International Journal on Perceptive and Cognitive Computing (IJPCC) Vol 9, Issue 2 (2023)
https://doi.org/10.31436/ijpcc.v9i2.396

Mental State Detection From Tweets By Machine Learning

Nabiul Farhan Nabil, Ashadullah Galib, Takumi Sase*

Dept. of Computer Science, KICT, International Islamic University Malaysia, 53100 Kuala Lumpur, Malaysia.

*Corresponding author: takumi@iium.edu.my
(Received: 3rd April 2023; Accepted: 20th May 2023; Published on-line: 28th July 2023)

Abstract— The world over, mental illness is a serious issue. Many people use the social media that may affect their mental health positively, but often result in negative sentiments. This research aims to determine an individual's mental state based on their social media behavior on Twitter. We analysed a dataset including 170000 real tweets by using natural language processing and machine learning techniques. Decision tree, support vector machine, and recurrent neural network (RNN) were used for classifying twitter users, to detect if they are in positive or negative mental state. These models were compared to determine which approach provides more accurate detection of a positive/negative mental state. Then, the RNN yielded the highest accuracy 0.76 among the models, with the precision, recall, and the F_1 score being 0.75, 0.74, and 0.75, respectively. The truncated singular value decomposition was also utilised to visualise the high-dimensional feature space of the data.

Keywords— Mental State, Machine Learning, Twitter, Artificial Intelligence

I. INTRODUCTION

Twitter, Facebook, and Instagram occupy 40%–45% of people's waking hours. 42% of Twitter users visit it regularly. Hence, there is a global concern about mental health, such as how social media can cause depression, anxiety, etc. Twitter is a big platform to express our feelings, how our day went, and what we've achieved or lost. Their mental states can be determined by their posts, and social media can help during mental breakdowns [1-2].

In recent years, machine learning and artificial intelligence (AI) techniques have been used to detect such a mental state from social data [3-8]. For example, the authors MR Islam et al. developed a method for depression detection from Facebook data by machine learning, and achieved high accuracy for classifying depression from other states [3]. The AI techniques, as represented by the deep learning have also been used to improve the classification [8-9]. A. Amanat et al [8]. has proposed a method for detecting depression from text data using recurrent neural network (RNN) with long short-term memory (LSTM) [8]. In conjunction with such machine learning techniques, it is necessary to apply natural language processing (NLP) for converting text data to numeric data, so that different sentiments can be well distributed in a numeric-valued feature space [10].

Although a lot of methods have been proposed for detecting a mental state from social data, it is still unclear what kind of NLP and machine learning approach is suitable for generating a feature space where different mental states are orderly distributed. In this study, we analyse a large dataset including 170000 real tweets, each of which has been labeled as positive or negative sentiment [11]. We aim to detect a person's mental state, positive or negative, by analysing their Twitter data through supervised and unsupervised machine learning, in conjunction with NLP. In the classification, we will compare the three models: decision tree (DT), support vector machine (SVM) [12], and RNN with LSTM [13], to determine which method is more suitable for detecting positive and negative sentiments from tweets. The findings of this research may contribute to a strategy for social media websites to assist those suffering from mental illness.

II. MATERIAL AND METHODS

Figure 1 presents the flow of our study. Data Collection, NLP Pre-processing, Classification, and Model Evaluation will be presented in turn. All the analyses were conducted using Python code.

A. Data Collection

A Twitter dataset was collected from the Kaggle, containing 1.6 million tweets each of which is labeled as 4 (positive sentiment) or 0 (negative sentiment) [11]. In this study, we analysed 170000 tweets ($n = 170000$) for the sake of computational efficiency. Note that the patterns like "@" were removed from every text on the dataset. Punctuations, numbers, special characters, suffixes, and words less than three alphabets were also removed from the texts for better classification. We denote the i-th tweet, after the removal of unnecessary characters, by a vector of words $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})^{\mathrm{T}}$, where m is the number of words in the i-th tweet and T denotes the transposition. Then, the Twitter dataset is represented by $X = (x_1, x_2, \dots, x_n)$.

https://doi.org/10.31436/ijpcc.v9i2.396

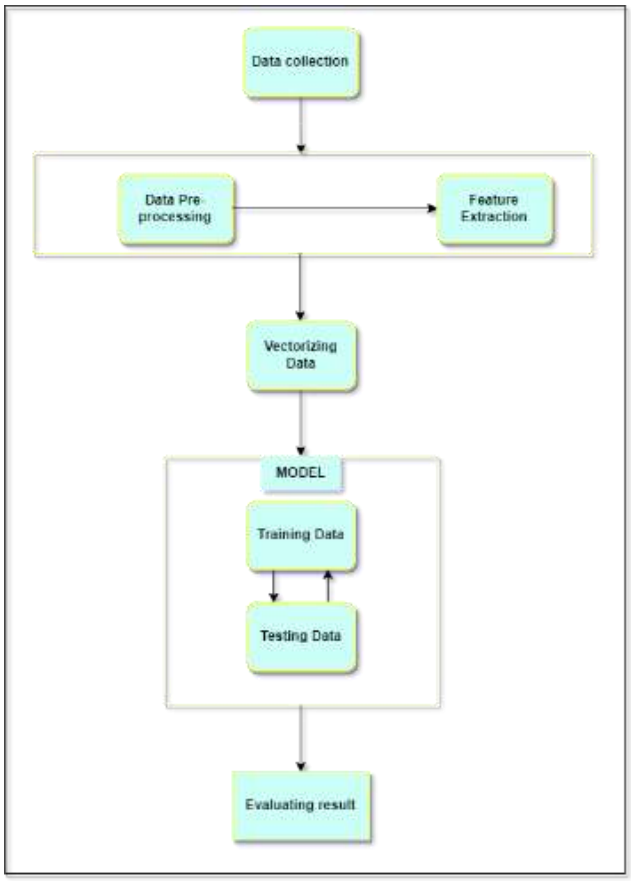

Fig. 1 Methodology Flowchart

Figures 2 to 4 show most frequent words and most frequent positive/negative words, which were obtained after the removal of unnecessary characters. The dataset contains 70000 positive tweets and 100000 negative tweets (Fig. 5).

Fig. 2 Most Frequent words

Fig. 3 Most Frequent Positive words

Fig. 4 Most Frequent negative words

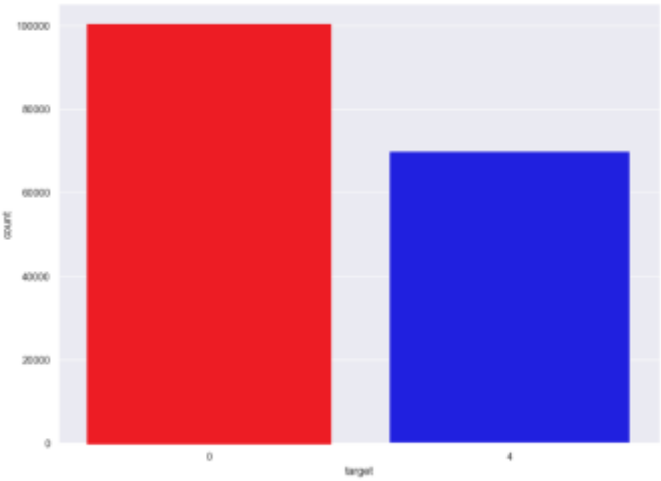

Fig. 5 Count of tweets by label (Red bar negative and Blue bar positive).

B. *NLP Pre-processing*

i. *Bag-of-Words (BOW)*

The BOW model [14] was applied to the dataset X, to convert the text data to numeric values, as preprocessing for DT and SVM. We first selected the top 1000 words (features; $D = 1000$) that mostly appeared in the Twitter dataset X.

https://doi.org/10.31436/ijpcc.v9i2.396

By denoting these D words by $w_1, w_2, \dots, w_D$, the i-th BOW b_i (the BOW of i-th tweet x_i) can be described as

$$b_i = \left(b_{i,1}, b_{i,2}, \dots, b_{i,D}\right)^{\mathrm{T}}, \quad (1)$$

where

$$b_{i,j} = \sum_{k=1}^{m} g(x_{i,k}, w_j) \quad (2)$$

and

$$g\left(x_{i,k}, w_j\right) = \begin{cases} 1 & \text{if } x_{i,k} = w_j \\ 0 & \text{if } x_{i,k} \neq w_j. \end{cases} \quad (3)$$

Then a matrix $B = (b_1, b_2, \dots, b_n)$ was created

ii. Index-Based Encoding

The Twitter dataset X was converted to integer sequences, as preprocessing for the RNN. Each i-th twitter x_i was converted to an integer sequence $y_i = \left(y_{i,1}, y_{i,2}, \dots, y_{i,m}\right)^{\mathrm{T}}$ as follows:

$$y_{i,j} = \sum_{k=1}^{D} kg(x_{i,j}, w_k), \quad (4)$$

for $j = 1, 2, \dots, m$. Then, each sequence y_i was padded as $y_i^* = \left(y_{i,1}^*, y_{i,2}^*, \dots, y_{i,M}^*\right)^{\mathrm{T}}$, such that

$$y_{i,j}^* = \begin{cases} 0 & \text{for } j = 1, 2, \dots M - m \\ y_{i,m-(M-j)} & \text{otherwise,} \end{cases} \quad (5)$$

where M was set to 200, so that every sequence has the same length. Then a matrix $Y = (y_1^*, y_2^*, \dots, y_n^*)$ was created. We used keras tokeniser and pad sequences for this process.

C. Data Collection

The three models: DT, SVM, and RNN with LSTM were used for the classification, and were compared based on the classification performance of each model.

i. Decision Tree

The Shannon entropy and Gini index were used and compared. The entropy at the i-th node is represented by

$$H(i) = -\sum_{y \in \{\mathrm{p,n}\}} p(y|i) \log p(y|i), \quad (6)$$

where $p(y|i)$ is the probability of y at node i. On the other hand, the Gini index is defind as:

$$I(i) = 1 - \sum_{y \in \{\mathrm{p,n}\}} p(y|i)^2. \quad (7)$$

For both criteria, higher value indicates higher impurity. Therefore, the algorithm tried to find branches such that $H(i) - H(i+1)$ or $I(i) - I(i+1)$ is maximized.

ii. Support Vector Machine

The SVMs with the linear, polynomial, and Gaussian kernels were used, and compared. This decision boundary, characterized by $w = (w_1, w_2, \dots, w_D)^{\mathrm{T}}$, was found by minimizing:

$$\frac{1}{2} w^{\mathrm{T}} w + \lambda \sum_{i=1}^{n} \xi_i. \quad (8)$$

Then, each SVM found a hyperplane that classifies positive and negative tweets better.

iii. RNN with LSTM

The M-dimensional dataset Y was fed into the embedding layer, LSTM layer, and three fully-connected layers with recurrent connections (Fig. 6). Through the embedding layer, y_i^* was transformed into a 128-dimensional real-valued vector. These vectors were input to the LSTM layer, and the output was fed into the three layers.

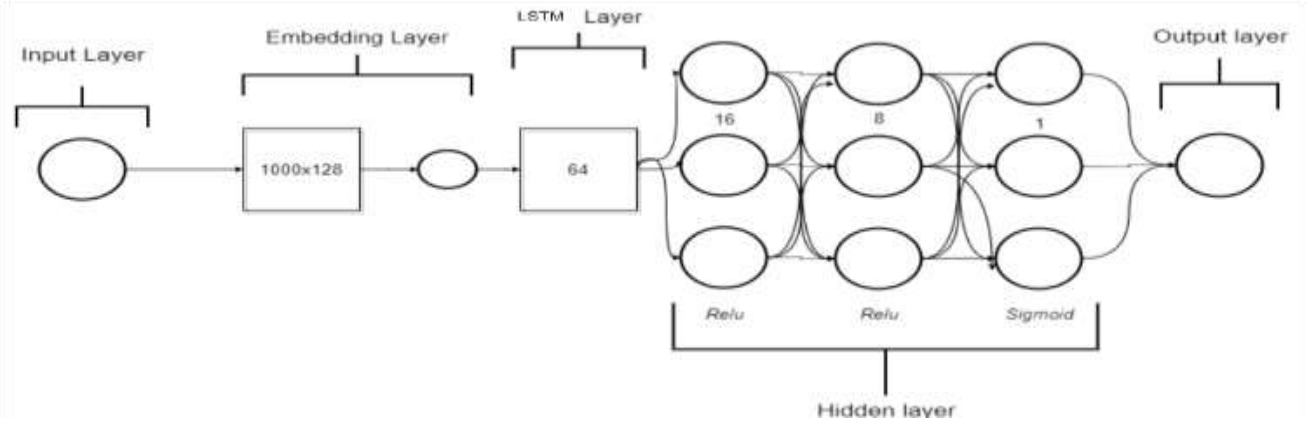

Fig. 6 Layers of RNN

https://doi.org/10.31436/ijpcc.v9i2.396

Three activation functions were used in hidden layers. Two of them were Relu and the other was the sigmoid activation function.

D. Model Evaluation

In this study, 70% of the dataset was utilised for training, and the rest was for testing. Then, the confusion matrix was used to evaluate the above models. To visualise and observe the high-dimensional feature space of our dataset, the truncated singular value decomposition (SVD) was used.

The confusion matrix has four entities indicating the number of true positives (N_{TP}), true negatives (N_{TN}), false positives (N_{FP}), and false negatives (N_{FN}). From these quantities, we calculated the accuracy, precision, recall, and the F_1 score as:

$$\text{accuracy} = \frac{N_{\mathrm{TP}} + N_{\mathrm{TN}}}{N_{\mathrm{TP}} + N_{\mathrm{TN}} + N_{\mathrm{FP}} + N_{\mathrm{FN}}} \quad (9)$$

$$\text{precision} = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}} + N_{\mathrm{FP}}} \quad (10)$$

$$\text{recall} = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}} + N_{\mathrm{FN}}} \quad (11)$$

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \quad (12)$$

III. PRESENTATION OF THE RESULTS

First, the top 1000 words that mostly appeared in the Twitter dataset X were selected. Table 1 presents the top 10 words by frequency.

Table 1. Top 10 Features (words) by frequency

Words	Counts
Just	9716
Thi	7136
Work	6783
Good	6521
Like	6345
Love	5454
Today	5246
Quot	5161
Miss	5119
http	5004

From these words and the Twitter dataset, the 170000-by-1000 BOW matrix B was created through Eqs. (1) to (3). Figure 7 shows the raster plot of the training part of this matrix. The BOW matrix was mostly filled with zero, i.e., sparse matrix, and therefore it was compressed in the CSR (compressed sparse row) format for the sake of computational efficiency. This matrix was used as input to the DT and SVM models.

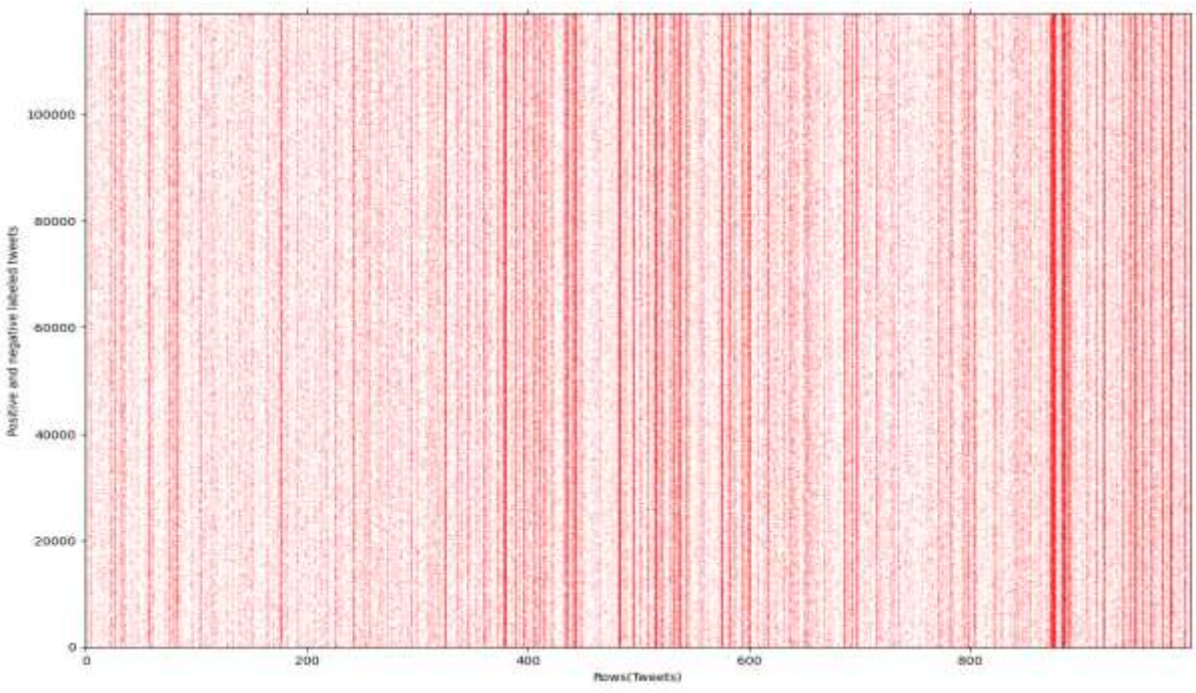

Fig. 7 Training dataset visualization for DT and SVM

https://doi.org/10.31436/ijpcc.v9i2.396

Furthermore, the matrix Y was created through Eqs. (4) and (5) via index-based encoding, and it was fed into the RNN. This matrix had the positive integets between 1 and 1000. Each feature vector was a sequence of these positive integers and zero padded.

Figure 8 presents the comparison of classification accuracy of each model. The accuracy of DT with entroy and Gini index, SVM with linear, polynomial, and Gaussian kernels, and RNN with LSTM were 0.69, 0.68, 0.74, 0.71, 0.75, and 0.76, respectively. Therefore, among the models we used, the RNN had the highest accuracy 0.76 in the Twitter dataset with 170000 samples. For the three kernels of SVM, the Gaussian kernel had slightly higher accuracy than other two.

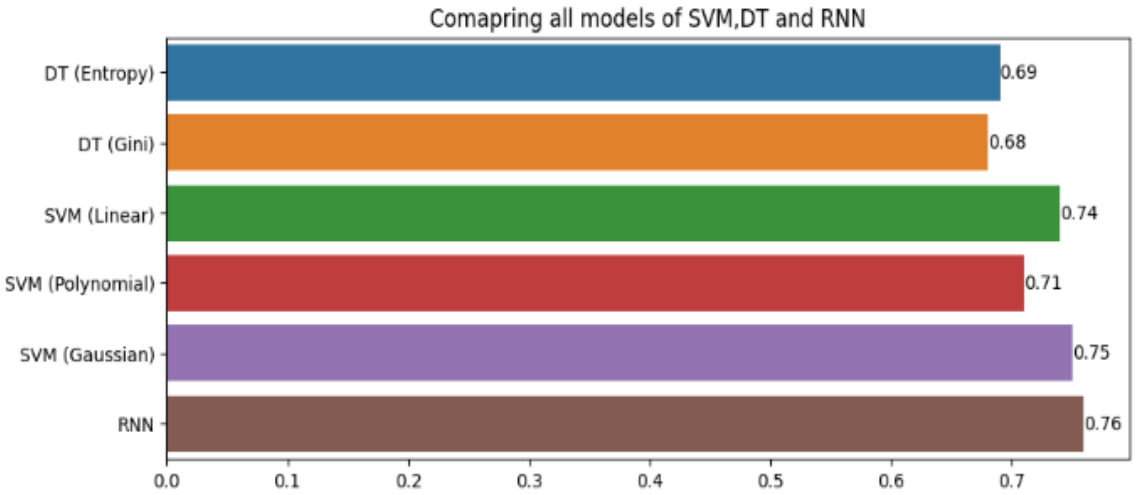

Fig. 8 Accuracy of Models

For the Gaussian kernel in SVM, we analysed the effect of the gamma value on the Gaussian kernel. A low gamma value indicates a distant influence; i.e., a high value indicates a nearby influence. The decision boundary is determined by using distant points when the gamma is small. When gamma is high, it indicates that data points are being considered that are on the borderline of being probable. When the gamma value was 0.7, the accuracy was 0.71. However, after decreasing the gamma value to 0.1, the accuracy jumped to 0.75.

Figure 9 shows the recall, precision, and F_1 score of each model. The recall is also the highest for RNN. For SVM, the value is the highest for the Gaussian kernel. From our results, one can see that the accuracy and precision both are very close except for the SVM polynomial kernel. The accuracy of this model was 0.71, but the precision was 0.68. So, it can be seen that this model did not performed well compared to others. The highest precision was 0.75 from RNN.

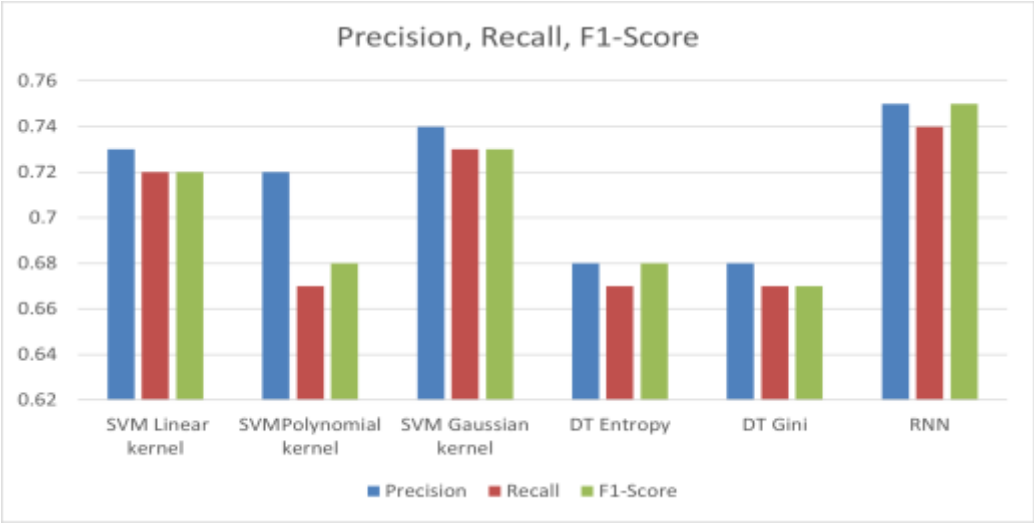

Fig. 9 Recall, Precision and F1-Score of Models

https://doi.org/10.31436/ijpcc.v9i2.396

By utilising the trained SVM model with linear kernel, we investigated as to what words can contribute to positive-negative classification. Figure 10 displays the top 20 positive/negative words, on the basis of the absolte value of SVM coefficient of each word. It can be seen that the words "smile," "proud," "shine," and "glad" were used to categorise positive words. On the other hand, the words "sadly," "bummer," "upset," "poor," and "disappoint" helped the model categorise negative words from positive.

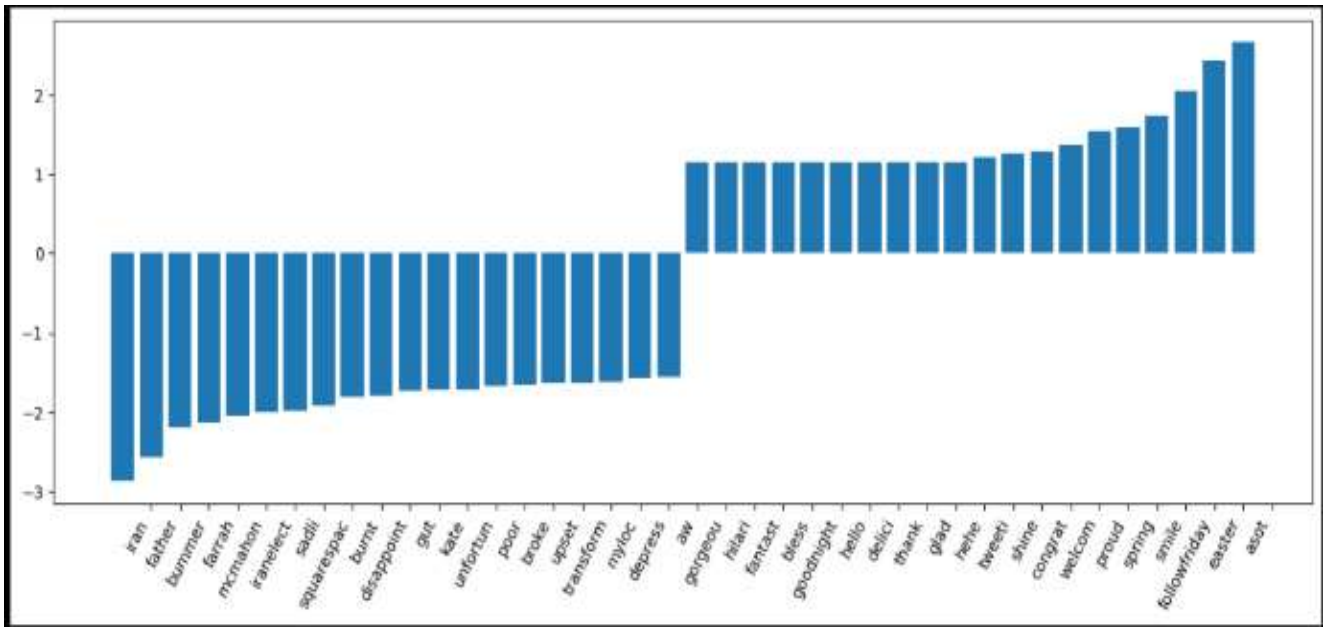

Fig. 10 Top 20 words by Contribution to the classification

In the end, we visualised the 1000-dimensional feature space of the BOW matrix B, onto a plane, to make our results more convincing. B was a sparse matrix, so we applied the truncated SVD to it for this dimensionality reduction. Figure 11 shows the first vs. second components, where the labeled 170000 samples are distributed (blue: positive, red: negative). Decision boundary has also been displayed together, that were obtained from SVM with linear kernel applied to this space. It is clear that there are the greater number of blue dots in the blue area compared to the red area, and vice versa.

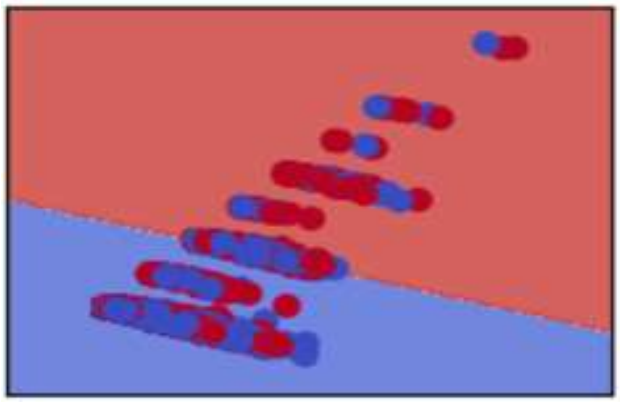

Fig. 11 SVM (Linear) Decision Surface

IV. Conclusions

With this study, we analysed a massive Twitter dataset consisting of over 170000 tweets [11]; each tweet is labeled as positive or negative sentiment. We applied the two kinds of NLP techniques as preprocessing, i.e., BOW and index-based encoding. The BOW was used to convert tweets to integer-valued vectors in a 1000-dimensional feature space, which were fed into DT and SVM. On the other hand, the index-based encoding was employed to convert tweets to integer sequences, being fed into the RNN with LSTM. As a whole, the RNN showed the highest accuracy 0.76 among the models. Furthermore, the trained SVM model (linear kernel) was examined to clarify what kinds of words can contribute to positive-negative classification, motivated by the fact that the SVM with linear kernel still yielded a good classification accuracy (0.74). The truncated SVD was also used to visualise the 1000-dimensional feature space of the BOW matrix (B), to observe how tweets were distributed on the high-dimensional feature space.

The trained SVM model with the linear kernel identified the top 20 positive/negative words that largely contributed to the positive-negative classification (Fig. 10). The words like "smile", "proud", "shine", and "glad" helped the model learn what the positive sentiment is. In contrast, the words

such as "sadly", "bummer", "upset", "poor", and "disappoint" played a vital role in detecting negative sentiment in tweets. Importantly, these positive and negative words were detected in a data-driven manner through training. Based on these words, we may be able to conduct early detection of mental illness; i.e., detecting a state before falling into a severe state.

The methods, in NLP preprocessing and in the classification of positive-negative sentiment, should be examined more rigorously. Our results suggest that the combination of index-based encoding and the RNN with LSTM (i.e. context-dependent feature extraction) is a better option than the BOW with feed-forward models. Visualization of the feature space in each step (input layer, NLP layer, and each layer until the output) would be helpful to understand how the original features are propagated and transformed into the output.

ACKNOWLEDGEMENT

The authors hereby acknowledge the review support offered by the IJPCC reviewers who took their time to study the manuscript and find it acceptable for publishing.

CONFLICT OF INTEREST

The authors declare that there is no conflict of Interest.

REFERENCES

[1] National Institute of Mental Health - Depression. U.S. Department of Health and Human Services. Available at: https://www.nimh.nih.gov/health/topics/depression (Accessed: May 1, 2022).

[2] World Health Organization (2021) Depression, World Health Organization. World Health Organization. Available at: https://www.who.int/news-room/fact-sheets/detail/depression (Accessed: June 30, 2022).

[3] M.R. Islam, M.A. Kabir, A. Ahmed, A.R. Kamal, H. Wang, H., Ulhaq, A. Depression detection from social network data using Machine Learning Techniques. Health Information Science and Systems, 6(1). 2018. https://doi.org/10.1007/s13755-018-0046-0

[4] X. Tao, Dharmalingam, R., Zhang, J., Zhou, X., Li, L., Gururajan, R. Twitter analysis for depression on social networks based on sentiment and stress. 2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC). https://doi.org/10.1109/besc48373.2019.8963550

[5] P. V. Rajaraman, Asim Nath, Akshaya. P. R, Chatur Bhuja. G. (2020). Depression detection of tweets and a comparative test. International Journal of Engineering Research and, V9(03). https://doi.org/10.17577/ijertv9is030270

[6] K. Kumar, Piyush & Samanta, Poulomi & Dutta, Suchandra & Chatterjee, Moumita & Sarkar, Dhrubasish. (2022). Feature Based Depression Detection from Twitter Data Using Machine Learning Techniques. Journal of Scientific Research. 66. 220-228. 10.37398/JSR.2022.660229.

[7] T. Sravanthi et al 2020 IOP Conf. Ser.: Mater. Sci. Eng. 981 022056

[8] A. Amanat, M. Rizwan, A.R. Javed, Abdelhaq, M.; Alsaqour, R.; Pandya, S.; Uddin, M. Deep Learning for Depression Detection from Textual Data. Electronics 2022, 11, 676. https://doi.org/10.3390/electronics11050676

[9] H. Ahmed, Prasadith Buddhitha, Mahmoud Husseini Orabi, and Diana Inkpen. 2018. Deep Learning for Depression Detection of Twitter Users. In Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology, pages 88–97, New Orleans, LA. Association for Computational Linguistics.

[10] A. Zunic P. Corcoran, Spasic I, Sentiment Analysis in Health and Well-Being: Systematic Review JMIR Med Inform 2020;8(1): e16023

[11] K. KazAnova Sentiment140 dataset with 1.6 million tweets, Kaggle. Available at: https://www.kaggle.com/datasets/kazanova/sentiment140 (Accessed: March 30, 2022).

[12] C. Bishop (2006) Pattern Recognition and Machine Learning. Springer

[13] A. Sherstinsky. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network, Physica D: Nonlinear Phenomena, 404, 2020, 132306,

[14] Y. Zhang, Y., Jin, R. & Zhou, ZH. Understanding bag-of-words model: a statistical framework. Int. J. Mach. Learn. & Cyber. 1, 43–52 (2010). https://doi.org/10.1007/s13042-010-0001-0.