873 (2019).

[7] kintoye, O., Wei, N., Liu, Q.: Suicide detection in tweets using lstm and transformers. In: 2024 4th Asia Conference on Information Engineering (ACIE), pp.22–27 (2024). IEEE.

[8] Haque, R., Islam, N., Islam, M., Ahsan, M.M.: A comparative analysis on suicidal ideation detection using nlp, machine, and deep learning. Technologies 10(3), 57 (2022)

[9] Lange, J., Baams, L., Bergen, D.D., Bos, H.M., Bosker, R.J.: Minority stress and suicidal ideation and suicide attempts among lgbt adolescents and young adults: a meta-analysis. LGBT health 9(4), 222–237 (2022).

[10] Nesi, J., Burke, T.A., Bettis, A.H., Kudinova, A.Y., Thompson, E.C., MacPher- son, H.A., Fox, K.A., Lawrence, H.R., Thomas, S.A., Wolff, J.C., et al.: Social media use and self-injurious thoughts and behaviors: A systematic review and meta-analysis. Clinical psychology review 87, 102038 (2021).

[11] Fahey, R.A., Boo, J., Ueda, M.: Covariance in diurnal patterns of suicide-related expressions on twitter and recorded suicide deaths. Social Science Medicine 253, 112960 (2020).

[12] Organization, W.H., et al.: Suicide worldwide in 2019: global health estimates (2021).

[13] Shah, F.M., Haque, F., Nur, R.U., Al Jahan, S., Mamud, Z.: A hybridized feature extraction approach to suicidal ideation detection from social media post. In: 2020 IEEE Region 10 Symposium (TENSYMP), pp. 985–988 (2020). IEEE.

[14] Cheng, R.: Understanding tf-idf: A traditional approach to feature extraction in nlp learn the fundamentals of tf-idf and how to implement it from scratch in python.

[15] Boyd, R.L., Ashokkumar, A., Seraj, S., Pennebaker, J.W.: The development and psychometric properties of liwc-22. Austin, TX: University of Texas at Austin, 1–47 (2022).

[16] Zahura, A., Mamun, K.A.: Intelligent system for predicting suicidal behaviour from social media and health data. In: 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT), pp. 319–324(2020). IEEE.

[17] Staudemeyer, R.C., Morris, E.R.: Understanding lstm–a tutorial into long short-term memory recurrent neural networks. arXiv preprint arXiv:1909.09586 (2019).

[18] Tadesse, M.M., Lin, H., Xu, B., Yang, L.: Detection of suicide ideation in social media forums using deep learning. Algorithms 13(1), 7 (2019).

[19] Islam, S., Forhad, M.S.A., Murad, H.: Banglasapm: A deep learning model for suicidal attempt prediction using social media content in bangla. In: 2022 25th International Conference on Computer and Information Technology (ICCIT), pp.1122–1126 (2022). IEEE.

[20] Sanchez-Lengeling, B., Reif, E., Pearce, A., Wiltschko, A.B.: A gentle introduction to graph neural networks. Distill 6(9), 33 (2021).

[21] Singh, K.N., Devi, S.D., Devi, H.M., Mahanta, A.K.: A novel approach for dimension reduction using word embedding: An enhanced text classification approach. International Journal of Information Management Data Insights 2(1), 100061 (2022).

[22] Di Gennaro, G., Buonanno, A., Palmieri, F.A.: Considerations about learning word2vec. The Journal of Supercomputing, 1–16 (2021).

[23] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprintarXiv:1810.04805 (2018).

[24] Ezerceli¨O., Dehkharghani, R.: Mental disorder and suicidal ideation detection from social media using deep neural networks. Journal of Computational Social Science, 1–31 (2024).

[25] Priyamvada, B., Singhal, S., Nayyar, A., Jain, R., Goel, P., Rani, M., Srivastava, 24 M.: Stacked cnn-lstm approach for prediction of suicidal ideation on social media. Multimedia Tools and Applications 82(18), 27883–27904 (2023).

[26] Jere, S., Patil, A.P.: Detection of suicidal ideation based on relational graph attention network with dnn classifier. International Journal of Intelligent Systems and Applications in Engineering 11(10s), 321–332 (2023).

[27] Renjith, S., Abraham, A., Jyothi, S.B., Chandran, L., Thomson, J.: An ensemble deep learning technique for detecting suicidal ideation from posts in social media platforms. Journal of King Saud University-Computer and Information Sciences 34(10), 9564–9575 (2022).

[28] Pradha, S., Halgamuge, M.N., Vinh, N.T.Q.: Effective text data preprocessing technique for sentiment analysis in social media data. In: 2019 11th International Conference on Knowledge and Systems Engineering (KSE), pp. 1–8 (2019). IEEE

[29] Das, B., Chakraborty, S.: An improved text sentiment classification model using tf-idf and next word negation. arXiv preprint arXiv:1806.06407 (2018).

[30] Min, B., Ross, H., Sulem, E., Veyseh, A.P.B., Nguyen, T.H., Sainz, O., Agirre, E., Heintz, I., Roth, D.: Recent advances in natural language processing via large pretrained language models: A survey. ACM Computing Surveys 56(2), 1–40 (2023).

[52] Wu, T., He, S., Liu, J., Sun, S., Liu, K., Han, Q.-L., Tang, Y.: A brief overview of chatgpt: The history, status quo and potential future development. IEEE/CAA Journal of Automatica Sinica 10(5), 1122–1136 (2023).

[32] Ji-Won Baek and Kyungyong Chung. 2020. Context Deep Neural Network Model for Predicting Depression Risk Using Multiple Regression. IEEE Access 8 (2020), 18171–18181.

[33] Worth, P.J.: Word embeddings and semantic spaces in natural language processing. International Journal of Intelligence Science 13(1), 1–21 (2023).

[34] Arora, S., May, A., Zhang, J., R´e, C.: Contextual embeddings: When are they worth it? arXiv preprint arXiv:2005.09117 (2020).

[35] Bie, Y., Yang, Y., Zhang, Y.: Fusing syntactic structure information and lexicalsemantic information for end-to-end aspect-based sentiment analysis. Tsinghua Science and Technology 28(2), 230–243 (2022).

[36] Zhu, Y., Xia, Y., Li, M., Zhang, T., Wu, B.: Data augmented graph neural networks for personality detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 664–672 (2024).

[37] Liu, W., Peng, L., Wen, L., Yang, J., Liu, Y.: Decomposing shared networks for separate cooperation with multi-agent reinforcement learning. Information Sciences 641, 119085 (2023).

[38] Kim, T.K.: Understanding one-way anova using conceptual figures. Korean journal of anesthesiology 70(1), 22 (2017).

[39] Dinov, I.D., Dinov, I.D.: Variable/feature selection. Data Science and Predictive Analytics: Biomedical and Health Applications using R, 557–572 (2018).

[40] Dogra, V., Verma, S., Kavita, Chatterjee, P., Shafi, J., Choi, J., Ijaz, M.F.: A complete process of text classification system using state-of-the-art nlp models. Computational Intelligence and Neuroscience 2022(1), 1883698 (2022).

[41] Pisner, D.A., Schnyer, D.M.: Support vector machine. In: Machine Learning, pp.101–121. Elsevier, ??? (2020).

[42] Mai, H., Le, T.C., Chen, D., Winkler, D.A., Caruso, R.A.: Machine learning for electrocatalyst and photocatalyst design and discovery. Chemical Reviews122(16), 13478–13515 (2022).

[43] Hope, T.M.: Linear regression. In: Machine Learning, pp. 67–81. Elsevier, ???(2020).

[44] Yang, F.-J.: An implementation of naive bayes classifier. In: 2018 International Conference on Computational Science and Computational Intelligence (CSCI),pp. 301–306 (2018). IEEE.

[45] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).

[46] Landron-Rivera, B.A.: Text classification of student predicate use for conceptual change assessment. PhD thesis (2016).

[47] Kiilu, K.K.: Sentiment classification for hate tweet detection in kenya on twitter data using na¨ıve bayes algorithm. PhD thesis, JKUAT-COETEC (2021).

[48] Abbaschian, B.J., Sierra-Sosa, D., Elmaghraby, A.: Deep learning techniques for speech emotion recognition, from databases to models. Sensors 21(4), 1249 (2021).

[49] Alshahrani, A.: Automated Message Analyses Using Transformer-Based Models and Their Applications. The Florida State University, ??? (2021).

[50] Sarker, I.H.: Machine learning: Algorithms, real-world applications and research directions. SN computer science 2(3), 160 (2021).

[51] Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In:International Conference on Machine Learning, pp. 27268–27286 (2022). PMLR.

[52] Wu, C., Wu, F., Qi, T., Huang, Y., Xie, X.: Fastformer: Additive attention can be all you need. arXiv preprint arXiv:2108.09084 (2021) 26

[53] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.:Graph neural networks: A review of methods and applications. AI open 1, 57–81(2020).

[54] Ardia, D., Bluteau, K., Boudt, K., Inghelbrecht, K.: Climate change concerns and the performance of green vs. brown stocks. Management Science 69(12), 7607–7632 (2023).

[55] Pete Burnap, Gualtiero Colombo, Rosie Amery, Andrei Hodorog, and Jonathan Scourfield. 2017. Multi-class machine classification of suicide-related communication on Twitter.Online social networks and media 2 (2017), 32–44.

[56] Pete Burnap, Walter Colombo, and Jonathan Scourfield. 2015. Machine classification and analysis of suicide-related communication on twitter. In Proceedings of the 26th ACM conference on hypertext social media. 75–84.

[57] Lipisha Chaudhary, Vignesh Vijaykumar Nair, and Ishika Prasad. 2019. Descriptive Analysis of Suicide Ideation on Twitter. (2019).

[58] Qijin Cheng, Tim MH Li, Chi-Leung Kwok, Tingshao Zhu, and Paul SF Yip. 2017. Assessing suicide risk and emotional distress in Chinese social media: a text mining and machine learning study. Journal of medical internet research 19, 7 (2017), e243.

[59] Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. 2019. Hierarchical clustering: Objective functions and algorithms. Journal of the ACM (JACM) 66, 4 (2019).

[60] Alize J Ferrari, Rosana E Norman, Greg Freedman, Amanda J Baxter, Jane E Pirkis, Meredith G Harris, Andrew Page, Emily Carnahan, Louisa Degenhardt,Theo Vos, et al. 2014. The burden attributable to mental and substance use disorders as risk factors for suicide: findings from the Global Burden of Disease Study 2010. PloS one 9, 4 (2014), e91936.

[61] Krzysztof Fiok, Waldemar Karwowski, Edgar Gutierrez, and Tareq Ahram. 2020.Predicting the Volume of Response to Tweets Posted by a Single Twitter Account. Symmetry 12, 6 (2020), 1054.

[62] Kyung Sang Lee, Hyewon Lee, Woojae Myung, Gil-Young Song, Kihwang Lee, Ho Kim, Bernard J Carroll, and Doh Kwan Kim. 2018. Advanced daily prediction model for national suicide numbers with social media data. Psychiatry investigation 15, 4 (2018), 344.

[63] Gen-Min Lin, Masanori Nagamine, Szu-Nian Yang, Yueh-Ming Tai, Chin Lin, and Hiroshi Sato. 2020. Machine learning based suicide ideation prediction for military personnel. IEEE journal of biomedical and health informatics (2020).

[64] Maureen Barlow Pugh. 2000. Stedman's medical dictionary. Lippincott Williams Wilkins.

[65] Fuji Ren, Xin Kang, and Changqin Quan. 2015. Examining accumulated emotional traits in suicide blogs with an emotion topic model. IEEE journal of biomedical and health informatics 20, 5 (2015), 1384–1396

[66] Maya Semrau, Sara Evans-Lacko, Atalay Alem, Jose Luis Ayuso Mateos, Dan Chisholm, Oye Gureje, Charlotte Hanlon, Mark Jordans, Fred Kigozi, HeidiLempp, et al. 2015. Strengthening mental health systems in low and middle income countries: the Emerald programme. BMC medicine 13, 1 (2015), 79.

[67] M Johnson Vioules, Bilel Moulahi, Jérôme Azé, and Sandra Bringay. 2018. Detection of suicide-related posts in Twitter data streams. IBM Journal of Research and Development 62, 1 (2018), 7–1.

[68] Jie Zhang, Shuhua Jia, William F Wieczorek, and Chao Jiang. 2002. An overview of suicide research in China. Archives of Suicide Research 6, 2 (2002), 167–184.

[69] [n.d.].Effects Of Social Media On Teens (2021 official projections) https://www.bartleby.com/essay/Effects-Of-Social-Media-On-Teens-PZBGMWUPPR.

[70] J. Joo, S. Hwang, and J. J. Gallo, "Death ideation and suicidal ideation in a community sample who do not meet criteria for major depression," Crisis, 2016.

[71] Alexander R.E. (2001). Stress-Related Suicide by Dentists and Other Health Care Workers: Fact or Folklore?. Journal of the American Dental Association 132(6), 786-794.

[72] Aseltine R.H. and De Martino R. (2004). An Outcome Evaluation of the SOS Suicide Prevention Program. American Journal of Public Health 94(3), 446-451.

[73] Johansson L, Lindqvist P and Eriksson A. (2006). Teenage suicide cluster formation and contagion: implications for primary care. Biomedical Central Family Practice, 7, 32.

[74] Conwell Y., Duberstein P.R., Connor K., Eberiy S., Cox C. and Caine E.D (2002). Access to Firearms and Risk for Suicide in Middle-aged and Older Adults. American Journal of Geriatric Psychiatry, 10(4), 407-416.

[75] Di Gennaro, Giovanni, Amedeo Buonanno, and Francesco AN Palmieri. "Considerations about learning Word2Vec." The Journal of Supercomputing (2021): 1-16.

[76] Cheng, Raymond. "Understanding TF-IDF: A Traditional Approach to Feature Extraction in NLP Learn the fundamentals of TF-IDF and how to implement it from scratch in Python."

[77] Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina (11 October 2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". arXiv:1810.04805v2 [cs.CL].

[78] "Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing". Google AI Blog. 2 November 2018. Retrieved 2019-11-27.

[79] McCarley, J. S., Rishav Chakravarti, and Avirup Sil. "Structured pruning of a bert-based question answering model." arXiv preprint arXiv:1910.06360 (2019).

[80] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018)..

[81] Rosario, Barbara. "Latent semantic indexing: An overview." Techn. rep. INFOSYS 240 (2000): 1-16.

[82] Boyd, Ryan L., et al. "The development and psychometric properties of LIWC-22." Austin, TX: University of Texas at Austin (2022): 1-47.

[83] Sanchez-Lengeling, Benjamin, et al. "A gentle introduction to graph neural networks." Distill 6.9 (2021): e33.

[84] Kim, Tae Kyun. "Understanding one-way ANOVA using conceptual figures." Korean journal of anesthesiology 70.1 (2017): 22-26.

[85] Dinov, I.D. (2018). Variable/Feature Selection. In: Data Science and Predictive Analytics. Springer, Cham. https://doi.org/10.1007/978-3-319-72347-1_17

[86] https://www.statisticshowto.com/probability-and-statistics/z-score/

[87] Gu, Quanquan, Zhenhui Li, and Jiawei Han. "Generalized fisher score for feature selection." arXiv preprint arXiv:1202.3725 (2012).

[88] Staudemeyer, Ralf C., and Eric Rothstein Morris. "Understanding LSTM–a tutorial into long short-term memory recurrent neural networks." arXiv preprint arXiv:1909.09586 (2019).

[89] Demigha, Souad. "Decision Support Systems (DSS) and Management Information Systems (MIS) in Today's Organizations." European Conference on Research Methodology for Business and Management Studies. Academic Conferences International Limited, 2021.

[90] Horrocks, Matthew, et al. "An electronic clinical decision support system for the assessment and management of suicidality in primary care: protocol for a mixed-methods study." JMIR research protocols 7.12 (2018): e11135.

[91] Shah, F.M., Haque, F., Un Nur, R., Al Jahan, S., Mamud, Z. (2020). A Hybridized Feature Extraction Approach To Suicidal Ideation Detection From Social Media Post. 2020 IEEE Region 10 Symposium (TENSYMP), 985-988.

[92] A. Zahura and K. A. Mamun, "Intelligent System for Predicting Suicidal Behaviour from Social Media and Health Data," 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT), 2020, pp. 319-324, doi: 10.1109/ICAICT51780.2020.9333463.

[93] Jere, Shreekant, and Annapurna P. Patil. "Detection of Suicidal Ideation Based on Relational Graph Attention Network with DNN Classifier." International Journal of Intelligent Systems and Applications in Engineering 11.10s (2023): 321-332.

[94] Power, Daniel J. Decision support systems: concepts and resources for managers. Quorum Books, 2002.

[95] https://datareportal.com/reports/digital-2023-bangladesh

[96] https://wisevoter.com/country-rankings/facebook-users-by-country/

[97] Billah, Masum, and Enamul Hassan. "Depression detection from Bangla Facebook status using machine learning approach." Int. J. Comput. Appl 975 (2019): 8887.

[98] Platt, Stephen. "Unemployment and suicidal behaviour: a review of the literature." Social science medicine 19.2 (1984): 93-115.

[99] Yoshikawa, Hirokazu, J. Lawrence Aber, and William R. Beardslee. "The effects of poverty on the mental, emotional, and behavioral health of children and youth: implications for prevention." American psychologist 67.4 (2012): 272.

[100] Sawyer, Steven F. "Analysis of variance: the fundamental concepts." Journal of Manual Manipulative Therapy 17.2 (2009): 27E-38E.

[101] Sun, Lin, et al. "Feature selection using Fisher score and multilabel neighborhood rough sets for multilabel classification." Information Sciences 578 (2021): 887-912.

[102] Pennebaker, J. W., Boyd, R. L., Jordan, K., Blackburn, K. (2015). The Development and Psychometric Properties of LIWC2015. Technical Report.

[103] Tausczik, Y., Pennebaker, J. W. (2010). The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. Journal of Language and Social Psychology, 29(1), 24-54.

[104] S. Pradha, M. N. Halgamuge and N. Tran Quoc Vinh, "Effective Text Data Preprocessing Technique for Sentiment Analysis in Social Media Data," 2019 11th International Conference on Knowledge and Systems Engineering (KSE), Da Nang, Vietnam, 2019, pp. 1-8, doi: 10.1109/KSE.2019.8919368.

[105] Cao, Lei, Huijun Zhang, and Ling Feng. "Building and using personal knowledge graph to improve suicidal ideation detection on social media." IEEE Transactions on Multimedia 24 (2020): 87-102.

[106] Tadesse, Michael Mesfin, et al. "Detection of suicide ideation in social media forums using deep learning." Algorithms 13.1 (2019): 7.

[107] Renjith, Shini, et al. "An ensemble deep learning technique for detecting suicidal ideation from posts in social media platforms." Journal of King Saud University-Computer and Information Sciences 34.10 (2022): 9564-9575.

[108] Li, Yan, and Nianfeng Li. "Sentiment analysis of Weibo comments based on graph neural network." IEEE Access 10 (2022): 23497-23510.

[109] S. Islam, M. S. Alam Forhad and H. Murad, "BanglaSAPM: A Deep Learning Model for Suicidal Attempt Prediction Using Social Media Content in Bangla," 2022 25th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, 2022, pp. 1122-1126, doi: 10.1109/ICCIT57492.2022.10055237.

[110] World Health Organization. "Suicide worldwide in 2019: global health estimates." Suicide worldwide in 2019: global health estimates. 2021.

[111] M. Kowsher, A. A. Sami, N. J. Prottasha, M. S. Arefin, P. K. Dhar and T. Koshiba, "Bangla-BERT: Transformer-based Efficient Model for Transfer Learning and Language Understanding," in IEEE Access, 2022, doi: 10.1109/ACCESS.2022.3197662.