Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Leveraging Hybrid Model for Classification of Disaster-Related Tweets using TF-IDF and GCN

Basudev Nath[1], Deepak Sahoo[1], Sudhansu Shekhar Patra[2]

[1]*Faculty of Engineering and Technology, Sri Sri University, Cuttack, India*
nathbasudev@gmail.com
[2]*School of Computer Applications, KIIT Deemed to be University, Bhubaneswar, India*

During catastrophes, Twitter is essential for sharing information in real time; however, due to the large amount of data, it is necessary to classify tweets as connected to the disaster or not. This research provides a hybrid model to improve the categorization of catastrophe tweets using Graph Convolutional Networks (GCNs) and Term Frequency-Inverse Document Frequency (TF-IDF). The top weighted words are used by TF-IDF to create feature vectors, while GCNs use graphs built using Normalized Point-wise Mutual Information (NPMI) to represent the relationship structure between tweets. Tweets are classified using a fully connected neural network using concatenated features from GCN and TF-IDF. The model also visualizes communication networks, preprocesses and tokenizes tweets, and uses t-SNE and PCA to visualize node embeddings and analyze tweet length distributions. In categorizing tweets connected to disasters, the hybrid model outperforms a GCN-only method (80.99%) with an accuracy of 91.55%. Enhancing classification accuracy while offering insightful information on the dynamics of information sharing and community involvement in times of crisis is one of the main benefits of this technique.

Keywords: Twitter analysis, Graph Convolution Network (GCN), Natural Language Processing, Hybrid TF-IDF and GCN, Twitter categorization.

1. Introduction

Social media platforms like Twitter's exponential growth have a significant impact on how people exchange and consume information. During emergencies, Twitter is a priceless source of up-to-date information that offers immediate perspectives into the situation on the ground. The volume of tweets generated at such events greatly hinders efficient data extraction and classification. Deciding which tweets are relevant to catastrophes and which are not is crucial for response teams, governmental agencies, and humanitarian groups to make decisions in a timely manner (Acikara et al.,2023). The casual and even loud language used on Twitter presents unique challenges for classification, particularly when identifying material about disasters. The majority of representations used in classical text classification approaches are either Term Frequency-Inverse Document Frequency (TF-IDF) or Bag-of-Words (BoW). Support Vector Machines (SVM) and Naive Bayes are two examples of these methods. while these methods have their uses, they often fall short of capturing the intricate relationships and interdependence that occur between words in tweet settings. We provide a hybrid model that combines the advantages of TF-IDF and Graph Convolutional Networks (GCN) in order to overcome these drawbacks. The TF-IDF method efficiently captures the significance of individual words in a tweet by using a statistical metric to assess word importance. However, by visualizing the content as a graph, GCNs are skilled at simulating the structural links between words. A more complete representation of the tweet content is produced by the model's ability to use both statistical and graph-based characteristics thanks to this dual approach. Using the TF-IDF approach, we first create a vector of twenty components for each tweet in our proposed model, which is obtained from the top twenty weighted terms. The co-occurrence associations between words are then captured via a graph that is created using these vectors and Normalized Point-wise Mutual Information (NPMI). Using GCN, the graph-based features are retrieved, and the TF-IDF vectors are concatenated with these features. For categorization into catastrophe and non-disaster categories, the aggregated feature vector is then input into a fully connected neural network. We evaluate the effectiveness of our hybrid strategy using a benchmark Twitter dataset, demonstrating its superior performance over more traditional methods. This work shows that GCN and TF-IDF (Danday and Murthy, 2022) may be used to classify tweets, offering a dependable way to extract relevant data from social media during emergency situations.

A. Objective

The objectives of the paper are as follows:

- To Examine Twitter data to determine whether it is "disaster-related" or "not-disaster-related".
- By creating a hybrid model that combines TF-IDF and Graph Convolutional Networks (GCNs), this paper seeks to improve disaster tweet classification. It does this by increasing accuracy and offering insightful information about community involvement and information dissemination during emergencies.

B. Organization

The format of article is organized in this manner. Section 1 serves as an introduction; Section 2 provides a summary of pertinent research on tweet classification. The approach, involving collecting data, initial processing, TF-IDF vectorization, and model training, is covered in Section 3. Results analysis are shown in Section 4 along with a performance assessment of each model. Section 5 concludes with an overview of the findings and future research.

2. Related Work

Disaster management has paid a lot of attention to the usage of social media data in recent years. In order to categorize and examine tweets about disasters, several research have investigated various approaches and datasets. In order to place our study within the present research environment, this section evaluates the most pertinent literature. There are several uses for NLP, such as sentiment analysis, text messages about disasters or not, and sentiment analysis, where categorizing tweets requires in-depth research. The right approach must be used to represent the tweets in order to classify them successfully. Researchers identify tweets using a representation approach called Bag-of-words (BOW) (Wang et al., 2009), which considers statistical information but ignores contextual information. As natural language processing research advances, pre-trained word embedding approaches like word2vec (Mikolov, 2013) are being used in place of the Bag-Of-Words (Blei et al.,2003) representation, providing an improved representation of words in a dataset. Using machine learning techniques, a number of studies have categorized tweets related to tragedies. During natural disasters, authors (Imran et al., 2016) used ML methods like SVM and Random Forests to classify tweets into groupings like "damage," "needs," and "donations." Their research highlights how effectively SVM handles high-dimensional twitter data, and is similar to our approach of utilizing TF-IDF and SVM to categorize catastrophic tweets. Researchers (Sudha and Dhanalakshmi, 2023) looked explored a number of machine learning methods for the categorization of tweets pertaining to catastrophes. Among these algorithms were K-Nearest Neighbors (KNN), Decision Trees, and Naive Bayes. In line with our use of Naive Bayes for tweet categorization concerning catastrophes, they found that the system performed well since it was simple to use and efficient for processing text data.

In order to identify actionable pleas for assistance during natural catastrophes, the authors (Devaraj et al., 2020) create tweet classifiers. The study indicates that CNN, SVM, and MLP models attain high F1 scores above 0.86 using Hurricane Harvey tweets. Average word embeddings' usefulness for non-neural models is emphasized, with competitive outcomes when compared to conventional features. As far as the authors are aware, there aren't many studies that employ graph convolutional neural networks (GCNs) (Yao et al., 2019) to categorize the disaster-related tweets that we used for our research. In order to categorize the tweets, Liu et al.'s approach (Liu et al.,2016) builds a graph using words and documents and integrates embeddings from bidirectional LSTM (Linmei et al., 2019) and GCN (Yao et al., 2019). In order to use CNNs for tweet categorization tasks, authors (Li et al., 2021) showed how well they work at capturing geographical hierarchies in data. Likewise, trained language models like as BERT have been refined for certain tasks, demonstrating significant gains in natural language comprehension and categorization. The authors (Kipf and Welling, 2016) developed graph convolutional networks (GCNs) for semi-supervised learning by extending convolutional neural networks to graph-structured input. Their method disperses label information using graph convolutions, enabling efficient predictions on both labelled and unlabeled nodes. This work has significantly enhanced the use of neural networks in graph domain applications. Authors (Paul and Balabantaray, 2022) provide VocabGCN-BERT, a hybrid model that combines pre-trained BERT with Graph Convolutional Networks (GCN) built from twitter vocabulary graphs. By combining the local contextual knowledge of BERT with the global structural information of GCN, it improves tweet categorization. The findings demonstrate better performance over the current GCN-based models on seven datasets, with a notable improvement in F1 score and accuracy.

In order to classify catastrophe tweets using Twitter data, the authors (Pratama and Pardede, 2023) suggested a CNN-GRU hybrid model. By merging CNN's ability to handle high-dimensional data with GRU's effectiveness in processing sequential data, the model incorporates FastText embeddings and achieves better performance compared to conventional and standalone deep learning methods.

3. Model Description:

The proposed model consisting of various phases. Fig. 1 gives the flow chart of the proposed model. The different phases are depicted below:

1. Start: The process initiates.

2. Collect Dataset: Gather tweets to create a dataset containing both disaster-related and non-disaster-related tweets.

3. Preprocess Data: Perform data cleaning and preprocessing, including text cleaning, stop word removal, and tokenization.

4. TF-IDF Vectorization: Convert the cleaned text data into numerical vectors using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization.

5. Graph Convolutional Network (GCN): Using GCN for graph creation and embedding.

6. Fully Connected Neural Network: Using a fully connected neural network to combine embedding and classification.

7. Model Evaluation: Evaluate the model's performance such as accuracy, precision, recall, and F1-score to understand their effectiveness.

8. Compare Results: Compare the evaluation metrics of all models to determine the best-performing one.

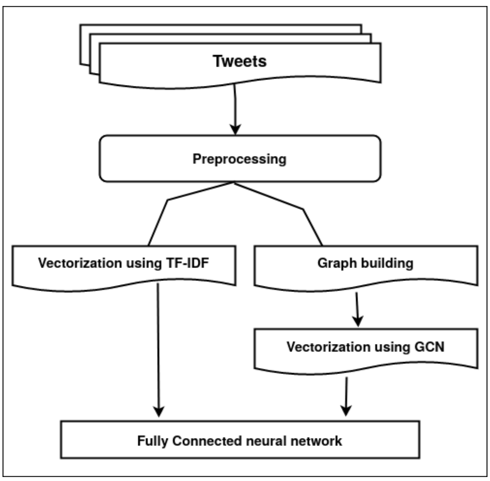

Fig. 1. Flow diagram of proposed classification model

3.1. Data Description

A benchmark dataset from Twitter, comprising a wide variety of tweets classified into disaster and non-disaster categories are utilized for this study. For our study, we've used a tweeter dataset of 33,370 tweets. The dataset consists of a collection of several catastrophic event kinds. The tweets contained 80% training data and 20% data for testing. We were using Python 3.8 and Google Colab as our working environment. The Target Distribution of classes is shown in fig.2. The study of the pie chart shows that there is a minor class imbalance in our dataset, with 56.8% of the tweets connected to disasters and 43.2% not related to them. For this reason, our Graph Convolutional Network (GCN) classifier has an 80.19% accuracy rate. This implies that the imbalance may be effectively handled by our approach, which is able to distinguish between tweets related to disasters and those not. There are three attributes in our dataset: Target, Tweet _text, and Tweet _Id. Fig. 3 shows a portion of the dataset is displayed. Fig. 4 shows Tweet data, including Tweet_Id, Target, Tweet_Text, cleaned_tweet, and length are displayed in the table. The ID, original text, cleaned text, tweet length, and catastrophe relevance (1 for relevant, 0 for irrelevant) are all displayed in each row. A common word frequency distribution, or Zipf's law, is depicted by the curve in fig. 5 and is frequently observed in natural language processing (NLP). It demonstrates that while stop words are rather common, they add very little to the meaning of the words. Words become less common as we go down the curve, but their importance or worth rises. Even though they are rare, words in the tail of the curve usually convey more particular information, making them more useful for tasks like text categorization or sentiment analysis for determining meaning.

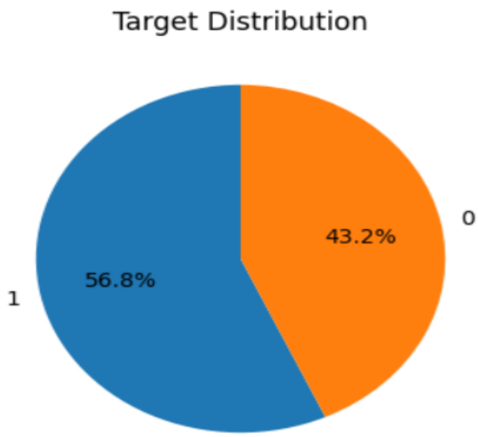

Fig. 2. Target Distribution of classes

	Tweet_id	Target	Tweet_text
0	5.400000e+17	1	fallen tree branches scattered in sorsogon cit...
1	5.080000e+17	1	chiniot flood relief camps mr khademe ala how ...
2	4.770000e+17	0	landslide fletwod mac
3	5.420000e+17	0	my nieces are forcing me to play the frozen ag...
4	4.770000e+17	1	africa algeria mers death reported a yearold m...

Fig. 3. The tweet dataset

	Tweet_id	Target	Tweet_text	cleaned_tweets	length
0	5.400000e+17	1	fallen tree branches scattered in sorsogon cit...	fallen tree branch scattered sorsogon city str...	75
1	5.080000e+17	1	chiniot flood relief camps mr khademe ala how ...	chiniot flood relief camp mr khademe ala long ...	100
2	4.770000e+17	0	landslide fletwod mac	landslide fletwod mac	21
3	5.420000e+17	0	my nieces are forcing me to play the frozen ag...	niece forcing play frozen rubyph clear sky nee...	135
4	4.770000e+17	1	africa algeria mers death reported a yearold m...	africa algeria mers death reported yearold man...	101

Fig. 4. Tweet along with its cleaned tweets and length

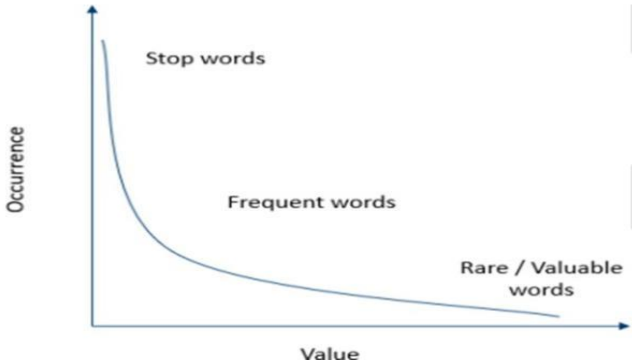

Fig. 5. The occurrences and values of various categories of words

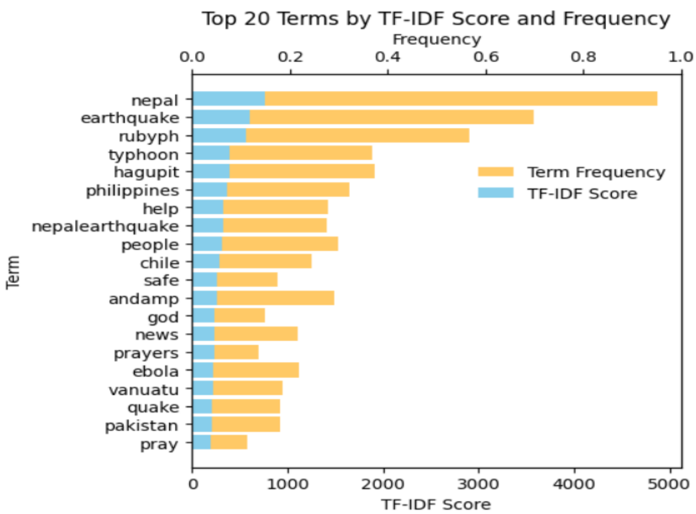

Fig. 6. Top twenty terms related to Disaster: Frequency vs TF-IDF

Fig. 6. displays the top 20 terms in a dataset pertaining to disaster communication, sorted by both term frequencies (orange) and TF-IDF scores (blue). On the y-axis, the terms are listed, and on the x-axis, the TF-IDF score and term frequency. While frequency indicates the term's frequency of occurrence, TF-IDF quantifies a word's value in relation to the text and the corpus. As an illustration of its key position in the dataset, "nepal" has the greatest frequency and TF-IDF score. The most important terms and the frequency with which they occur in the

dataset may be found using this dual visualization. Fig. 7 shows the maximum, minimum and average tweet length. The word clouds for disaster and non-disaster tweets are shown in fig. 8. The tweet length distribution for catastrophe and non-disaster tweets is displayed in the histogram fig. 9. The frequency is plotted on the y-axis, while tweet length is represented on the x-axis. Red bars show tweets about disasters, and blue bars show tweets that are not related to them. The graph facilitates the comparison of tweet lengths in these two groups.

```
                      Tweet_length
Max Length Tweet               157
Min Length Tweet                 3
Average Tweet Length: 77.29668254967186
```

Fig.7. Maximum, Minimum and Average tweet lengths

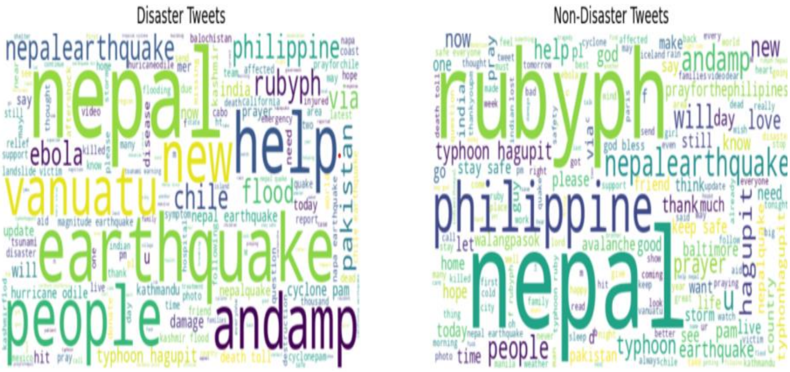

Fig. 8. Word cloud for Disaster and Non-disaster Tweets

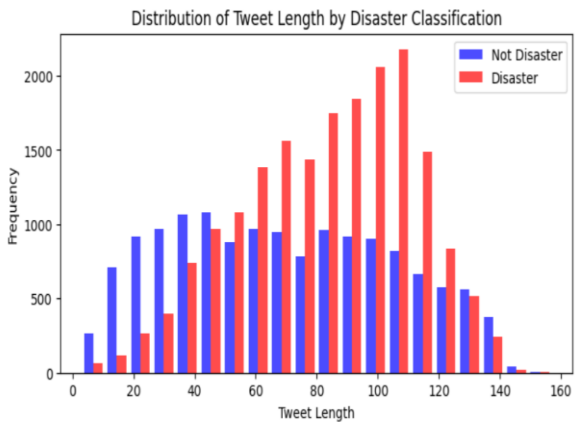

Fig. 9. Distribution of tweet length by disaster classification

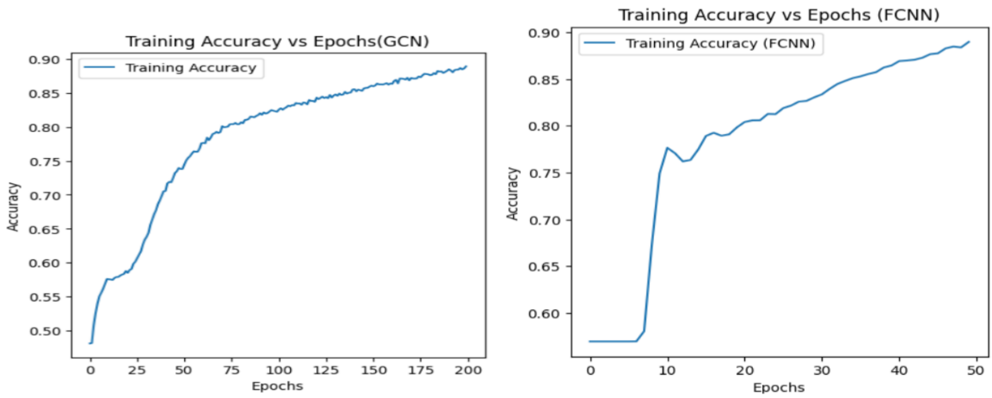

Fig. 10. Training Accuracy vs Epochs for GCN and FCNN

Fig.10 shows how a Graph Convolutional Network (GCN) and a Fully Connected Neural Network (FCNN) compare in terms of training accuracy across several epochs. In the first 10 epochs, the FCNN model (first figure) learns swiftly, attaining accuracy of 0.80 soon and stabilizing at 0.88 by epoch 50. However, the GCN model (second figure) shows a more gradual increase, taking 25 epochs to achieve 0.70 and around the same accuracy as the FCNN (approximately 0.88), but over a longer 200 epochs. In the end, the performance of both models is comparable, but GCN shows continuous, consistent learning whereas FCNN converges more quickly. The graphical structure of tweets pertaining to disasters is visualized in the graphic shown in fig. 11. Edges show linkages based on co-occurrence patterns, and each node represents a tweet. To provide a visual picture of the structure and clustering patterns of the data, nodes are color-coded based on the labels they bear (disaster or non-disaster). Tweets about disasters are represented by red nodes and purple nodes indicate tweets unrelated to disasters.

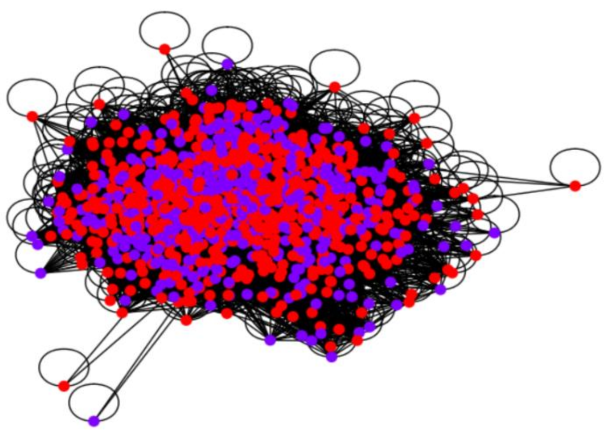

Fig. 11. Visualization of the Disaster Communication Network

The community structure of tweets is shown in fig. 12. The nodes' colors in the above figure correspond to various communities inside the network. According to the Louvain approach, a well-liked technique for community discovery, each individual color represents a different community. The color of a node within the same community indicates that its connections to other nodes are not as dense as those of nodes outside the community. Since the tweets are tightly connected based on their relationships, the color-coding makes it easier to visually discern between the various clusters or groupings of tweets.

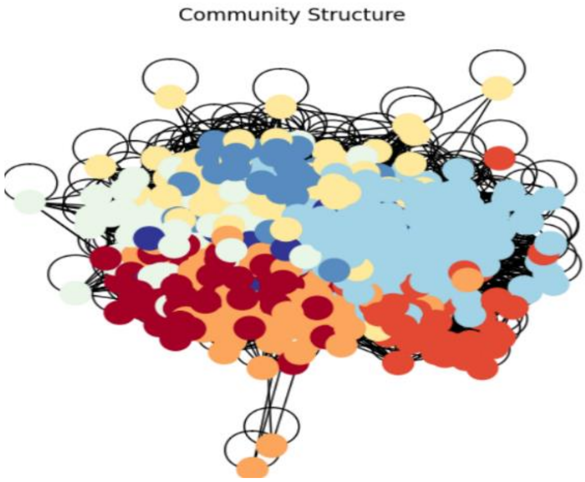

Fig. 12. Community structure of tweets

Fig.13. shows the node embeddings of tweets that have been visualized using Principal Component Analysis (PCA) and t-SNE visualization which are used to represent nodes in a network in a lower-dimensional space. Each dot represents a tweet; tweets that are not disasters (Target 0) are denoted by blue dots, whereas tweets that are disasters (Target 1) are depicted by orange dots. The model's classification performance and the two classes' separability in the embedding space are both made clear by plotting. The nodes form greater complexity and distinct clusters in the t-SNE plot, demonstrating a strong division between the classes. The nodes in the PCA plot are more dispersed, show less clear grouping, and have greater class overlap.

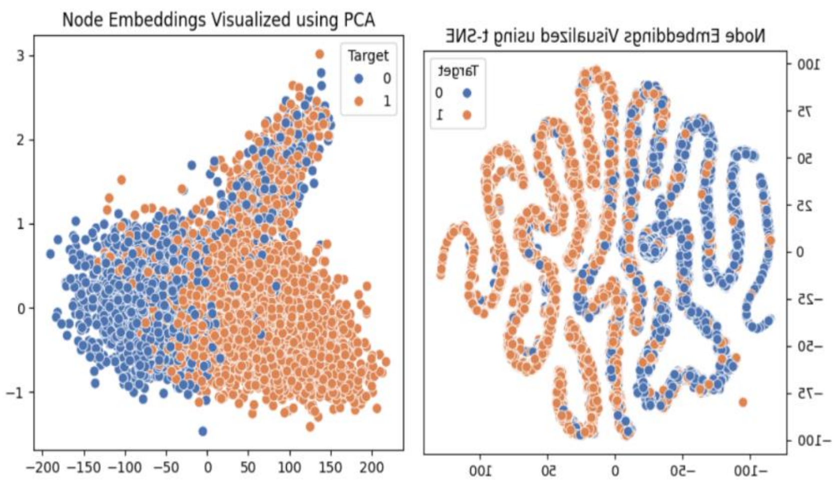

Fig. 13. Node Embeddings visualized using PCA and t-SNE

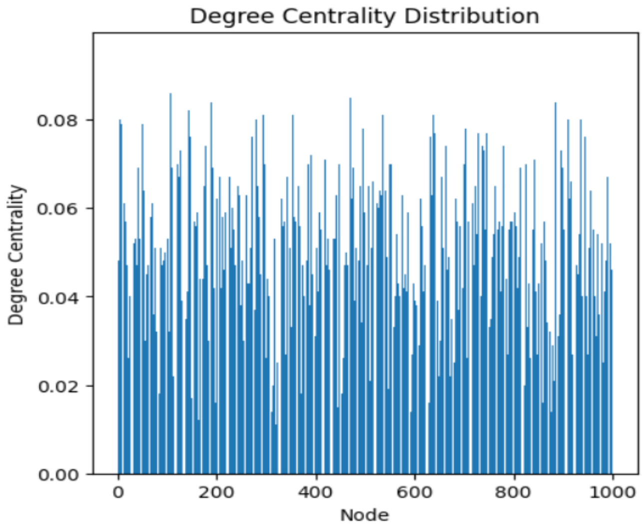

Fig. 14. Degree Centrality Distribution

Fig. 14.shows the Degree Centrality Distribution of nodes in a graph, which is important for classifying tweets with Graph Convolutional Networks (GCNs). Whereas the edges in this case denote interactions like mentions, responses, and retweets, the nodes here stand for tweets. Individual tweets are displayed on the x-axis, which is indexed from 0 to 1000, and their degree centrality values, which range from 0 to 0.08, are displayed on the y-axis. The tweets that have higher degree centrality are highlighted by spikes in the graph, suggesting that these tweets are more important to the network and are essential for disseminating information connected to disasters. During a crisis, these high-centrality tweets are vital for disseminating critical updates since they frequently link to other significant tweets. Sorting these tweets according to their network influence can aid in prioritizing the dissemination of information and emergency relief activities.

3.2. Tweet Preprocessing

Before extracting features from the tweets and training the model, data preparation is essential. The actions consist of:

Text cleaning: To maintain consistency throughout, all text was changed to lowercase. To cut down on noise, punctuation, special characters, and numerals were eliminated. By doing this step, the model is able to prevent disparities that arise from text characteristics that are not significant.

Tokenization: To separate tweets into discrete words, or tokens, we employed word-level tokenization. In order to break down phrases into digestible chunks that the model can learn from, this procedure is essential.

Stop Word Elimination: Term occurrences such as "and," "the," and "is" were eliminated. Despite their frequency, these stop words have little bearing on how disasters are classified, thus eliminating them allows the model to concentrate on terms that are more useful while also simplifying the data.

Stemming: To assist the model, generalize across various variants of the same word, we used Porter stemming to reduce words to their root forms (e.g., "flooding" becomes "flood"). Improving model accuracy requires these preprocessing techniques. The model is trained on consistent, significant data by streamlining the language and eliminating superfluous complexity. For example, stop word elimination and stemming assist in helping the model concentrate on important details, while text cleaning and tokenization guarantee that the data is consistent. All things considered, these actions help to streamline the training process and improve the ability to categorize tweets pertaining to disasters.

3.3. Word Embedding using TF-IDF

We created a vector with twenty components to represent every tweet in the dataset. The data set's top twenty weighted words make up the twenty components. Using the TF-IDF statistical technique, the top twenty weighted terms are filtered. Here are the steps to turn every tweet into a twenty-component vector.

Determine the TF (term frequency):

This calculates how often a word appears in tweet.

TF (t, d) = (term t's number of appearances in tweet d) / (total terms in tweet d). The term frequency (TF) for the word "earthquake" in a tweet containing ten words is 3/10=0.3 if the phrase occurs three times in the tweet.

Inverse Document Frequency (IDF):

This calculates a term's significance over all tweets

IDF(t) = log (Number of tweets) / (Number of tweets containing term t). For instance, if there are 2000 tweets overall and 50 of those tweets contain the phrase "earthquake," then log (2000/50) = log (40) = 1.6020 is the inverse document frequency (IDF) for "earthquake".

Vectorization:

"vectorization" converts text input data into vectors of numbers, which machine learning models can handle. This strategy has been useful in several industries since the development of computers and is now employed in NLP. Vectorization occurs during ML feature extraction. By converting text into numeric vectors, the objective is to extract unique text features for model training. The steps of vectorization are as follows:

Step 1: Locate DFi and TFi

- Remove stop words and tokenize every tweet in the dataset to extract every single word.
- Determine the term frequency (TFi) of every word.
- Determine the document frequency (DFi), which is the number of documents in which a phrase appears.
-

Step 2: Compute the inverse document frequency (IDFi)

IDF_i= $LogN/Df_i$, (1)

where N is the total number of tweets in the dataset. Tfi is a term's frequency.

Step 3: Calculate Term's weight $w_i = Tfi * IDF_i$ (2)

Step 4: Order the keywords by weight and select the top 20.

Step 5: Select the top twenty terms from each tweet and compare them; if a phrase matches, its weight is shown, and otherwise, it is represented by zero.

Step 6: provide a Tweet's fixed-size vector

3.4. Graph Construction and embedding using GCN

Graph-structured data is the intended application domain for graph convolutional networks (GCNs). A GCN is made up of layers that directly execute convolutions on the graph, allowing information to be propagated across the network according to the topology of the graph. Fig.15 gives the graph construction and embedding using GCN. An outline of how GCNs operate is provided as follows.

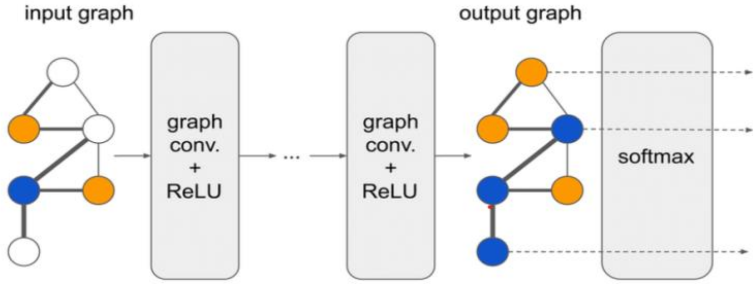

Fig. 15. Graph Construction and embedding using GCN (Liu et al.,2021)

1) Representation of Graphs

For the sake of our twitter categorization assignment, we use a graph G= (V, E) to represent the tweets and their relationships. V stands for the set of tweets, and E stands for the set of edges, or the connections between tweets. The Normalized Pointwise Mutual Information (NPMI) measure is used to quantify the connections.

2) The Normalized Pointwise Mutual Information (NPMI)

The NPMI calculates the correlation between word pairings in tweets. It is employed in the construction of the graph's adjacency matrix A, where the NPMI of each word in two tweets determines the edge's weight. Here are the steps to compute NPMI and create the graph:

3) Determine the Pointwise Mutual Information (PMI)

$PMI (wi,wj)=log(p(wi,wj)/P(wi)\times P(wj))$ (3)

Here P(wi,wj) represents the likelihood that the words wi and wj will appear together in the same context, and P(wi) and P(wj) represent the likelihood that each word will occur separately.

4) Normalize PMI

NPMI (wi,wj)=PMI(wi,wj)/ -log P(wi,wj) (4)

The value of NPMI lies between -1 to 1. NPMI vale zero indicates w1 and w2 are independent, no relationship between w1 and w2. The NPMI values of the first five rows (the first five tweets) and five columns are displayed in the below fig. 16.

```
[[ 0.81385164 -0.78202929 -0.78858031 -0.84969291 -0.82153647]
 [-0.78202929  0.82817795 -0.77449719 -0.83560979 -0.80745335]
 [-0.78858031 -0.77449719  0.80479471  0.05661244 -0.81400437]
 [-0.84969291 -0.83560979  0.05661244  0.74773537 -0.87511698]
 [-0.82153647 -0.80745335 -0.81400437 -0.87511698  0.7987819 ]]
```

Fig. 16. The NPMI values of the first five rows and columns

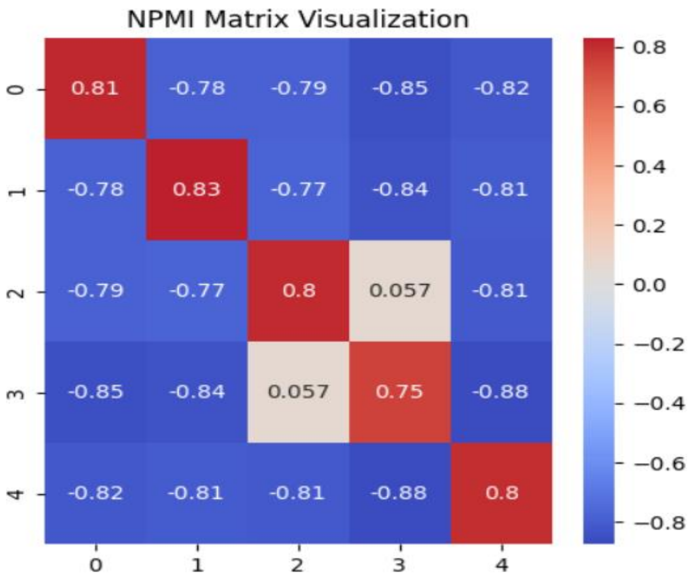

Fig. 17. Hit map of NPMI matrix visualization

Fig. 17 shows these NPMI values as a matrix, giving a more lucid depiction of word connections in tweets about disasters. The heat map presented in the figure illustrates the degree of co-occurrence between words using colour gradients. More positive relationships are shown by redder regions, whilst more negative associations are shown by bluer areas. Words are fully associated with one another in the diagonal elements (about 0.8), which exhibit strong self-association. The connections between multiple words are shown by off-diagonal components. A weak co-occurrence, for example, is indicated by a value of -0.85 between words 0 and 3. The contextual linkages between words are understood by this visualization, which aids in the identification of important word correlations, improving feature selection, and strengthening disaster tweet classification models. When paired with graph characteristics, the statistical foundation for word association provided by the addition of NPMI can result in a tweet categorization that is more precise. For example, the words "rescue" and "shelter" may appear with the word "earthquake" in tweets following a tragedy. Our approach is better able to classify tweets into important classes—like differentiating between crisis and non-disaster tweets—by recognizing these relationships.

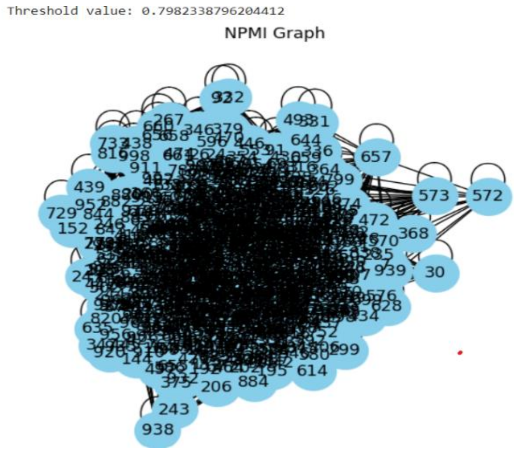

Fig. 18. NPMI Graph construction where node size is 1000

For a deeper exploration of these word associations, Fig. 18 builds an NPMI network graph, building on the insights obtained from the NPMI matrix. This graph highlights important relationships between words using 1000 nodes and a threshold NPMI value of 0.798. A feature or word in the dataset is associated with each node in the graph, which is represented by a number. Significant links between nodes are shown by their edges, with larger edges denoting stronger connections. Strong mutual links between the nodes are implied by the tight cluster at the centre, which points to a strongly linked set of nodes. The strength and structure of interactions within a dataset are probably visualised using this graph, where higher NPMI scores between nodes indicate more meaningful co-occurrences or correlations.

The reason NPMI was selected is that it can detect important and subtle co-occurrence associations, which are important for deciphering semantic linkages in tweets about disasters. NPMI directly reflects the co-occurrence strength adjusted for word frequency, providing a more nuanced understanding of word relationships in sparse textual data than other methods like Cosine Similarity (which measures the angle between vectors representing word pairs) or Jaccard Similarity (which takes the overlap between sets into consideration).

Another option, Mutual Information (MI), may be skewed towards often occurring word pairings; however, NPMI corrects this by normalizing the metric and lessening the influence of frequently occurring terms. Hence, NPMI offers a fair and comprehensible measure that complements the objectives of graph-based text analysis in this investigation.

5) Graph Convolution Operation

By combining data from its neighbors, the graph convolution procedure changes each node's (or tweet's) feature representation. Graph convolution can assist in capturing the context and significance of nearby tweets in the context of catastrophe tweets, allowing one to gauge the urgency or severity of the current situation. We use the NODE-SELECT algorithm, which collects data from nearby nodes according to a selectivity score, to improve the performance of the GCN.

Algorithm: Structure for Using NODE-SELECT in Disaster Tweet Management

Input:

- The input graph G= (V, E) shows the relationship between the tweets.
- Neighbourhood of node N (v_i): The collection of

nodes (tweets) for node v_i

- input features x_i of node v_i : features of tweet v_i
- Layer L € [1, L]: The GCN'S number of layers
- T as the threshold value:

Output:

- for tweet I. Use NODE-SELECT embedding h_i^l

Procedure:

Step 1. Set up the embedding $h_i^0 \leftarrow x_i$ //At first, tweet v_i is not chosen.

Step 2. For every layer l, ranging from 1to *L*

- Initially, assign the embedding $h_i^l \leftarrow x_i$. Determine the tweet's initial embedding, *i*.
- Determine the selectivity score, $\widehat{P}_i$ for the tweet v_i using its neighbourhood N_i

If $\widehat{P}_i \geq T$

Assign *S* (v_i) ← 1 //tweet v_i chosen

End if

- Combine texts from a chosen selection of nearby tweets A_i
- Make changes to the embedding. $h_i^l \leftarrow h_i^l + A_i$
- Normalize: $h_i^l + h_i^{l-1}$
- Please send back the completed embedding.

End of For loop

Step 3. The final embedding h_i^L € R^F should be returned.

The convolution operation can be mathematically described as:

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^L\omega^L)$$

$\widetilde{A} = A + I$ is the adjacency matrix containing self-loop

$\widetilde{D}$ represents the diagonal node degree matrix of $\widetilde{A}$

H^l indicates the feature matrix at layer L.

w^L learnable weight matrix at layer L and $\boldsymbol{\sigma}$ indicates activation function ReLu.

6) Layer-wise Propagation

To gradually improve twitter embeddings, our method makes use of multi-layer GCNs. To update the node features, a graph convolution is performed by each layer. The initial feature matrix X serves as the input for the first layer, and each layer's output feeds into the one after it. The fig.19.illustrates the information's layer by layer communication.

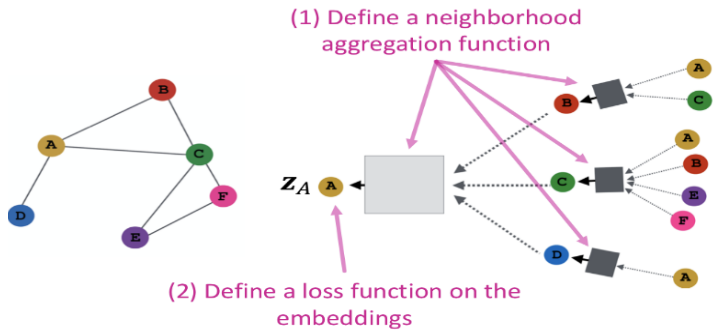

Fig. 19. Graph convolution at each layer GCN [17]

Neural networks are shown as grey boxes in the illustration above. Every colored circle is a tweet. Links among tweets determined by several factors (e.g., co-occurrence of keywords, semantic similarity). Data is compiled from nearby tweets B, C, D, E, and F for the center tweet A. Information from nearby tweets B, C, D, E, and F is combined into the core tweet A. In order to do this, a neighborhood aggregation function is used, which may perform operations such as summing or averaging the feature vectors of the nearby tweets. The aggregation algorithm improves and becomes more context-aware in the depiction of the central tweet by incorporating data from neighboring tweets. The main tweet (z_A) the computation of A's embedding involves merging the aggregated data from its neighbors with its own characteristics. To train the GCN, these embeddings are subjected to a loss function. The objective is to minimize prediction error and accurately characterize each tweet as either linked to a tragedy or not.

7) Polling Operation

The pooling function is used to aggregate the feature vectors of individual tweets (nodes) into a single feature vector that represents the complete collection of tweets (graph) while processing tweets connected to disasters. Then, using this aggregated feature vector, tasks like as identifying the type of catastrophe or categorizing the general sentiment of the tweets may be performed. The following are mathematical expressions for frequently used pooling or readout functions:

A. Global Mean Polling

By using this pooling function, the average of the node characteristics throughout the whole network is calculated:

$$h_G = \frac{1}{|v|} \Sigma_{i \epsilon v} h_i$$

B. Global Max Pooling

For every feature dimension, global max pooling determines the largest value across all graph nodes:

$$h_G = nax_{i \in v} h_i \cdot$$

v is the graph's collection of all nodes.

C. Global Sum polling

The total of the node characteristics throughout the whole network is calculated via global sum pooling:

$$h_G = \Sigma_{i \in \nu} h_i$$

$\Sigma_{i \in \nu} h_i$indicates the total number of node's feature vectors.

3.5 The Hybrid Model

Fig. 20 showcases a hybrid model that is especially designed for tweet categorization linked to disasters, fusing classic TF-IDF-based word embeddings with Graph Convolutional Networks (GCNs). A vector representation of every word in the dataset is created using Term Frequency-Inverse Document Frequency (TF-IDF). This approach downplays terms that are prevalent throughout the dataset and highlights words that are more significant in a particular text (tweet). In this model, words (W1, W2..., Wn) are shown as nodes in a graph, while the connections between their co-occurrences are represented by edges. Metrics that capture both syntactic and semantic relationships between words, such as Normalized Pointwise Mutual Information (NPMI), are used to find the edges. The embeddings produced by the GCN from these interactions, which capture contextual word information, then explain the network structure. Next, a GCN is applied to this graph in order to produce word embeddings that combine the global context of word usage throughout the dataset with the structural information of the word graph. Based on the co-occurrence of words, the GCN analyses the network to create embeddings that represent the interconnection of words.

We used many measures to guarantee the consistency and dependability of the model, particularly in light of worries about unpredictability brought on by random weight initialization in the GCN. To ensure consistent random weight initialization across runs, we first fixed a random seed at the start of our function. In order to further reduce variability and provide a more accurate evaluation of the model's performance, we also used k-fold cross-validation. Finally, in order to provide a more accurate representation of the model's performance distribution, we repeated the training procedure using several random seeds and reported the average accuracy and standard deviation. Subsequently, the GCN embeddings are fused with the TF-IDF vectors, which represent the word-level local relevance in each tweet. After that, a Fully Connected Neural Network (FCNN) is fed this concatenated embedding vector, which captures both the local importance of words (from TF-IDF) and their global associations (from GCN). With a 0.001 learning rate and a 0.05 dropout rate during training, the FCNN additionally employed the Adam optimizer. The final classification is carried out by the FCNN, and a softmax function makes sure that the total of the output probabilities equals 1. The hybrid model makes use of both the term significance measured by TF-IDF and the relational knowledge of terms from the GCN. Due to its dual approach, which takes into account both the relevance of individual words and structural linkages in text categorization tasks, the model is able to be more resilient and accurate.

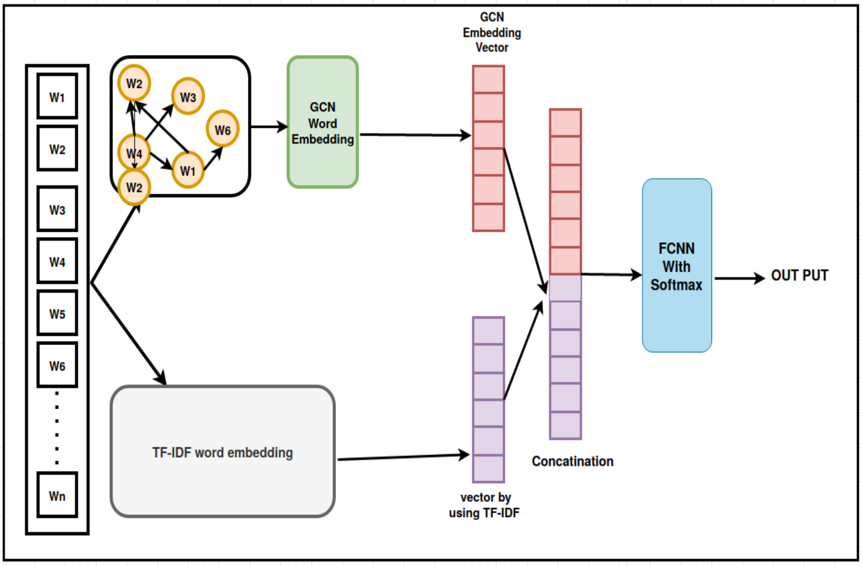

Fig. 20. Hybrid embedding approach to classify crises related short text

4. Result Analysis

Metrics for performance evaluation are essential for assessing a model's efficacy on test data and provide information about how well it operates in actual situations. These metrics work as standards for evaluating the production version's functioning. The accuracy, precision, recall and F1 score are few performance metrices. Fig. 21 shows the performance metrics of GCN and hybrid classifier models in the form of bar graph. The score for each measure is shown as a bar height, which highlights how Hybrid Model typically outperforms GCN in all categories. The confusion matrix for the GCN and hybrid models are shown in figs. 22 and 23 respectively. Fig. 24 shows the ROC curve for each of the various models.

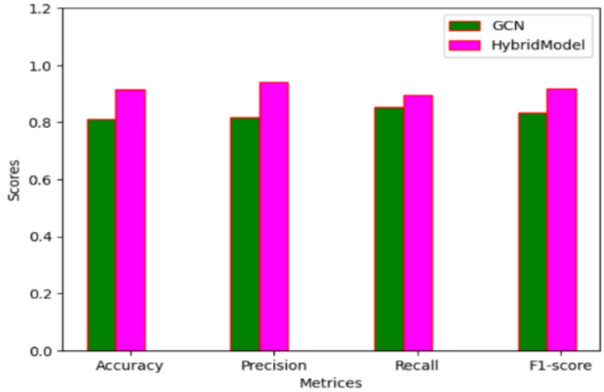

Fig. 21. Performance metrices and scores of GCN and Hybrid model

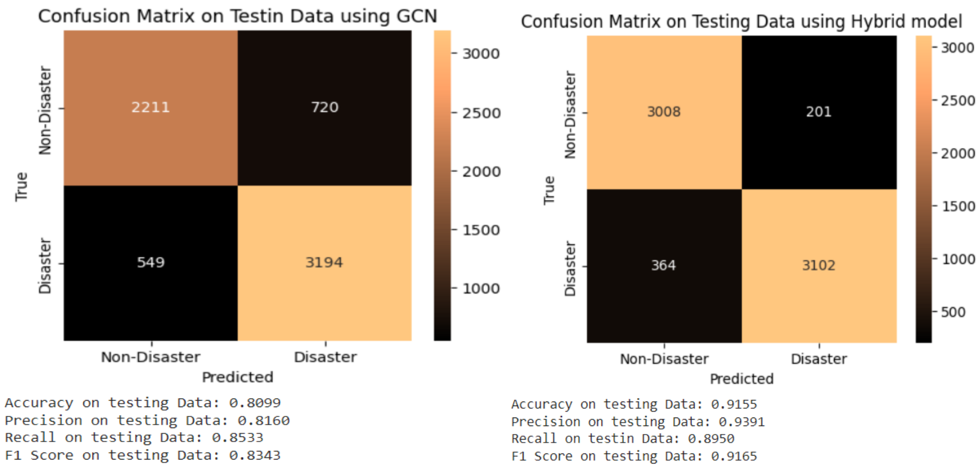

Fig. 22. Confusion matrices of GCN and Hybrid model

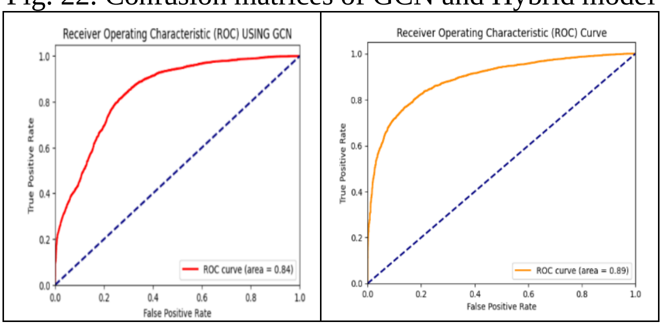

Fig. 23. ROC curves of GCN and Hybrid Model

TABLE I PERFORMANCE METRICS OF THE GCN AND HYBRID

Models	Precision	Recall	F1score	Accuracy
GCN	0.8160	0.8533	0.8343	80.99

Hybrid Model (TF-IDF & GCN)	0.9391	0.8950	0.9165	91.55

5. Conclusion

To classify tweets connected to disasters, we presented a hybrid model that incorporates Graph Convolutional Networks (GCN) and Term Frequency-Inverse Document Frequency (TF-IDF). The outcomes show there is a notable increase in classification accuracy both for structural and textual information. In this model the textual elements are combined with graph-based associations to evaluate the social media data in the context of crisis and helps in management of the crisis. In order to facilitate prompt and efficient disaster response, the study has mainly concentrated on improving the categorization of tweets pertaining to catastrophe situations. The algorithm can contribute to a better knowledge of the public communication landscape during catastrophes by precisely identifying these tweets, which will eventually help with more effective resource allocation and disaster management. The shortcomings of the present work are that there are several restrictions such as the temporal dynamics of tweet patterns during disasters, which might provide important insights into how public discourse and information transmission change over time. Furthermore, although TF-IDF and GCN are successfully combined in the hybrid model, there is room for improvement in the way these methods are integrated or in the alternative preprocessing procedures that might be investigated to improve model performance.

References

1. Acikara, T., Xia, B., Yigitcanlar, T., & Hon, C. (2023). Contribution of social media analytics to disaster response effectiveness: A systematic review of the literature. Sustainability, 15(11), 8860.
2. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.
3. Danday, A., & Murthy, T. S. (2022). “Twitter data analysis using distill BERT and graph-based convolution neural network during disaster”.
4. Devaraj, A., Murthy, D., & Dontula, A. (2020). “Machine-learning methods for identifying social media-based requests for urgent help during hurricanes”. International Journal of Disaster Risk Reduction, 51, 101757.
5. Imran, M., Mitra, P., & Castillo, C. (2016). Twitter as a Lifeline: Humanannotated Twitter Corpora for NLP of Crisis-related Messages. arXiv preprint arXiv:1605.05894.
6. Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.
7. Li, H., Caragea, D., & Caragea, C. (2021). Combining self-training with deep learning for disaster tweet classification. In The 18th international conference on information systems for crisis response and management (ISCRAM 2021).
8. Linmei, H., Yang, T., Shi, C., Ji, H., & Li, X. (2019). Heterogeneous graph attention networks for semi-supervised short text classification. In Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp. 4821-4830.

9. Liu, P., Qiu, X., & Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101.
10. Liu, J., Zeng, D., Guo, R., Lu, M., Wu, F. X., & Wang, J. (2021). Mmhge: detecting mild cognitive impairment based on multi-atlas multi-view hybrid graph convolutional networks and ensemble learning. Cluster Computing, 24, pp.103-113.
11. Paul, N. R., Sahoo, D., & Balabantaray, R. C. (2022). VocabGCN-BERT: A Hybrid Model to Classify Disaster Related Tweets. In 2022 OITS International Conference on Information Technology (OCIT), IEEE, pp. 55-60.
12. Pratama, R. A., & Pardede, H. F. (2023). Disaster tweetclassifications using hybrid convolutional layers and gated recurrent unit. International Journal of Computing and Digital Systems, 14(1), pp. 1-1.
13. Mikolov, T. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
14. Sudha, S. B., & Dhanalakshmi, S. (2023). Sentiment analysis using various machine learning algorithms for disaster related tweets classification. International Journal of Intelligent Engineering Informatics, 11(4), pp. 390-417.
15. Wang, P., Hu, J., Zeng, H. J., & Chen, Z. (2009). Using Wikipedia knowledge to improve text classification. Knowledge and Information Systems, 19, pp. 265-281.
16. Yao, L., Mao, C., & Luo, Y. (2019). Graph convolutional networks for text classification. In Proceedings of the AAAI conference on artificial intelligence, Vol. 33, No. 01, pp. 7370-7377.
17. https://snap-stanford.github.io/cs224w-notes/machine-learning-with-networks/graph-neural-networks