# SNAP-BATNET: Cascading Author Profiling and Social Network Graphs for Suicide Ideation Detection on Social Media

**Rohan Mishra***
Delhi Technological University
rohan.mishra1997@gmail.com

**Pradyumna Prakhar Sinha***
Delhi Technological University
pradyumna_bt2k15@dtu.ac.in

**Ramit Sawhney**
Netaji Subhas Institute of Technology
ramits.co@nsit.net.in

**Debanjan Mahata**
Bloomberg
dmahata@bloomberg.net

**Puneet Mathur**
MIDAS, IIIT-Delhi
pmathur3k6@gmail.com

**Rajiv Ratn Shah**
MIDAS, IIIT-Delhi
rajivratn@iiitd.ac.in

## Abstract

Suicide is a leading cause of death among youth, and the use of social media to detect suicidal ideation is an active line of research. While it has been established that these users share a common set of properties, the current state-of-the-art approaches utilize only text-based (stylistic and semantic) cues. We contend that the use of information from networks in the form of condensed social graph embeddings and author profiling using features from historical data can be combined with an existing set of features to improve the performance. To that end, we experiment on a manually annotated dataset of tweets created using a three-phase strategy and propose SNAP-BATNET, a deep learning based model to extract text-based features and a novel Feature Stacking approach to combine other community-based information such as historical author profiling and graph embeddings that outperform the current state-of-the-art. We conduct a comprehensive quantitative analysis with baselines, both generic and specific, that presents the case for SNAP-BATNET, along with an error analysis that highlights the limitations and challenges faced paving the way to the future of AI-based suicide ideation detection.

## 1 Introduction

Suicide is among the top three causes of death among youth worldwide. According to a WHO report[1], almost one million people die from suicide annually and 20 times more people attempt suicide. Therefore, suicide causes a global mortality rate of 16 per 100,000, and there is one attempt every 3 seconds on average (Radhakrishnan and Andrade, 2012). Moreover, the effect of it on friends and family members are often devastating (E. Clark and D. Goldney, 2000). What compounds the issue is that while it is preventable and, early detection is crucial in effective treatment, there is a lot of social stigma related to it which prevents people from disclosing their thoughts and seeking professional help. It has been found that people suffering from suicidal ideation make use of social media networks to share information about their mental health online (Park et al., 2012) with many having disclosed their suicidal thoughts and plans (Prieto et al., 2014). Therefore it is a pressing issue to be able to utilize the signals available on social media in order to identify individuals who suffer from *suicide ideation* in an automated manner and offer them the required help and treatment.

There exists an active field of research in the field of suicidal ideation detection (O'Dea et al., 2015; Sawhney et al., 2018a) that are able to extract meaningful patterns of behavior from users of social media in order to predict suicidal behavior. These have utilized the information presented in the text of the posts that were shared and utilized both traditional as well as deep learning methods. A rich body of literature exists to show the influence of social interactions of at-risk individuals for their effective detection and treatment. However, to the best of our knowledge, no advances have been made to include information from social engagement, ego networks and other user attributes

* Denotes equal contribution.

[1] https://www.who.int/mental_health/prevention/suicide/suicideprevent/

*Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 147–156
Minneapolis, Minnesota, June 3 - 5, 2019. ©2017 Association for Computational Linguistics

which we hypothesize would help us in being able to detect suicidal behavior better. Since the interaction of a person with their social surrounding in the form of author profiling from historical tweets and social graph based information can give us a plethora of information about their mental health (Luxton et al., 2012), we explore the usage of author profiling and other features to detect the presence of suicide ideation in tweets better.

Our contributions to the field are as follows -

1. Creation of a significantly large manually annotated dataset for detection of patterns in suicidal behavior in social media along with historical tweet data and social network graphs which will be made publicly available after anonymization keeping all ethical considerations in mind.

2. Proposing *SNAP-BATNET (Social Network Author Profiling - BiLSTM Attention NETwork)*, a feature stacking based architecture that uses novel handcrafted features: *author profiling, historical stylistic features, social network graph embeddings* and *tweet metadata* with an ablation study for validation.

3. Conducted an extensive quantitative comparison with several traditional and state-of-the-art baselines along with an in-depth error analysis to highlight the challenges faced.

## 2 Related Work

### 2.1 Suicidal Ideation Detection

There have been certain advances in the usage of social media to automatically detect cases of suicidal ideation in the past (Sawhney et al., 2018a; De Choudhury et al., 2013; Benton et al., 2017). Cavazos-Rehg et al. (2016) performed a content-based analysis on a small number of depression related tweets to derive certain qualitative insights into the behavior of users displaying suicidal behavior but did not propose any automated solution for the task of detection. Sawhney et al. (2018a) prepared a manually annotated dataset of tweets and proposed a set of features to be used to improve classifier performance but included only text-based features which limits the performance of the classifiers. De Choudhury et al. (2013) developed a crowd-sourced set of patients diagnosed with Major Depressive Disorder(MDD) and used their social media posting through the course of a year to establish a set of signals to help predict depression before its onset. Benton et al. (2017) utilized a novel multitask learning framework to predict atypical mental health conditions with a scope of predicting suicidal behavior but included only text-based features for their multi-task framework.

Furthermore, there have been several forays into tweet classification that utilize a similar set of signals for other applications such as detection of abuse, cyberbullying and hate speech (Mathur et al., 2018b), (Mathur et al., 2018a). Waseem and Hovy (2016) used a public dataset and used a collection of features to show the usefulness of gender-based and location-based information in improving the effectiveness of classifiers. Gambäck and Sikdar (2017) developed a CNN model that used both character n-grams and word2vec features in order to improve the classifier performance greatly. Badjatiya et al. (2017) made use of the same benchmarking dataset, provided a set of baselines and used a combination of randomly initialized embeddings along with LSTM and Gradient Boosting Decision Trees to achieve state of the art performance.

### 2.2 Author Profiling

The inclusion of author based information has been explored in some tasks related to natural language processing. Waseem and Hovy (2016) utilized gender and location-based information along with text-based features to achieve superior performance. Johannsen et al. (2015) used similar features for syntactic parsing. While it is accepted that such demographic features may improve performance, it is often not possible to extract such features from social media websites like Twitter since this information is often unavailable and unreliable. This has spawned an exciting line of research that makes use of a social graph of interaction between users to derive information about the user. Applications extract information about each user by representing each user as a node in a social graph and creating low dimensional representations usually induced by neural architecture (Grover and Leskovec, 2016; Qiu et al., 2018).

The application of such graph-based features overcomes the limitation caused by unavailability. Mishra et al. (2018); Qian et al. (2018) use such social graph based features to gain considerable improvement in the task of abuse detection. Tasks like sarcasm detection also gain improvement by using such features(Amir et al., 2016).

## 3 Data

The unavailability of a public dataset for performing benchmark tests, motivated us to develop our own dataset of considerable size in order to validate our hypothesis. We would like to make our dataset, lexicon, and embeddings public to the research community after making it anonymous and keeping all ethical considerations in mind to improve AI-based suicide prevention and analysis[2].

The dataset generation was a two-phase process: (i) A lexicon of suicidal phrases was generated (ii) Tweets were scraped using the lexicon and, historic tweets and social engagement data was gathered for each of the users.

### 3.1 Developing a Lexicon of Suicidal Phrases

In order to scrape tweets to create the dataset, a lexicon of phrases which could indicate suicidal ideation was created. The top posts, most of which are much larger than tweets, were scraped from three different forums which have an abundance of posts with suicidal ideation. These are *r/suicidalthoughts* [3] (top 100), *r/suicidewatch*[4](top 100) and *takethislife.com* [5] (top 200). Pytextrank [6] is a python module which implements a ranking model for text processing (Mihalcea and Tarau, 2004). This was used to rank and gather the list of the most prominent phrases from these posts. A manual filtering pass was also done to remove posts with little or no suicidal ideation information. The resulting list had 143 phrases such as *hit life, think suicide, wanting to die, suicide times, last day, feel pain point, alternate life, time to go, beautiful suicide, hate life*. Furthermore, the lexicon was extended by using the lexicon shared in (Sawhney et al., 2018a).

### 3.2 Data Collection

**Collecting tweets**: For each phrase in the curated lexicon, tweets were scraped using the Twitter REST API[7]. A total of 48,887 tweets were obtained. Furthermore, retweets and non-English language tweets were removed. A manual check was done to remove the tweets (around 3000) which were trivially non-suicidal. The final dataset has 34,306 tweets. Each tweet in the dataset is described by the fields given in Table 1.

| | |
|---|---|
| tweet_id | text |
| hashtags | user_mentions |
| user_id | retweet_count |
| favorite_count | |

Table 1: Dataset fields.

| **Graph Type** | **Edge Represents** | **Sparsity** $(10^{-5})$ | **Avg Degree** |
|---|---|---|---|
| quotes | A quoted B | 0.570 | 0.185 |
| mentions | A mentioned B | 2.780 | 0.905 |
| repliedTo | A replied to B | 1.755 | 0.571 |
| follower | A follows B | 1.587 | 0.516 |

Table 2: Graph comparisons(A and B represent users along an edge in the graph).

**Data for Author Profiling**: For the 34,306 tweets in the dataset, there are 32,558 unique users. For each of these users, the tweet timeline (previous 100 tweets or as many available) was scraped. Texts from historical tweets were combined for each of the users to generate the historical corpus for author profiling.

**Social Graphs**: The engagement between the users from the dataset was captured in the form of social graphs where the users were represented as vertices and edges denoted the relationships. Table 2 shows the different graphs constructed corresponding to four different relationships and also the statistical comparisons between them.

For the *Follower Graph*, follower lists were scraped for each of the users while for the other three graphs, tweets from the dataset and the historical collection were crawled through.

### 3.3 Data Annotation

Two annotators, who are students in clinical psychology adept in using social media on a daily basis, were provided with the guidelines to label the tweets as used in (Sawhney et al., 2018b). The guidelines were based on the following classification system -

1. **Suicidal intent present**

   - Posts where suicide plan and/or previous attempts are discussed.

[2] https://github.com/ramitsawhney27/NAACLSRW19Suicide

[3] https://www.reddit.com/r/suicidalthoughts/

[4] https://www.reddit.com/r/SuicideWatch/

[5] https://www.takethislife.com

[6] https://pypi.org/project/pytextrank/

[7] https://developer.twitter.com/en/docs.html

- Text conveys a serious display of suicidal ideation.
- Posts where suicide risk is not conditional unless some event is a clear risk factor eg:depression, bullying, etc.
- Tone of text is sombre and not flippant.

2. **Suicidal intent absent**

   - Posts emphasizing on suicide related news or information.
   - Posts containing no reasonable evidence that the risk of suicide is present; includes posts containing song lyrics, etc.
   - Condolences and awareness posts.

An acceptable Cohen's Kappa score was found between the two annotations (**0.72**). In cases of ambiguity in labeling or conflicts in merging, the default class 0 (non-suicidal) was assigned. The resulting dataset had 3984 suicidal tweets (12% of the entire dataset).

## 4 Methodology

The overall methodology is split into three phases: *preprocessing of data, extraction of features* and finally *evaluation of models and feature sets.*

### 4.1 Preprocessing

Due to the unstructured format of the text used in social media, a set of filters were employed to reduce the noise while not losing useful information.

1. A tweet-tokenizer was used [8] to parse the tweet and replace every username mentions, hashtags, and urls with <mention>, <hashtag>and <url>respectively.

2. The tokenized text then underwent stopword removal and was used as an input to WordNet Lemmatizer provided by nltk(Bird and Loper, 2004).

3. Using Lancaster Stemmer, provided by nltk(Bird and Loper, 2004) stemmed text was also generated to be used as inputs for some feature extraction methods.

### 4.2 Feature Extraction

The features extracted from the data set can be broadly classified into four types: *Text-based features, tweet metadata features, User Historical tweets features and Social Graph-based features.*

**Text Based Features**

- *TF-IDF*: Term Frequency-Inverse Document Frequency was used with the unigrams and bigrams from the stemmed text, using a total of 2000 features chosen by the tf-idf scores across the training dataset. The tf-idf scores were l2 normalized.

- *POS*: Parts of Speech counts for each lemmatized text using The Penn Tree Bank(Marcus et al., 1993) from the Averaged Perceptron Tagger in nltk is used to extract 34 features.

- *GloVe Embeddings*: The word embeddings for each word present in the pre-trained GloVe embeddings trained on Twitter (Pennington et al., 2014) were extracted, and for each tweet, the average of these is taken.

- *NRC Emotion*: The NRC Emotion Lexicon (Mohammad and Turney, 2013) is a publicly available lexicon that contains commonly occurring words along with their affect category (anger, fear, anticipation, trust, surprise, sadness, joy, or disgust) and two polarities (negative or positive). The score along these 10 features was computed for each tweet.

- *LDA*: Topic Modelling using the probability distribution over the most commonly occurring 100 topics was used as a feature for each tweet. LDA features were extracted by using scikit-learn's Latent Dirichlet Allocation module (Pedregosa et al., 2011). Only those tokens were considered which occurred at least 10 times in the entire corpus.

**Tweet Metadata Features**: The count of hashtags, mentions, URLs, and emojis along with the retweet count and favorite count of every tweet was extracted and used as a feature to gain information about the tweets response by the authors environment.

**User Historical tweets**: To gain information about the behavior of the author and their stylistic choices, a collection of their tweets were preprocessed, and stylistic and semantic features such as the averaged GloVe embeddings, NRC sentiment scores and Parts of Speech counts were extracted.

**Social Graph Features**: Grover and Leskovec (2016) describe an algorithm *node2vec* for converting nodes in a graph (weighted or unweighted) into feature representations.This method has been

[8] https://pypi.org/project/tweet-preprocessor/

employed by Mishra et al. (2018) in the task of abuse detection in tweets. *node2vec* vectors were generated for each of the graphs as introduced in Section 3.2.

## 5 Baselines

A set of baselines that reflect the current state-of-the-art approaches in short text classification were established. These include methods that use traditional learning algorithms as well as deep learning based models.

- **Character n-gram + Logistic Regression**: Character n-gram with Logistic Regression in the range (1,4) has often been used effectively for classification and works as a strong baseline (Waseem and Hovy, 2016; Badjatiya et al., 2017; Mishra et al., 2018).

- **Bag of Words + GloVe + GBDT**: A Bag of Words(BoW) corpus was generated with unigram and bigram features, the averaged pre-trained GloVe embeddings were then used on a Gradient Boosting Decision Tree which incrementally builds in stage-wise fashion. It is used as a baseline in (Badjatiya et al., 2017).

- **GloVe + CNN**: A CNN architecture inspired from (Kim, 2014; Badjatiya et al., 2017) was used with filter sizes (3,4,5).

- **GloVe + LSTM**: An LSTM with 50 cells was used along with dropout layers ($p = 0.25$ and $0.5$, preceding and following, respectively).

- **ELMo**: Tensorflow Hub [9] was used to get ELMo(Peters et al., 2018) embeddings which are known to have an excellent performance in several fields including sentiment analysis and text classification.

- **USE** - The Universal Sentence Encoder (Cer et al., 2018) encodes text into high dimensional vectors that can be used for tasks like text classification, semantic similarity, and clustering. Tensorflow Hub was used to get sentence encoding. Each tweet was converted encoded onto a dense 512 feature space.

- **Sawhney C-LSTM**: We replicated the C-LSTM architecture used in (Sawhney et al., 2018a) which uses CNN to capture local features of phrases and RNN to capture global and temporal sentence semantics.

- **R-CNN**: Recurrent Convolutional Neural Networks as proposed by (Lai et al., 2015) make use of a recurrent structure to capture contextual information as far as possible when learning word representations.

## 6 Methodology: SNAP-BATNET

### 6.1 Graph Embeddings

As discussed in the previous sections, social graphs were constructed for author profiling which could capture demographic features and improve the performance of the classifier. Four such weighted and undirected graphs were constructed: *Follower Graph, Mentions Graph, RepliedTo Graph* and *Quotes Graph*. All the self-loops were removed from the graphs, as they do not contribute to suicide-related communication features.

To obtain the author profiles, the nodes in the graphs were converted into feature representation using *node2vec* (Grover and Leskovec, 2016). *node2vec* works on the lines of word2vec and determines the context of the nodes by looking into their neighborhoods in the graph. It constructs a fixed number of random walks of constant length for each of the nodes to define the neighborhood of the nodes. The random walks are governed by the parameters *p (return parameter)* and *q (in-out parameter)* which have the ability to fluctuate the sampling between a depth-first strategy and a breadth-first strategy.

*node2vec* by itself does not generate embeddings for solitary nodes which comprised about 2/3rd of the total nodes. As per the empirical rule of normal distribution, 99.73% of the values lie within three standard deviations of the mean. To isolate the solitary nodes from the remaining ones, a random vector was generated three standard deviations away from the mean and was assigned to them.

Embeddings were generated for both weighted and unweighted graphs and were individually studied for the classification task. The number of dimensions and the number of epochs was set to 200 and 10 respectively. A stratified 5-fold grid search was carried out on the hyperparameters - *p, q, walk-length, window-size*. It was found that the default values for $p$ and $q$(1 and 1) along with

[9] https://tfhub.dev/

| Combination | F1 | AP |
|---|---|---|
| Follower+Mentions (CG) | 0.808 | 0.203 |
| Follower+RepliedTo (CG) | 0.806 | 0.196 |
| Mentions+RepliedTo (CG) | 0.803 | 0.197 |
| Follower+Mentions + RepliedTo (CG) | 0.807 | 0.201 |
| Follower+Mentions + RepliedTo (CE) | **0.849** | **0.268** |

Table 3: Graph combination results(CG-Combining graphs, CE-Combining embeddings) with weighted F1 and area under precision recall curve.

the combination of walk length 10 and window-size 5 performed best. This performance of short walks can be attributed to the sparse nature of the graphs. It was determined that unweighted graphs performed better and were used for generating combined social graph embeddings.

**Combining Graph Embeddings**: *Quotes Graph* was discarded from any further study owing to its individual performance in contrast with the other graphs. Its poor performance can be attributed to its statistics as given in the table 2. The rest of the graphs were combined followed two methods: by combining graphs or by combining embeddings using a deep learning approach. The resulting embeddings were trained using a Balanced Random Forest classifier. These results are shown in Table 3.

For generating these combined embeddings, a deep learning model as shown in Figure 1 was designed to be trained on the dataset. After the training, the concatenation layer was picked up as the embedding for the combination, and this was generated for all the users. These embeddings were then used in an LR classifier and a balanced random forest classifier. The results from the balanced random forest classifier were superior and were further used for feature stacking as mentioned in Section 6.2. *SNAP-BATNET* uses Follower, Mention and RepliedTo embeddings combined using the deep learning approach to generate social graph based features.

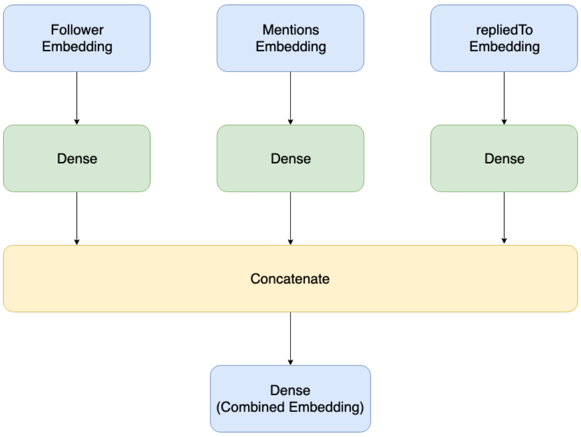

Figure 1: Combining graph embeddings.

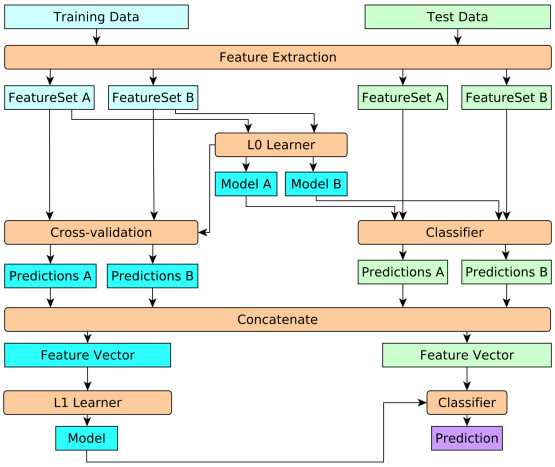

Figure 2: Feature Stacking: A meta-learning approach (figure taken from (Lui, 2012)).

## 6.2 Feature Stacking

The competing systems make use of the text based features for classification. To leverage the availability of different kinds of information in form of tweet metadata, historical author profiling and social graph based embeddings so as to overcome the unavailability of a predefined lexico-semantic pattern in the text, methods of combining information were explored. While tweet metadata is sparse, social graph based embeddings are dense in nature.

Initially, concatenation was used, and several models were tried by performing ablation studies. It was observed that the performance of the classifiers did not change significantly and in some cases deteriorated as features were concatenated. Therefore, it was reasoned that the feature sets should be combined in a way that would have the ability to join them related to their relative importance and also allow learning of non-linear relationships between them. Instead of using concatenation which proved to be ineffective or relative weighing, which is cumbersome, we used a meta learning approach inspired from (Lui, 2012).

One major difference between (Lui, 2012) and our approach is that while it uses Logistic Regression as weak learner for each feature set, different weak learners depending on the feature set or

| | **F1** | **AP** | **F1** | **AP** | **F1** | **AP** | **F1** | **AP** |
|---|---|---|---|---|---|---|---|---|
| **Model** | **Text Based** | | **+ Tweet Metadata** | | **+ Author Profiling** | | **+ Graph** | |
| FeatStackLR | .891 | .641 | .893 | .640 | .894 | .643 | .896 | .671 |
| Char ngram+ LR | .892 | .646 | .910 | .653 | .912 | .647 | **.915** | **.679** |
| BoWV+GloVe+GBDT | .897 | .567 | .896 | .534 | .897 | .542 | .899 | .584 |
| GloVe+CNN | .908 | .619 | .910 | .619 | .910 | .623 | .913 | .644 |
| GloVe+LSTM | .908 | .612 | .906 | .613 | .907 | .617 | .910 | .642 |
| USE | .915 | .669 | .916 | .667 | .916 | .666 | .914 | .663 |
| ELMo | .913 | .650 | .894 | .629 | .909 | .629 | .911 | .623 |
| Sawhney-C-LSTM | .915 | .662 | .915 | .662 | .916 | .661 | .912 | .687 |
| RCNN | .921 | .704 | .919 | .705 | .920 | .706 | .923 | .726 |
| SNAP-BATNET | .923 | .709 | .925 | .707 | .925 | .708 | **.926** | **.726** |

Table 4: Results with weighted F1 and area under precision recall curve.

baseline models were employed in our approach. The weak learners were chosen by using grid search over { *Logistic Regression, Balanced Random Forest Classifier, SVM* }. For each of the baselines, features from tweet metadata, historical author profiling, and social graph embeddings were combined using Feature Stacking. Logistic Regression was used as L1 learner since stacked LR is theoretically closer to a neural network and can help introduce non-linearity between the features (Dreiseitl and Ohno-Machado, 2002).

Our model *SNAP-BATNET* uses feature stacking with different L0 learners to combine the feature sets pertaining to text-based information(BiLSTM+Attention), tweet metadata information(Logistic Regression), historical author profiling(Logistic Regression) and social graph embeddings(Balanced Random Forest Classifier). Furthermore, a simple architecture(*FeatStackLR*) is proposed that uses Logistic Regression as both L0 and L1 learners. An ablation study of the handcrafted feature sets was carried out using *FeatStackLR*, which is shown in Table 5. The addition of GloVe based embedding leads to an improvement in results as these embeddings encode semantic information that is missing from statistical features.

| **Features** | **F1** | **AP** |
|---|---|---|
| TF-IDF + EMB<br>+ POS + LDA + NRC | **0.891** | **0.641** |
| TF-IDF + EMB<br>+ POS + LDA | 0.891 | 0.640 |
| TF-IDF + EMB<br>+POS | 0.890 | 0.641 |
| TF-IDF + EMB | 0.890 | 0.641 |
| TF - IDF | 0.888 | 0.618 |

Table 5: Ablation study (measured using weighted F1 score and area under Precision-Recall curve).

## 7 Experiments and Results

### 7.1 Experimental Setup

All the experiments were conducted with a train-test split of 0.2. The hyperparameters for each learner were calculated by using a 5-fold stratified cross-validation grid search. The CNN and LSTM architectures used 200-dimensional GloVe embeddings pre-trained on Twitter corpus using the Adam optimizer and were run for 10 epochs. The models were implemented in Keras with a Tensorflow Backend. In CNN and LSTM models, 0.1 of the training data was held out as validation data to prevent the model from overfitting. Each baseline model uses Feature Stacking and is used as a L0 learner to extract text-based features to be combined with other feature sets such as tweet metadata, historical author profiling and finally social graph embeddings.

### 7.2 Results

Zhang and Luo (2018) describe the lacunae of reporting metrics such as micro F1, Precision or Recall provided in cases of highly imbalanced datasets such as Abuse Detection. In order to properly gauge the ability of a system to detect suicidal ideation from tweets, we report the F1, Precision and Recall scores on a per class basis in Table 4. The results in Table 6 include the weighted F1 Score along with the area under the

| Models and Classes | | Text Based | | | + Metadata | | | + Author Profiling | | | + Graph | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | P | R | F | P | R | F | P | R | F | P | R | F |
| FeatStackLR | 0 | .97 | .89 | .93 | .97 | .90 | .93 | .97 | .90 | .93 | .97 | .90 | .93 |
| | 1 | .47 | .75 | .57 | .47 | .74 | .58 | .47 | .76 | .58 | .48 | .77 | .59 |
| Char n-gram | 0 | .97 | .89 | .93 | .95 | .94 | .95 | .95 | .94 | .95 | .96 | .94 | .95 |
| +LR | 1 | .47 | .77 | .58 | .57 | .63 | .60 | .58 | .63 | .60 | **.59** | **.66** | **.62** |
| BoWV + GloVe | 0 | .92 | .98 | .95 | .94 | .94 | .94 | .94 | .94 | .94 | .95 | .94 | .94 |
| + GBDT | 1 | .71 | .32 | .44 | .52 | .53 | .52 | .52 | .53 | .53 | .52 | .58 | .55 |
| GloVe | 0 | .94 | .97 | .95 | .95 | .96 | .95 | .95 | .96 | .95 | .95 | .96 | .95 |
| +CNN | 1 | .64 | .47 | .55 | .60 | .55 | .57 | .61 | .54 | .57 | .62 | .56 | .59 |
| GloVe | 0 | .94 | .97 | .95 | .95 | .94 | .95 | .95 | .94 | .95 | .95 | .95 | .95 |
| +LSTM | 1 | .65 | .46 | .54 | .56 | .59 | .58 | .56 | .59 | .58 | .58 | .60 | .59 |
| Universal | 0 | .94 | .97 | .95 | .96 | .94 | .95 | .96 | .94 | .95 | .96 | .94 | .95 |
| Sentence Encoder | 1 | .65 | .54 | .59 | .58 | .68 | .63 | .59 | .66 | .63 | .57 | .69 | .62 |
| ELMo | 0 | .94 | .98 | .96 | .97 | .90 | .93 | .94 | .97 | .95 | .96 | .94 | .95 |
| | 1 | .70 | .46 | .56 | .48 | .73 | .58 | .64 | .48 | .55 | .57 | .64 | .60 |
| Sawhney- | 0 | .94 | .98 | .96 | .94 | .96 | .95 | .94 | .97 | .96 | .96 | .93 | .95 |
| C-LSTM | 1 | .70 | .48 | .57 | .64 | .56 | .59 | .67 | .53 | .59 | .55 | .72 | .63 |
| RCNN | 0 | .95 | .96 | .96 | .96 | .95 | .95 | .96 | .95 | .95 | .96 | .96 | .96 |
| | 1 | .65 | .62 | .63 | .62 | .66 | .64 | .61 | .66 | .64 | .64 | .66 | .65 |
| SNAP-BATNET | 0 | .95 | .97 | .96 | .95 | .97 | .96 | .95 | .97 | .96 | .95 | .96 | .96 |
| | 1 | .71 | .55 | .62 | .69 | .60 | .64 | .68 | .61 | .64 | **.68** | **.62** | **.65** |

Table 6: Results with precision, recall and F1 score on a per class basis.

Precision-Recall Curve.

It was observed that adding features such as social graph embeddings, historical author profiling, and tweet metadata led to a considerable improvement in the performance of the classifiers since each feature set encodes a different kind of information that gives the resulting models more adept at the task of classification. As per our hypothesis, the addition of social graph embeddings led to significant improvement in performance across all baseline models. There was an increase in the recall value which is desirable because the reduction of false negatives was more important than the reduction of false positives. Among the traditional classifiers, character n-gram with Logistic Regression performed the best. Moreover, the use of LSTMs in the model such as Sawhney-C-LSTM, RCNN, and SNAP-BATNET improved the classifier performance. This can be reasoned by the effectiveness of LSTM in capturing long term dependencies. Among all the deep learning models, SNAP-BATNET, when combined with all other feature sets using Feature Stacking, performed the best outperforming the current state-of-the-art, i.e., Sawhney-C-LSTM.

### 7.3 Error Analysis

Here we go through some examples posed challenges to highlight limitations and future scope.

1. **Subtle indication**: "*Death gives meaning to life*" contains subtle indications of suicidal behavior but caused ambiguity between annotators and was not detected by the model.

2. **Sarcasm**: "*I want to f**king kill myself lol xD* " is one of the several examples where the frivolity of the tweet couldn't be determined.

3. **Quotes and Lyrics**: "*Better off Dead S̃leeping With Sirens; I'm as mad, and I'm not going to take this anymore!*" are song lyrics and movie dialogues which the annotators were able to identify but the model could not as it lacked real-world knowledge.

## 8 Conclusion

This paper explores the use of information from the behavior of users on social media by using features such as text-based stylistic features in combination with historical tweets based profiling and social graph based embeddings. We develop a manually annotated dataset on detection of suicidal ideation in tweets, a set of handcrafted features were extracted which were utilized by a set of traditional and state of the art deep learning based models and a quantitative comparison was carried out which validated the hypothesis of the effectiveness of social graph based features and author profiling in suicidal behavior detection with our proposed SNAP-BATNET model, particularly in improving recall. An extensive error analysis and comparison with baselines presents the case for our methodology.

In the future, this work can be extended by exploiting multi-modalities in the data in the form of images, videos, and hyperlinks. Multi-modal approaches have extensively been used for various tasks like predicting social media popularity (Meghawat et al., 2018; Shah and Zimmermann, 2017). Another interesting aspect would be to adapt the pipeline described in this paper to different problems like identifying mentions of personal intake of medicine in social media (Mahata et al., 2018b,a).

## References

Silvio Amir, Byron C Wallace, Hao Lyu, and Paula Carvalho Mário J Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976*.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760. International World Wide Web Conferences Steering Committee.

Adrian Benton, Margaret Mitchell, and Dirk Hovy. 2017. Multi-task learning for mental health using social media text. *arXiv preprint arXiv:1712.03538*.

Steven Bird and Edward Loper. 2004. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics.

Patricia A Cavazos-Rehg, Melissa J Krauss, Shaina Sowles, Sarah Connolly, Carlos Rosas, Meghana Bharadwaj, and Laura J Bierut. 2016. A content analysis of depression-related tweets. *Computers in human behavior*, 54:351–357.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. 2013. Predicting depression via social media. *ICWSM*, 13:1–10.

Stephan Dreiseitl and Lucila Ohno-Machado. 2002. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, 35(5-6):352–359.

Sheila E. Clark and Robert D. Goldney. 2000. The impact of suicide on relatives and friends. *The international handbook of suicide and attempted suicide*, pages 467–484.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM.

Anders Johannsen, Dirk Hovy, and Anders Søgaard. 2015. Cross-lingual syntactic variation over age and gender. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 103–112.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pages 2267–2273.

Marco Lui. 2012. Feature stacking for sentence classification in evidence-based medicine. In *Proceedings of the Australasian Language Technology Association Workshop 2012*, pages 134–138.

David D Luxton, Jennifer D June, and Jonathan M Fairall. 2012. Social media and suicide: a public health perspective. *American journal of public health*, 102(S2):S195–S200.

Debanjan Mahata, Jasper Friedrichs, Rajiv Ratn Shah, and Jing Jiang. 2018a. Detecting personal intake of medicine from twitter. *IEEE Intelligent Systems*, 33(4):87–95.

Debanjan Mahata, Jasper Friedrichs, Rajiv Ratn Shah, et al. 2018b. # phramacovigilance-exploring deep learning techniques for identifying mentions of medication intake from twitter. *arXiv preprint arXiv:1805.06375*.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Puneet Mathur, Ramit Sawhney, Meghna Ayyar, and Rajiv Shah. 2018a. Did you offend me? classification of offensive tweets in hinglish language. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 138–148.

Puneet Mathur, Rajiv Shah, Ramit Sawhney, and Debanjan Mahata. 2018b. Detecting offensive tweets in hindi-english code-switched language. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 18–26.

Mayank Meghawat, Satyendra Yadav, Debanjan Mahata, Yifang Yin, Rajiv Ratn Shah, and Roger Zimmermann. 2018. A multimodal approach to predict social media popularity. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 190–195. IEEE.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2018. Author profiling for abuse detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1088–1098.

Saif M Mohammad and Peter D Turney. 2013. Nrc emotion lexicon. *National Research Council, Canada*.

Bridianne O'Dea, Stephen Wan, Philip J Batterham, Alison L Calear, Cecile Paris, and Helen Christensen. 2015. Detecting suicidality on twitter. *Internet Interventions*, 2(2):183–188.

Minsu Park, Chiyoung Cha, and Meeyoung Cha. 2012. Depressive moods of users portrayed in twitter. In *Proceedings of the ACM SIGKDD Workshop on healthcare informatics (HI-KDD)*, volume 2012, pages 1–8. ACM New York, NY.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Víctor M Prieto, Sergio Matos, Manuel Alvarez, Fidel Cacheda, and José Luís Oliveira. 2014. Twitter: a good place to detect health conditions. *PloS one*, 9(1):e86191.

Jing Qian, Mai ElSherief, Elizabeth M Belding, and William Yang Wang. 2018. Leveraging intra-user and inter-user representation learning for automated hate speech detection. *arXiv preprint arXiv:1804.03124*.

Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 459–467. ACM.

Rajiv Radhakrishnan and Chittaranjan Andrade. 2012. Suicide: an indian perspective. *Indian journal of psychiatry*, 54(4):304.

Ramit Sawhney, Prachi Manchanda, Puneet Mathur, Rajiv Shah, and Raj Singh. 2018a. Exploring and learning suicidal ideation connotations on social media with deep learning. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 167–175.

Ramit Sawhney, Prachi Manchanda, Raj Singh, and Swati Aggarwal. 2018b. A computational approach to feature extraction for identification of suicidal ideation in tweets. In *Proceedings of ACL 2018, Student Research Workshop*, pages 91–98.

Rajiv Shah and Roger Zimmermann. 2017. *Multimodal analysis of user-generated multimedia content*. Springer.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Ziqi Zhang and Lei Luo. 2018. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *arXiv preprint arXiv:1803.03662*.