Sentiment Analysis of User Generated Online Content to detect Suicidal Tendencies

Vasu Jain
Department of Computer Science and Software Engineering
Concordia University
jain.vasu92@gmail.com
(Student ID: 40057063)

Simran Sidhu
Department of Computer Science and Software Engineering
Concordia University
simrantechm@gmail.com
(Student ID: 40011611)

Mandeep Kaur
Department of Computer Science and Software Engineering
Concordia University
kaur.mandeep2295@gmail.com
(Student ID: 40059801)

***Abstract*–Suicide has become a critical mental health problem faced globally. Online communication channels are becoming a new way for people to express their suicidal tendencies, due to the anonymity it offers. A lot of people prefer to channel their feeling on social networking websites and online forums seeking a cry for help. This gives researchers and data analysts, a new source of collecting general public data to analyze such behavior enabling them to early detect and prevent suspects of taking one's own life. This project aims to understand the approaches followed to detect suicidal ideation using supervised learning methods by processing the post uploaded by users on the Internet. This process classifies if an individual is a potential candidate for committing suicide from content posted on social websites. We analyze the user's choice of words, language preference, and topic selection in order to create the features which are able to classify a post as suicidal. Using these feature we will train the classifiers followed by the generation of a confusion matrix to evaluate the performance and accuracy of different classification algorithms.**

I. INTRODUCTION

According to WHO report, approximately 800,000 people die due to suicide every year, which is one person every 40 second. The study on this subject matter by WHO in 2016 revealed that suicide accounted for 1.4% of all deaths worldwide, making it the 18th leading cause of death [1]. Many factors can lead to suicide, for example, personal issues, such as hopelessness, severe anxiety, schizophrenia, alcoholism, or impulsivity; social factors, like social isolation and overexposure to deaths; or negative life events, including traumatic events, physical illness, affective disorders, and previous suicide attempts [2].

Motivation: A lot of research has been done to detect this behavior by collecting clinical data and suicidal notes – but this data was either expensive or far too less to be able to build an effective system. On the other hand, in today's time, more people tend to post their thoughts online, therefore making it easier to collect user-generated content from social websites. This ignited the idea how data mining techniques and machine learning algorithms can be used to identify people thinking of committing suicide and motivated us to take a step to help tackle the pressing issue of the modern society.

Objective: To detect suicidal ideation among people by processing the posts uploaded on the internet or social media websites and our main focus is on training the classifiers efficiently by building features, such as general lexical domain based features, linguistic, sentiment score, short forms or slangs, word embedding, which are capable of accurately classifying people who are prone to committing suicide. For this project, we have used user post from famous online website-Twitter. The data we collected from Twitter,

by using the keyword filtering technique, will just include a body content to be studied. The content extracted will be compared with the man-made rules indicating whether a post can be considered as suicide oriented or not. Additionally, to better understand suicidal individual we have evaluated their choice of the words, language and topic preferences.

Thus, in this paper we aim to use the texts in various online social platforms for suicide ideation. Using the existing studies related to the sentiments or emotions obtained from suicide notes and questionnaires conducted by various organizations, we can create useful features for our machine learning algorithms. The results from the studies help create a knowledge of words, grammar and parts of speech which people who have committed suicide tend to use. Our aim will be to use as many grammatical features as possible to create an application which will be able to make decent predictions and can help NGOs or government organizations make early predictions to prevent loss of lives.

II. BACKGROUND AND LITERATURE REVIEW

The increasing rate of suicide every year has led to many researchers and statisticians to work towards analyzing and developing a system/process that can aid in the early detection and prevention. The reasons for suicide are complicated and attributed to a complex interaction of many factors [3]. The classical method includes using questionnaires to assess the potential risk of suicide and applying clinician-patient interactions [4]. Other means of gathering data was through suicidal notes i.e. previous approaches have examined suicide notes using content analysis [5], sentiment analysis [6, 7], and emotion detection [8].

However, with the boom of internet and many social network websites – researchers started capturing user-produced content, available online and applied many machine learning method especially supervised learning techniques for their research in this field. Some studies report a positive correlation between suicide rates and the volume of social media posts that may be related to suicidal ideation and intent [9, 10]. Most of these studies have been carried out using features such as N-gram feature, syntactic feature and knowledge-based features, context features and class-specific features [6]. Apart from this, features such as word embedding [11] and sentence embedding [12] are seen to be used in many researches.

Notable recent work in this field of study has been done by researchers like Cash et al. [13], Shepherd et al. [14], and Jashinsky et al. [10] who performed data analysis over social networking websites such as Twitter and MySpace for content suggesting suicidal tendencies. O'Dea et al. developed automatic suicide detection on Twitter by applying logistic regression and SVM on TF-IDF features [15]. De Choudhury et al. contributed by working on suicidal post available on Reddit Website by providing an analysis on the effect of celebrity suicides on suicide-related content [16] and the transition from mental health illness to suicidal ideation [17].
In this project, we referred three research papers: Supervised Learning for Suicidal Ideation Detection in Online User Content [2], Machine Classification and Analysis of Suicide-Related Communication on Twitter [18], Prediction of Suicidal Ideation in Twitter Data using Machine Learning algorithms [19].

III. PROPOSED MODEL/ FRAMEWORK

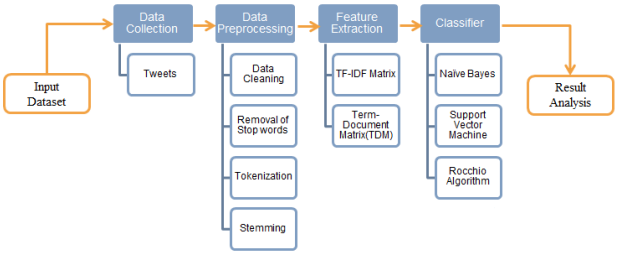

Figure 1: Proposed Framework

The framework for the application is divided into the following parts:

Data collection: In order to analyse suicidal conversations posted on internet, we selected Twitter, a popular social media platform, where people usually post their emotions. For that we first identified the set of terms that were likely to identify suicidal communication within text. We collected the tweets using Twitter4j Api, that were either dedicated to discussion of suicidal thoughts and feelings or containing a large and easily identifiable body of such material. Human annotators were asked to identify content containing suicidal thoughts and feelings and mark label as '-1' for suicidal post and '1' for non-suicidal post.

Our dataset consists of text id, tweets collected from twitter and sentiment label (Figure-2).

	Id	Content	Sentiment
207	9.200000e+...	I know in the end I will be left again because myself is not important at all	-1
171	9.200000e+...	RT @deepee_yo: It's all fun & games until you're throwing up hotcheetos https://t.co/WFnQHu3ieA	1
166	9.200000e+...	I have amazing people in my life that encourage nothing except the positive. Thank you Lord for blessing me.	1
81	4.310000e+...	I'm hopeless and awkward and desperate.	-1
22	4.300000e+...	RT @gokaimomurda408: Sometimes ifeel like my family would be better off without me.	-1
283	9.370000e+...	plz kill me	-1
91	4.310000e+...	I'm such an outcast in my family:	-1
57	4.310000e+...	So heavy hearted today.#ripmom #ripryan ??????	-1
275	9.200000e+...	You need to LOVE YOURSELF	1

Figure 2: Raw Dataset

Data Pre-processing: The most important part of our application is the data preprocessing stage, it is mainly divided into following parts for our algorithm:

- **Data cleaning:** Data that we collected from online sources was in the form of raw tweets. Generally, tweets contains various hash tags, emoticons, special characters, stop words, numbers, html links etc. So it was imperative to clean the tweets of the non-required data before actually using them for further processing. We used a collection of regular expressions to clean the tweets of each of the above mentioned dirt.
  Below are the examples of a few:
  a) To remove retweet entities –
  '(RT|via)((?:\\b\\W*@\\w+)+)'
  b) To remove punctuations - [[:punct:]]'
  c) To remove emoticons - '<.*>'

- **Data Transformation:** Cleaned tweets cannot directly be used for classification. This step includes the creation of our features from raw cleaned tweets by tokenization of tweet data. There are two types of feature set, we will create and test for classification.

  a) **Frequency matrix:** The tokenized words are used as features against the documents(tweets). Frequency of occurrence of each token in the given document is stamped as the value of the token for that document. This gives us a discrete feature set of tokens.

courag	cut	d	day	dead	depress	die
0	1	0	0	1	1	1
1	0	0	1	0	1	0
0	1	0	0	2	0	1
1	0	0	2	0	1	0
0	0	0	0	0	2	2
1	1	0	1	1	1	0
0	0	0	0	0	1	0
0	2	0	2	2	3	1
0	0	0	0	0	1	0

Figure 3: Frequency Matrix

  b) **TF-IDF matrix:** On the tokenized feature set, TF-IDF(Term frequency - Inverse Document Frequency) algorithm is applied to get the values for respective feature sets and documents.

better	co	cut	d	day	dead	depress	die
0	0	0	0	0	0	0.386408	0.361168
0	0	0.350232	0	0	0	0	0.270876
0	0	0	0	0	0	0	0.135438
0	0.111591	0	0	0	0	0	0.120389
0	0	0	0	0	0	0.165603	0.154786
0.319331	0	0	0	0	0	0	0
0.319331	0	0	0	0	0	0	0
0.255465	0	0	0	0	0	0	0

Figure 4: TF-IDF matrix

TF-IDF factor is calculated for all features as below:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)
IDF(t) = log_e(Total number of documents / Number of documents with term t in it).
TF-IDF(t) score = TF(t) * IDF(t)
We experimented with both the feature sets for all the algorithms.

- **Feature Selection:** This is the final step of data preprocessing. It includes removal of unwanted features from the feature sets. The selection of unwanted features was done on the basis of whether the occurrence of the token happened in more than a threshold percentage of documents, as the feature is relevant only if it occurred in a sufficiently high number of documents. The sparsity threshold factor was set to 1%.

Classification algorithms: We used the two derived features sets with the most popular classifiers, Decision Trees (DT), Naive Bayes (NB), Support Vector Machine (SVM) and Nearest Neighbor (Rocchio).

Nearest Neighbor (Rocchio) Algorithm

This algorithm is based on the K-nearest neighbor algorithm. Based on the lazy learner algorithm, it performs well on non-linear data. The graphical representation of Nearest neighbor makes it easy to understand and interpret.
It is a slight deviation to the well-known K-nearest neighbor algorithm wherein the test point is classified to the majority of k-nearest training points to the test point. This approach may not be perfectly suitable for test classification for suicidal tweets considering the following example: life is a suicidal tweet may be something like "I want to end my life", and in the case of non-suicidal it can be "I love my life". So, considering the majority of nearest neighbors a suicidal tweet can get misclassified to the non-suicidal class.
So, Rocchio algorithm finds the mean for all the features or tokens for both the classes separately, say µ1 and µ2 with the training data.

While making prediction, it tries to classify the test point to the mean µ of the class the test point is closest to. The measure for distance used is Euclidean and represented by the following equation:

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$
$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

Where p and q are the two data points taken from the training data and d(p,q) represents the distance between these two points.

Naïve Bayes(Maximum a Posteriori)

Goal : Build a classifier model based on Naive Bayes (which is formulated on the Bayes Theorem). In this algorithm we calculate :

- Likelihood of each feature v/s class attribute

- Class Probability

These components are used to compute Maximum posterior probability of all the target classes given the test feature vector and we select the max posterior probability to predict the classes of the test dataset.

For building the classifier model:
Input: Feature Set (TF-IDF/Frequency Matrix)
Output: Feature's likelihood and class probabilities Matrix i,e Classifier

Algorithm:

1. We Construct the frequency table for every attribute against the class attributes.

2. We compute the column wise sum of this frequency table, which gives the total count of each class.

3. We transform this frequency table to likelihood table i.e. we divide the frequencies (obtained in step 1) by the column wise sum (Step2) and bind the likelihood table with their respective column index.

Step 1 to Step 3 were performed for all the attributes of the dataset and the result of each attribute is bound together to form a matrix of likelihood.

4. Then we compute the class probabilities and update the length of the data structure holding these class probabilities to match the length of the data structure of the likelihood table by adding -2 as last column value of class probabilities and bind the likelihood matrix and class probability together.

The resultant matrix is used to predict the class of the test dataset.

To predict the class of test dataset:

Input: A row of test dataset and Classifier (Feature's likelihood and class probabilities Matrix)
Output: Predicted Class of the test row.
Algorithm:

1. We convert any non-integer value in the index column of the Classifier to integer value.

2. Initialized the predicted class as the Maximum of the prior class probabilities, so that if a test row with features that the classifier has no knowledge about can be predicted as the maximum of the prior probabilities.

3. We then iterate over the test_row to check if any feature value of the test row is missing from learned classifier - if any feature is found to be missing, that feature column is removed from the test row.

4. Then we fetch all the rows from classifier that have same value under the column: “Rownames” as that of the feature vector of the test row.

5. After this, we selected the likelihood probabilities for a specific value under a particular column by matching the column index and column value with the test row's feature value and feature's (column) index.

This provided us with all the likelihood of the test row's feature vector against all the unique classes of the dataset.

To this result we bind the class probabilities ensuring we do not have 0 probability for any of classes by updating that value with 0.0001.

6. Then we performed a column wise multiplication to compute the posterior probabilities for all the classes of dataset and test row relevant features (i.e. Bayes Theorem).

7. Then we find the maximum of these posterior probability of all the classes and set the name of that class as the predicted class.

Support Vector Machine (SVM)

A Support Vector Machine is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side[21].

We implemented Linear SVM, for which learning of hyperplane is done by transforming our problem using linear algebra. This is where kernel comes into the picture. We used two types of kernels:

Kernel function is given by,

$$K(x, z) = \varphi(x), \varphi(z)$$

Where x, z are points taken from the training data.

For linear kernel, $\varphi(x) = (x_i . x_j)$ and for polynomial kernel: $\varphi(x) = (x_i . x_j)^d$

Where d = degree of polynomial (in our case d = 2), C= constant. x_i is the input vector and x_j represents the support vector.

The goal of the classifier is to first find the decision boundary between the two classes, y = (-1,1) (indicating a binary response) that is maximally far from any point in the training data(considering some points to be outliers or noise). Then Using aforementioned kernel methods, SVM classifier finds two hyperplanes that separates the two classes and distance between them is set as large as possible so as to find a maximum margin hyperplane (that lies midway between them). The distance is computed by using **distance from point to plane** equation and to prevent the points from falling inside the margin we added the condition that for all $1 \leq i \leq n$, $y_i(w^t x_i + b) \geq 1$ so that all points lie on the correct side of the margin.

For optimization, we used Stochastic sub-gradient descent given by,

$$J^t(w) = \frac{1}{2} w^t w + C\ max(0, 1 - y_i(w^t x_i))$$

Where $J^t(w)$ is a gradient function of $\vec{w}$, for defined number of steps per epoch(i.e., 1000), the training data is shuffled randomly at start of each epoch and for each training example gradient function is computed. A step is taken in the direction of the vector selected from the gradient function if $y_i(w^t x_i) \geq 1$ else direction is changed when condition is not satisfied and $J^t(w)$ is equal to $w - C y_i x_i$.

Decision Tree

Goal: Build a classifier model based using CART algorithm. In CART Gini index is used to measure the impurity and evaluate the spilts over the dataset.

Below Equation were used -

Gini Index:

Gini = 1-P^2(Target=-1)—P^2(Target=1)

Target = -1 (indicates suicidal)

Target = 1 (indicated non-suicidal)

P - Probability of the class

Information Gain:

Two branches were build after a split on a node - left and right

$$P = \frac{(\text{num\_of\_rows(left)})}{(\text{num\_of\_rows(left)} + \text{num\_of\_rows(right)})}$$

Information Gain

$$= (\text{gini(dataset\_in\_current\_node)} - \text{p} * \text{gini(left)} - (1 - \text{p}) * \text{gini(right)})$$

The information gain is computed using uncertainty of the node, minus the weighted impurity (p) of two child node i.e left and right branch.

To build the classifier model :

Input: Feature Set (TF-IDF; Frequency Matrix)

Output: Decision Tree

Algorithm:

1. The root node of the Tree was provided with the entire dataset.

2. We then computed the gini index of the dataset.

3. Then we performed below task for all the feature/attribute's unique values :

2.1 Partitioned the dataset into two branches – (criteria used - for a column and a given value in consideration, are the values in that particular column less than the value in question?)
2.2 Then we compute the information gain on each split and find the maximum information gain spilt value for a particular column ensuring we have the atleast one row in both the branch.

This step gives the best split value for every feature in the dataset.

3. Then we find the feature with the best split and use it to split the dataset at the tree node

4. Steps 2 and 3 were repeated until the stopping criteria's were met:

4.1 For a give node the information_gain > 0 and number of rows in that node should be a minimum of 20.

4.2 For a node if the number of rows in either of the two branch is less than 10.

Predicting the class of test dataset:

Input: Root of the Decision Tree and a row of the test dataset

Output: Predicted Class of the test row.

Algorithm:

1. Traversed over the decision Tree using Pre-order tree traversal- we compared the best split criteria value at each node with the corresponding feature's value of the test_row and if the test_row value was found to be less then we moved onto to left branch otherwise we traversed the right branch of that respective node.

2. Step1 was followed till we reached a leaf node and then we computed the frequency table of classes at that leaf node and returned the class with the maximum result as predicted class.

IV. EXPERIMENTS AND RESULTS

We used k-fold cross-validation technique to calculate and compare the accuracies of various classifiers.

Subsequently, confusion matrix is generated and performance indicators like precision, recall, specificity and F-measure are calculated to demonstrate the results for both Term Frequency- Inverse Document Frequency Matrix And Frequency Matrix (FM).

Nearest Neighbor (Rocchio) Algorithm

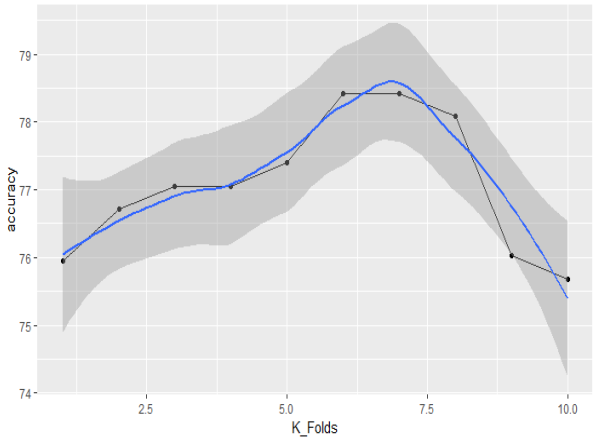

Figure 5: TF-IDF 10-fold Accuracy

	Suicide	Non-suicide
Suicide	99	17
Non-suicide	49	127

Figure 6 Confusion Matrix – Rocchio (TF-IDF)

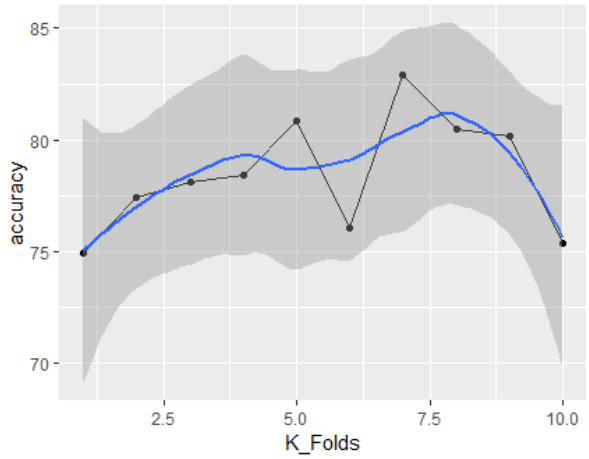

Figure 7 FM: 10-fold Accuracy

	Suicide	Non-suicide
Suicide	97	17
Non-suicide	51	127

Figure 8 Confusion matrix: Rocchio (FM)

Naïve Bayes Algorithm

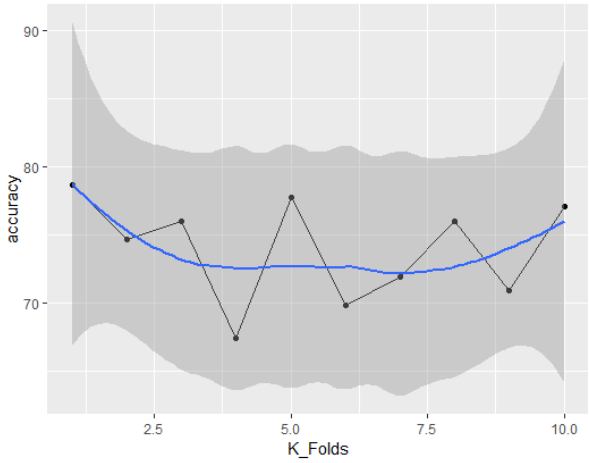

Figure 9 TF-IDF 10-fold Accuracy

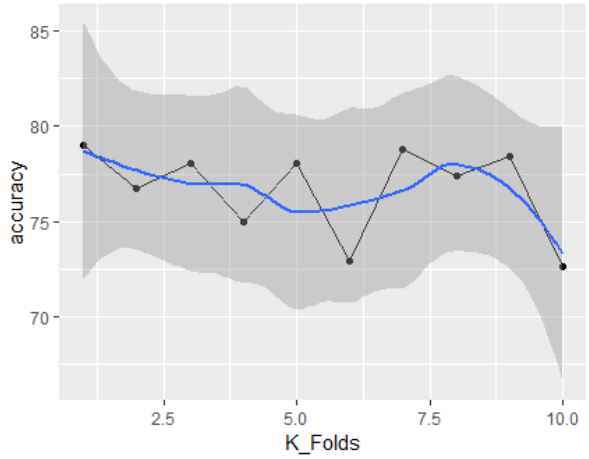

Figure 10 FM:10-fold Accuracy

	Suicide	Non-suicide
Suicide	135	48
Non-suicide	19	90

Figure 11 Confusion Matrix - Naive Bayes(TF-IDF)

	Suicide	Non-suicide
Suicide	116	45
Non-suicide	32	99

Figure 12 Confusion Matrix: Naive Bayes(FM)

Support Vector Machine(SVM)

For SVM, dataset is partitioned into three sets; 80% is training data, 10% is taken for testing and another 10% is used for validation set. For optimization of algorithm with stochastic gradient descent, we are using testing data for training purposes as well. That is why as compared to other classifiers SVM has less variables for confusion matrix. Given below are the confusion matrix for both TF-IDF and frequency matrix(FM):

	Suicide	Non-suicide
Suicide	90	10
Non-suicide	56	136

Figure 13 Confusion Matrix - Linear Kernel (TFIDF)

	Suicide	Non-suicide
Suicide	82	6
Non-suicide	64	140

Figure 14 Confusion Matrix-Polynomial Kernel(TF-IDF)

	Suicide	Non-suicide
Suicide	94	11
Non-suicide	64	123

Figure 15 Confusion matrix-Linear Kernel(FM)

	Suicide	Non-suicide
Suicide	92	12
Non-suicide	66	122

Figure 16 Confusion matrix-Poly Kernel(FM)

Decision Tree

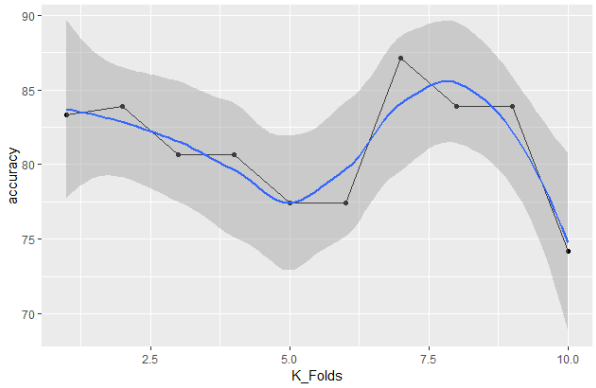

Figure 17 TF-IDF: 10-Fold Accuracy

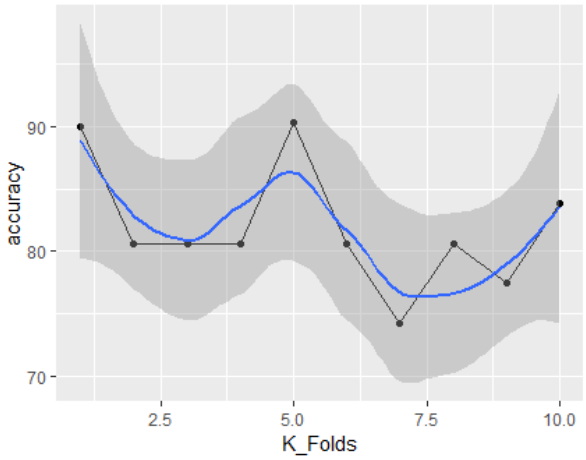

Figure 18 FM: 10-Fold Accuracy

	Suicide	Non-suicide
Suicide	164	49
Non-suicide	5	74

Figure 19 Confusion Matrix: Decision Tree(TF-IDF)

	Suicide	Non-suicide
Suicide	164	46
Non-suicide	7	75

Figure 20 Confusion matrix: Decision Tree(FM)

Result Analysis-

	SVM - linear	SVM - polynomial	Naïve Bayes	Decision Tree	Rocchio
Accuracy	80.13699	79.10959	74.033799	81.236558	77.0808
Precision	85.82677	90.74074	73.77049	76.54867	85.28139
Recall	73.15436	65.77181	87.66234	97.19101	66.37466
Specificity	87.41259	93.00699	65.21739	59.54198	88.15331
F-Measure	78.98551	76.26459	80.11869	85.64356	74.64949

Figure 21 Performance Evaluation: TF-IDF

	SVM - linear	SVM - polynomial	Naïve Bayes	Decision Tree	Rocchio
Accuracy	74.31507	73.28767	73.449843	81.903225	76.42894
Precision	89.52381	88.46154	72.00497	77.92793	85.03521
Recall	59.49367	58.22785	78.16712	96.11111	65.09434
Specificity	91.79104	91.04478	68.57143	62.0155	88.15331
F-Measure	71.48289	70.22901	74.95961	86.06965	73.74046

Figure 22 Performance Evaluation: FM

- As we infer from the table above, Decision tree performed the best amongst all algorithms for both the datasets. It gave the highest values of Accuracy and Recall.
- Also, count of False-Negatives is less than False-Positives for Decision tree indicating lesser count of suicidal tweets classified as non-suicidal. This is actually the purpose of our classification algorithms, as we would appreciate a misclassification(Non suicidal tweets to be classified as suicidal) to be more than the misclassification(Suicidal to be classified as Non-suicidal).
- Accuracies for TF-IDF outperforms that for TDM feature set indicating that relative term frequency and Inverse document frequency is a better measure of similarities in texts.

Contribution and Discussion- The research papers we citited, describes the framework which consists of first collecting the data (suspected tweets) from Twitter, which is associated with the vocabulary used by the sucidal social media users, to convey their feelings and then doing prediction and evaluation of the analysed resultd. The text obtained from the data collected was classified into one of seven suicide-related classes categorized by a crowd sourced team of human annotators (stated in the table shown below).

class	description
c1	Evidence of possible suicidal intent
c2	Campaigning (i.e. petitions etc.)
c3	Flippant reference to suicide
c4	Information or support
c5	Memorial or condolence
c6	Reporting of suicide (not bombing)
c7	None of the above

They had used various machine learning classifiers to identify the language used for suicide ideation on Twitter by doing predictive analysis of features collected from the dataset. Also, they built an ensemble classifier using Rotation forest machine learning algorithm which outperformed the results obtained by the baseline classifiers, achieving a F-measure of 0.69 for suicidal ideation class [18]. In addition to this, the paper focuses on understanding and detecting the suicidal thoughts in online user content along with extracting the features effectively. [2]

The only difference that can be found in our paper compared to the sources we referred, is in terms of the type of dataset taken and data mining algorithms used – enabling us to understand the applications of machine learning and data mining in real world problems.

The methods used might need further development, ideally with a larger sample of social media postings, and application to platforms. We note that it is important to retain collaboration with domain experts in suicidology throughout the experimental and interpretation phases of future research to improve classification accuracy by incorporating prior knowledge of the characteristics of suicidal language [18].

V. CONCLUSION

The amount of information from posts on social media will keep growing. Online social network has so much to explore and with proper research and tools, we can extract useful patterns which can be beneficial for communities.

In this paper, we investigated the problem of suicidality detection in online user-generated content. By collecting and analysing the anonymous online data from an active social media platform- Twitter, we provide rich knowledge that can complement the understanding of suicidal ideation and behaviour. Though applying feature processing and classification methods to our carefully built datasets, we evaluated, analysed, and demonstrated that our framework can achieve high performance (accuracy) in distinguishing suicidal thoughts out of normal posts in online user content.

While exploiting more effective feature sets, complex models or other factors such as n-grams, temporal information may improve the detection of suicidal ideation—these will be our future directions; the contribution and impact of this paper are threefold: (1) delivering rich knowledge in understanding suicidal ideation, (2) introducing datasets for the research community to study this significant problem, and (3) proposing informative features and effective models for suicidal ideation detection.

VI. REFERENCES

[1] “Suicide data” WHO report 2016 - http://www.who.int/mental_health/prevention/suicide/suicideprevent/en/

[2] Supervised Learning for Suicidal Ideation Detection in Online User Content, Shaoxiong Ji, Celina Ping Yu, Sai-fu Fung, Shirui Pan, and Guodong Long, “Supervised Learning for Suicidal Ideation Detection in Online User Content,” Complexity, vol. 2018, Article ID 6157249,

pages,2018. https://doi.org/10.1155/2018/6157249

[3] R. C. O'Connor and M. K. Nock, "The psychology of suicidal behavior," The Lancet Psychiatry, vol. 1, no. 1, pp. 73–85, 2014.

[4] W. C. Chiang, P. H. Cheng, M. J. Su, H. S. Chen, S. W. Wu, and J. K. Lin, "Socio-health with personal mental health records: suicidal-tendency observation system on Facebook for Taiwanese adolescents and young adults," in 2011 IEEE 13th International Conference on e-Health Networking, Applications and Services, pp. 46–51, Columbia, MO, USA, 2011, IEEE.

[5] J. Pestian, H. Nasrallah, P. Matykiewicz, A. Bennett, and A. Leenaars, "Suicide note classification using natural language processing: a content analysis," Biomedical Informatics Insights, vol. 3, pp. 19–28, 2010.

[6] W. Wang, L. Chen, M. Tan, S. Wang, and A. P. Sheth," Discovering fine-grained sentiment in suicide notes," Biomedical Informatics Insights, vol. 5, Supplement 1, pp. 137–145, 2012.

[7] J. P. Pestian, P. Matykiewicz, M. Linn-Gust et al., "Sentiment analysis of suicide notes: a shared task," Biomedical Informatics Insights, vol. 5, Supplement 1, pp. 3–16, 2012.

[8] M. Liakata, J. H. Kim, S. Saha, J. Hastings, and D. Rebholz-Schuhmann, "Three hybrid classifiers for the detection of emotions in suicide notes," Biomedical Informatics Insights, vol. 5,Supplement 1, 2012.

[9] H.-H. Won, W. Myung, G.-Y. Song,W.-H. Lee, J.-W.Kim, B. J. Carroll, and D. K. Kim. Predicting national suicide numbers with social media data. PloS one, 8(4):e61809, 2013.

[10] J. Jashinsky, S. H. Burton, C. L. Hanson, J. West, C. Giraud-Carrier, M. D. Barnes, and T. Argyle. Tracking suicide risk factors through twitter in the US. 2013.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/abs/1301.3781.

[12] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang,"DiSAN: directional self-attention network for RNN/CNNfree language understanding," 2017, https://arxiv.org/abs/1709.04696

[13] S. J. Cash, M. Thelwall, S. N. Peck, J. Z. Ferrell, and J. A. Bridge, "Adolescent suicide statements on MySpace," Cyberpsychology, Behavior, and Social Networking, vol. 16, no. 3, pp. 166–174, 2013.

[14] A. Shepherd, C. Sanders, M. Doyle, and J. Shaw, "Using social media for support and feedback by mental health service users: thematic analysis of a Twitter conversation," BMC Psychiatry, vol. 15, no. 1, p. 29, 2015.

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

[16] M. Kumar, M. Dredze, G. Coppersmith, and M. De Choudhury, "Detecting changes in suicide content manifested in social media following celebrity suicides," in Proceedings of the 26th ACM Conference on Hypertext & Social Media - HT '15,pp. 85–94, Guzelyurt, Northern Cyprus, 2015, ACM.

[17] M. De Choudhury, E. Kiciman, M. Dredze, G. Coppersmith, and M. Kumar, “Discovering shifts to suicidal ideation from mental health content in social media,” in Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems- CHI '16, pp. 2098–2110, San Jose, California, USA, 2016, ACM.

[18] P. Burnap, W. Colombo, and J. Scourfield. Machine classification and analysis of suicide-related communication on Twitter. In Proceedings of the 26th ACM Conference on Hypertext & Social Media, pages 75–84. ACM, 2015

[19] Birjali, M., Beni-hssane, A., & MohammedErritali (2016). Prediction of Suicidal Ideation in Twitter Data using Machine Learning algorithms.

[20] http://sentiwordnet.isti.cnr.it

[21] https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72