Classifying Disaster Tweets with Natural Language Processing

Setareh Sheikholeslamzadeh
Department of Electrical and Computer Engineering
University of Western Ontario
London, Ontario, Canada
ssheikho@uwo.ca

Mehrzad Khodashenas
Department of Electrical and Computer Engineering
University of Western Ontario
London, Ontario, Canada
mkhodash@uwo.ca

***Abstract*—Today, social networks play a huge role in the lives of most people in the world. An example of such a network could be Twitter. Users are tweeting on a wide range of topics every day, and politics, entertainment, sports, and technical issues account for most of these tweets. At times, this data can be related to human-made or natural disasters, and as such, this information could help the government and public understand the situation and make decisions. However, it is not always straightforward to distinguish between a genuine and a fake disaster tweet. Thus, in this work, three classification models are proposed which can identify and distinguish the tweets attributed to disasters, including the LSTM model, SVC and Random Forest. The evaluation results show that the SVC model has the major improvement in classification performance, which achieves the accuracy of 80.69.**

***Index Terms*—Tweet, Twitter Data Classification, LSTM, SVC, Random Forest, Disaster Analysis**

I. Introduction

In our modern era with the advancement of current information technology, social media has become increasingly essential for most people since it provides a window into people's actions and information sources. Social networks create a significant amount of digital data, where people share their thoughts and experiences on a variety of issues. The quick distribution of information through social links is one of the distinguishing aspects of online social networks [1].

One of the good instances of social media networks is Twitter, which allows users to share free short messages called tweets. In most circumstances, a Tweet's text content is limited to 280 characters. It is estimated that Twitter has over 397 million users [2]. Twitter has become a powerful platform to share information around the world. Furthermore, the information gathered through social media can be processed to derive important information, such as detecting disasters and notifying officials [3]. As a result, an increasing number of enterprises are tracking tweets programmatically. However, computers have a hard time distinguishing between the words themselves and their metaphorical meaning, so it's not always evident if a tweet is about disaster or not. Therefore, as a result, an increasing number of enterprises are tracking tweets programmatically. However, computers have a hard time distinguishing between the words themselves and their metaphorical meaning, so it's not always evident if a tweet is about disaster or not. Some words are clear to a human right away, especially with the visual aid. But it's less clear to a machine [4].

A Natural Language Processing (NLP) based strategy to process such information from tweets is being developed to improve disaster recognition, evaluation, and management. This project will distinguish between which tweets are about disastrous events and which not, using the LSTM, SVC and Random Forest models.

The paper is organized as follows: Section II describes related works. In Section III, the dataset is presented. Investigating the dataset is presented in Section IV as exploratory data analysis. Then, preprocessing of the dataset is discussed in Section V. Section VII the methodology is proposed. Section VIII the results and analysis are discussed and finally, in Section IX conclusion of the report is presented.

II. Related Work

Millions of pieces of information are published on social networks every day and Twitter is one of the most famous social network platforms, but not all the information generated by tweets can be used for disaster analysis due to the huge amount of unstructured data. so, the first step would be to preprocess the unprocessed content to analyze the data and produce the results in an understandable way.

A. Sarker, M. R. Islam and A. Y. Srizon proposed a few steps for the preprocessing phase. The first step is tokenization. While tokenization, punctuation, stop words, numbers, html tags and hashtags were removed. The second step would be the stemming. Stemming process is the reducing inflection in words to their root forms [5].

There is a lot of research in the field of data classification and Natural Language Processing. R. Monika, S. Deivalakshmi and B. Janet proposed a Recurrent Neural Network (RNN) model along with Long-Short Term Memory networks (LSTMs) for classifying US airline tweets [6].

Also, S. P. Algur and V. S have performed classification of disasters on specific tweets applying Random Forest, and SVM classifiers [7].

III. Data

The dataset used in this project is a set of tweets, which some are describing actual disasters, and some are not. The source of this data is a challenge from Kaggle, created by the company Figure-Eight [4]. This dataset includes 10000 tweets divided into two datasets. There are 7613 records in the training set and 2387 records in the test set. The variable to be predicted is “target” which contains a binary value to show whether the tweets are real or fake. The other variables are described in Table 1.

Columns	Description
Id	A unique identifier for each tweet
Text	The text of the tweet
Location	The location the tweet was sent from (may be blank)
Keywords	A particular keyword from the tweet (may be blank)
Target	In train.csv only, this denotes whether a tweet is about a real disaster (1) or not (0)

Table 1. Data Description

IV. EXPLORATORY DATA ANALYSIS

Exploratory data analysis is a way of evaluating data sets to summarize their essential characteristics, which frequently involves the use of statistical graphics and other data visualization techniques. This approach helps to have a better understanding about the dataset and be prepared for the preprocessing phase.

A. *Target Counts*

In this dataset, there are 7613 samples with the number of fake targets being 4342 and real targets being 3271, therefore, this dataset can be called an unbalanced dataset. In Figure 1, the number of real data which can be labeled as real disaster visualized in red and fake targets labeled as not disaster has been visualized in blue.

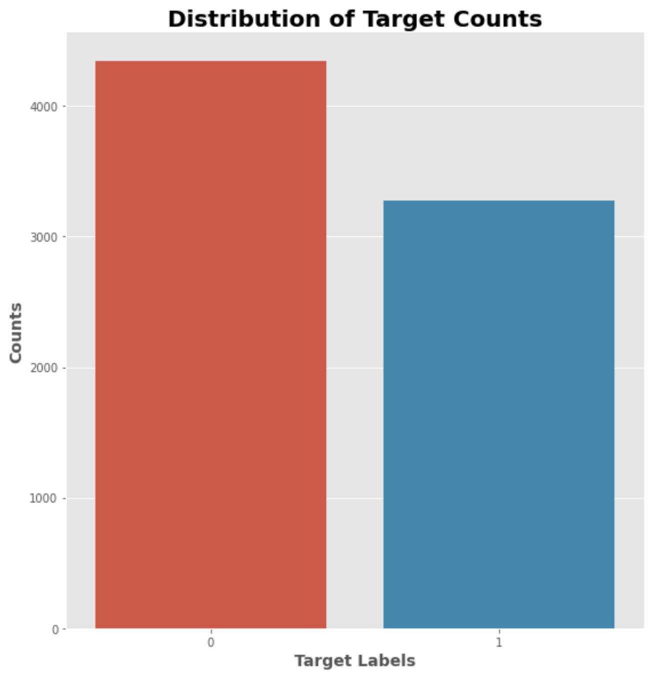


Figure 1. Distribution of Target Counts

B. *Keywords*

In this part, a cloud of words was generated getting all the words from the keyword column. It can be shown in Figure 2 that the more words are repeated in column keywords, the larger the word is displayed.
Figure 3 shows the count of top 15 keywords and top keywords for disaster and non-disaster tweets are shown in Figure 4 and 5.

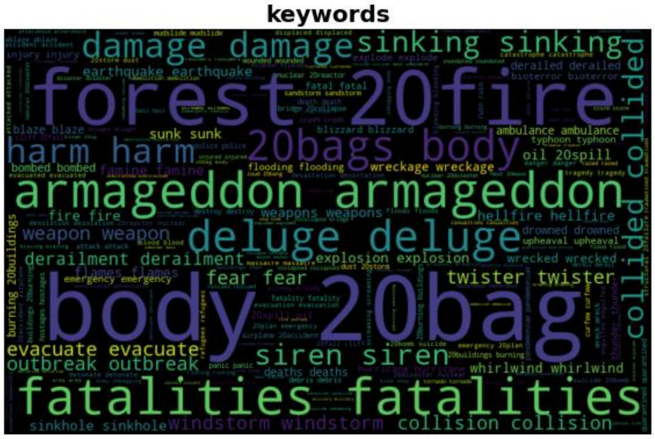


Figure 2. Keywords Word Cloud

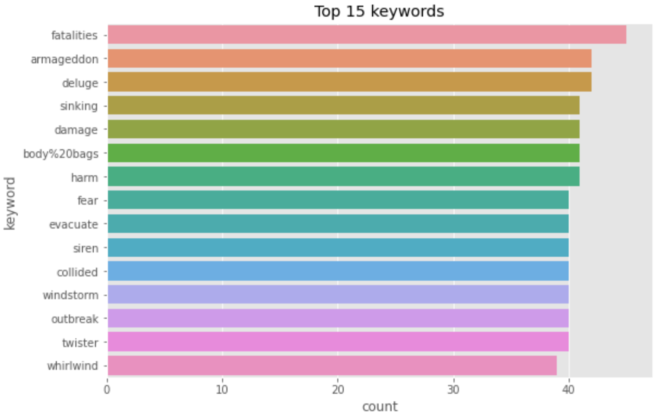


Figure 3. Top 15 Keywords

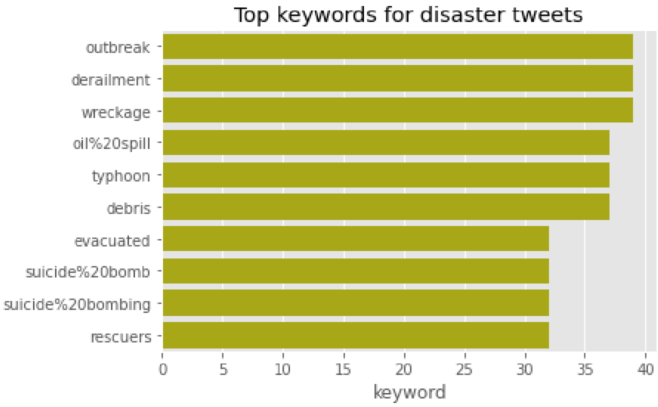


Figure 4. Top Keywords for Disaster Tweets

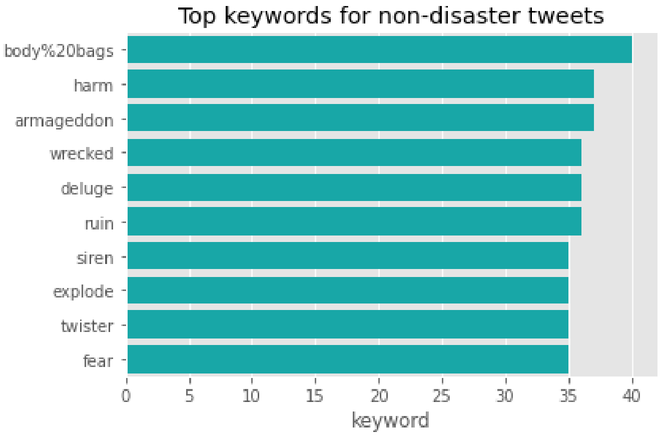


Figure 5. Top Keywords for non-disaster Tweets

V. PREPROCESSING MODEL

Text data must be preprocessed and translated into a uniform and digestible format before the NLP model can be built. The proposed model has three steps. The first step is to convert all text to lowercase, remove hyperlinks, hashtags, HTML tags, non-alphabetic characters, punctuation and stop words. The second step is to remove all emojis that might be in the text. Next, a function for converting the words of the tweets to their base form is written. Using the Wordnet Lemmatizer, the words will go to their base form. And finally, vectorizing. In the vectorizing process, to vectorize a corpus with the bag of words approach, the vectors were built based on n-grams (unigram and bigram) and the train vector has built. Figure 6 illustrates the proposed processing model.

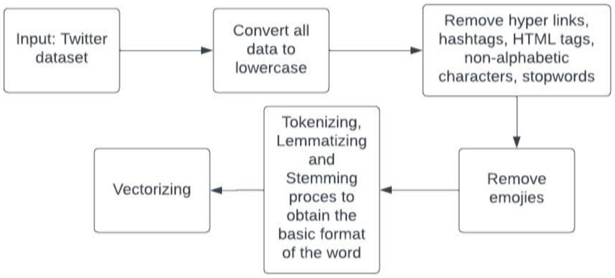

Input: Twitter dataset
Convert all data to lowercase
Remove hyper links, hashtags, HTML tags, non-alphabetic characters, stopwords
Remove emojies
Tokenizing, Lemmatizing and Stemming proces to obtain the basic format of the word
Vectorizing

Figure 6. Illustrate the Proposed Processing Model

VI. N-GRAM ANALYSIS

N-grams are a continuous sequence of n items or words or symbols in a document. When dealing with text data in NLP (Natural Language Processing), they come into play. Figures 7 and 8 show the distribution unigram and bigram of tweet text for real and fake data after preprocessing steps. It can be observed that real and fake disaster tweets use very distinct unigram and bigram sequences, with real disaster tweets more likely to include words such as “fire”, “disaster”, “suicide” and “police”. This is particularly useful in the development and training of the model to distinguish them.

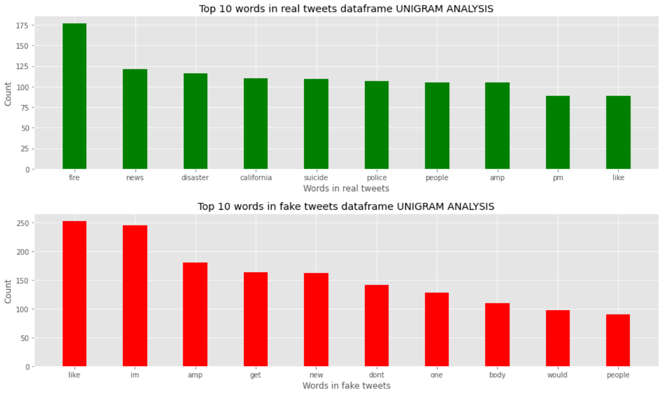


Figure 7. Unigram Analysis on Data

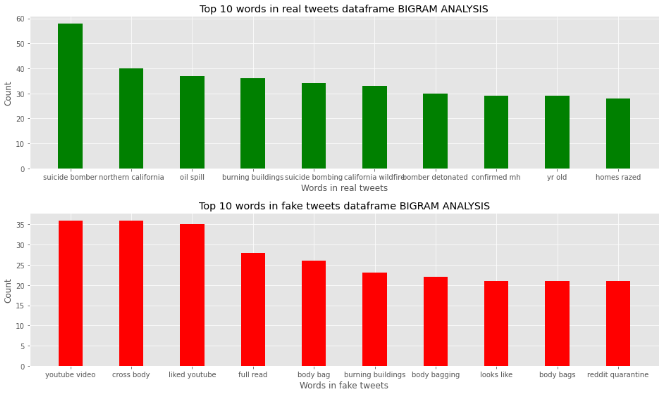


Figure 8. Bigram Analysis on Data

VII. METHODS

In this project, the data consists of text information tweets that are stored in a CSV file extracted from Kaggle dataset [4]. Out of the total dataset, 70% are used for training and 30% are used for testing the classification models. The classification models have been implemented to predict whether each tweet is real or fake based on LSTM, SVC and Random Forest algorithms. The model's performance is measured in terms of the number of instances properly classified based on Precision, Recall and F1-score.

The system model of the methodologies is represented in Figure 9.

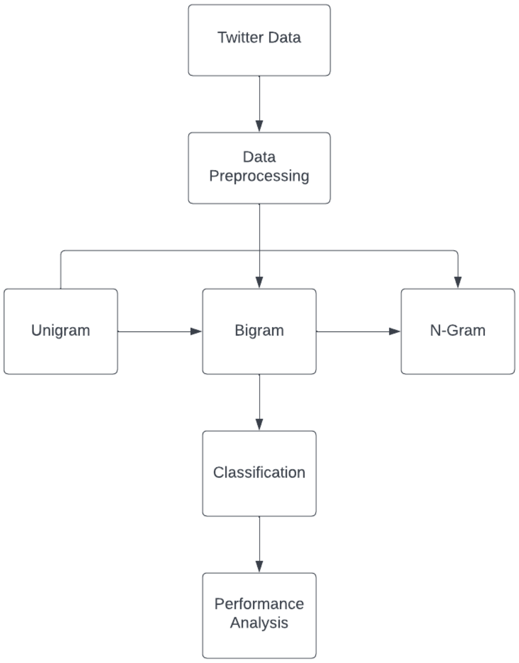


Figure 9. System Model of Methods

A. *LSTM*

LSTM or Long Short-Term memory is a kind of Recurrent Neural Network (RNN) in which the output of each step is an input to the next step. RNN can process sequence data and deal with sequence change better than a normal neural network. For instance, depending on the context, a word may have different meanings, and RNN can solve this problem. This method helps to retain the connection between each step, but, when dealing with long-term sequences, RNN has a long-term dependency problem, where the vanishing gradient problem occurs. In addition, in LSTM the long-term dependencies between each step are reduced by a forget gate, input gate and output gate in the hidden layer.

The memory cell is in charge of keeping track of the dependencies between the input sequence elements. The current and prior inputs are fed to a forget gate, and the forget gates output is fed to the previous cell state. The output of the input gate is then routed back into the prior cell state. The output gate will operate and generate the output as a result of this [8]. The structure of the LSTM network is shown in Figure 10.

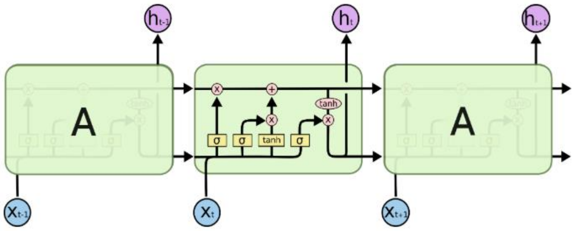
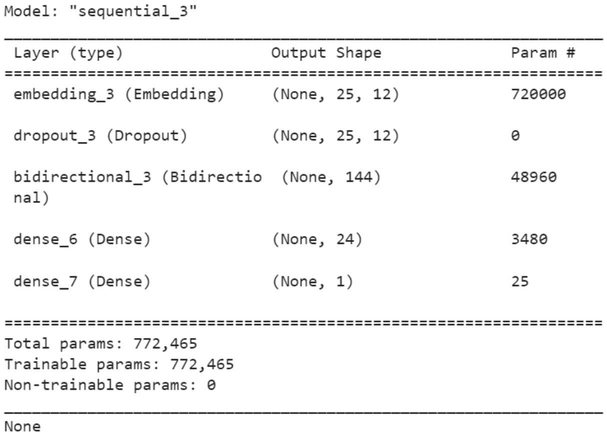


Figure 10. LSTM Structure

In this project, an LSTM model is constructed with the tweet text as the input layer. The number of parameters and the structure of the network model implemented in the project are shown in Figure 11.



```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 25, 12)            720000

 dropout_3 (Dropout)         (None, 25, 12)            0

 bidirectional_3 (Bidirectio  (None, 144)              48960
 nal)

 dense_6 (Dense)             (None, 24)                3480

 dense_7 (Dense)             (None, 1)                 25

=================================================================
Total params: 772,465
Trainable params: 772,465
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 11. LSTM Model Structure with Parameters

In this project, due to the low accuracy of LSTM algorithm on the model which is 51%, hyper-parameter tuning has been considered for the model.

The hyper-parameter used in this model are shown below:

- Embedding_dim is set to 12, therefore elements are going to be embedded in a 12-dimensional space.
- After the embedding layer, a spatial dropout layer with a dropout coefficient of 0.2 is applied. Instead of individual elements, it removes the complete 1-dimension feature maps from the 100 dimensions.
- The output layer's activation function is "sigmoid" due to the binary classification problem.
- Three optimizers have been selected for tuning: RMSprop, SGD and adam.
- Batch size varies from 4, 8 and 16.
- Epochs are from values of 25 and 50.

After hyper-parameter tuning with grid search on the model, the accuracy of 57.18% have been gained.

B. *SVC*

SVC or Support Vector Classifier is a decision-making algorithm that identifies the optimum decision border between vectors belonging to a specified group and vectors that do not. SVC models can categorize new text after being given sets of labeled training data for each category.

A reason to choose SVC as one of the methods applied on this dataset is that this model is a quick and dependable classification technique that works well with a limited amount of data. Higher speed and better performance are the two of the main advantages of SVC.

As said, SVC puts the best decision boundary between two vectors, so, texts should transfer into vectors of numbers. Texts are like a bag of words, and we have a feature for every word that appears in that bag and the value of the feature is the frequency each word is repeated in the text. This method counts how many times a word is repeated in a text and divides it by the total number of words. For instance, in the “Just got sent this photo from Ruby Alaska as smoke from wildfires pours into a school“ sentence from the text column, the frequency of smoke is 1/16.

Using SVC for the project, texts are represented as a vector with thousands of dimensions, each representing the frequency of one of the words of the text. Then, after preprocessing steps, they are given to the model to be trained.

C. *Random Forest*

Random Forest or Random Decision Forest is a method that applies averaging to increase predicted accuracy and control over-fitting by fitting a number of Decision Tree Classifiers on different samples of the dataset. When creating each individual tree, it applies bagging and feature randomization in order to generate an uncorrelated forest of trees whose committee prediction is more accurate than that of any specific tree. It is one of the famous methods for Machine Learning problems.

In order for the Random Forest algorithm to make accurate class predictions, the correlation between data features should be low, that in this dataset the features that are considered as the trees of the forest, are low, so, while the algorithm tries to create these low correlations using feature randomness, the features and hyper-parameters choose for this dataset, will also have an impact on the final correlations [9].

VIII. RESULTS AND ANALYSIS

To evaluate the accuracy of the model discussed, there are four accuracy measures evaluated. Precision, Recall, and F1-Score.

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$\text{Precision} = \frac{\mathit{True\ Positive}}{\mathit{True\ Positive} + \mathit{False\ Positive}}$$

This metric answers that of all tweets labeled as real disaster, how many actually were real? High precision relates to the low false positive rate.

Recall is the ratio of correctly predicted positive observations to all observations in actual class. Recall answers to that of all the tweets that were truly about disaster, how many were labeled.

$$\text{Recall} = \frac{\mathit{True\ Positive}}{\mathit{True\ Positive} + \mathit{False\ Negative}}$$

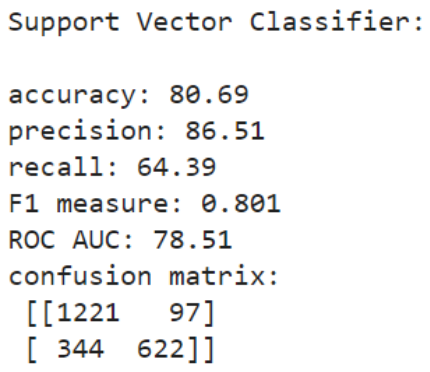
$$= \frac{\mathit{True\ Positive}}{\mathit{Total\ Actual\ Positive}}$$

And F1-Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. When the class distribution of the dataset is uneven, it is good to use F1-Score [10].

$$\text{F1} = 2 \text{ x } \frac{\mathit{Precision} * \mathit{Recall}}{\mathit{Precision} + \mathit{Recall}}$$

Finally, a confusion matrix is shown as a final result.

To evaluate the ability of SVC model, Figure 12 shows the value of accuracy, precision, recall and F1-score on a test set from this model and Figure 13 shows the corresponding confusion matrix.



```
Support Vector Classifier:

accuracy: 80.69
precision: 86.51
recall: 64.39
F1 measure: 0.801
ROC AUC: 78.51
confusion matrix:
 [[1221   97]
 [ 344  622]]
```

Figure 12. Accuracy, Precision, Recall and F1 measure on SVC

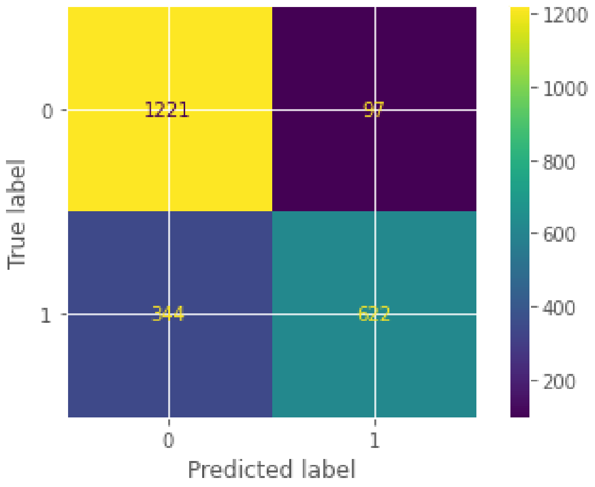
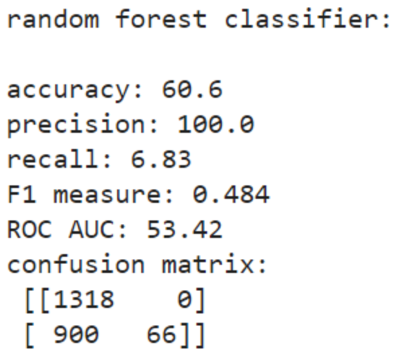


Figure 13. Confusion Matrix of SVC

Also, Figures 14 and 15 illustrate the performance of the Random Forest method.



```
random forest classifier:

accuracy: 60.6
precision: 100.0
recall: 6.83
F1 measure: 0.484
ROC AUC: 53.42
confusion matrix:
 [[1318    0]
 [ 900   66]]
```

Figure 14. Accuracy, Precision, Recall and F1 measure on Random Forest

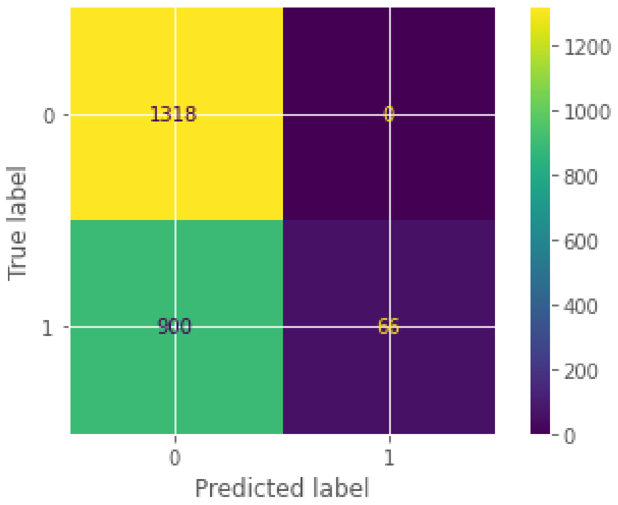


Figure 15. Confusion Matrix on Random Forest

In Table 2, the classification accuracy for the three methods is shown.

Model	Precision	Recall	F1-Score	Accuracy
LSTM	49.71	44.5	0.568	57.18
Random Forest	100	6.83	0.484	60.6
SVC	86.51	64.39	0.801	80.69

Table 2. Classification Accuracy for Different Models

IX. CONCLUSION

This paper compared the LSTM, SVC and Random Forest algorithms performance when predicting the real and fake disaster tweets. SVC outperformed the other two methods in different parameters such as Precision, Recall and F1-score and Accuracy. The LSTM method couldn’t give the best performance among other algorithms because the dataset was not large enough, however, in the future study, by adding more layers and using other deep learning approaches, the model accuracy can be enhanced. One reason behind choosing these three algorithms was that they are the most used methods in literature and papers for text and tweet classification.

A. *Contribution Statement*

In this project all the sections including writing the final report and implementing the code were done equally by the two of us.

REFERENCES

[1] K. Aziz, D. Zaidouni and M. Bellafkih, "Social Network Analytics: Natural Disaster Analysis Through Twitter," 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), 2019, pp. 1-7, doi: 10.1109/ICDS47004.2019.8942337.

[2] How Many People Use Twitter in 2022? [New Twitter Stats]. (2022). Retrieved 5 January 2022, from https://backlinko.com/twitter-users.

[3] S. Z. Razavi and M. Rahbari, "Understanding Reactions to Natural Disasters: a Text Mining Approach to Analyze Social Media Content," 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS), 2020, pp. 1-7, doi: 10.1109/SNAMS52053.2020.9336570.

[4] Natural Language Processing with Disaster Tweets — Kaggle. (2019). Retrieved 20 December 2019, from https://www.kaggle.com/c/nlp-getting-started/overview.

[5] A. Sarker, M. R. Islam and A. Y. Srizon, "A Comprehensive Pre-processing Approach for High-Performance Classification of Twitter Data with several Machine Learning Algorithms," 2020 IEEE Region 10 Symposium (TENSYMP), 2020, pp. 630-633, doi: 10.1109/TENSYMP50017.2020.9230590.

[6] R. Monika, S. Deivalakshmi and B. Janet, "Sentiment Analysis of US Airlines Tweets Using LSTM/RNN," 2019 IEEE 9th International Conference on Advanced Computing (IACC), 2019, pp. 92-95, doi: 10.1109/IACC48062.2019.8971592.

[7] S. P. Algur and V. S, "Classification of Disaster Specific Tweets - A Hybrid Approach," 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), 2021, pp. 774-777, doi: 10.1109/INDIACom51348.2021.00138.

[8] Recurrent Neural Network and Long-Term Dependencies. (2019). Retrieved 14 July 2019, from https://medium.com/tech-break/recurrent-neural-network-and-long-term-dependencies-e21773defd92#:~:text=In%20theory%2C%20RNN's%20are%20absolutely,is%20called%20Vanishing%20gradient%20problem.

[9] Understanding Random Forest. (2019). Retrieved 12 June 2019, from https://towardsdatascience.com/understanding-random-forest-58381e0602d2.

[10] Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures. (2016). Retrieved 9 September 2016, from https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/.