Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.073486

ARTICLE

Multi-Task Disaster Tweet Classification Using Hybrid TF-IDF and Graph Convolutional Networks

Basudev Nath[1], Deepak Sahoo[1], Sudhansu Shekhar Patra[2], Hassan Alkhiri[3], Subrata Chowdhury[4], Sheraz Aslam[5,6,*] and Kainat Mustafa[7]

[1]Faculty of Engineering Technologies, Sri Sri University, Cuttack, India
[2]School of Computer Applications, KIIT Deemed to be University, Bhubaneswar, India
[3]Department of Computer Science, Faculty of Computing and Information, Al-Baha University, Al-Baha, Saudi Arabia
[4]Department of Computer Science and Engineering, Sreenivasa Institute of Technology Management Studies (A), Chittoor, India
[5]Department of Computer Science, American University of Cyprus, Larnaca, Cyprus
[6]Department of Computer Science, CTL Eurocollege, Limassol, Cyprus
[7]Dpoint Technologies Ltd., Limassol, Cyprus

*Corresponding Author: Sheraz Aslam. Email: aslam.sheraz@aucy.ac.cy

Received: 19 September 2025; Accepted: 12 January 2026; Published: 12 March 2026

ABSTRACT: Accurate, up to date, and quick information related to any disaster supports disaster management team/authorities to perform quick, easy, and cost-effective response to enhance rescue operations to alleviate the possible loss of lives, financial risks, and properties. Due to damaged infrastructure in disaster-affected areas, social media is the only way to share/ exchange real time information. Therefore, 'X' (formerly Twitter) has become a major platform for disseminating real-time information during disaster events or emergencies, i.e., floods and earthquake. Rapid identification of actionable content is critical for effective humanitarian response; however, the brief and noisy nature of tweets makes automated classification challenging. To tackle this problem, this study proposes a hybrid classification framework that integrates term frequency–inverse document frequency (TF-IDF) features with graph convolutional networks (GCNs) to enhance disaster-related tweet analysis. The proposed model performs three classification tasks: identifying disaster-related tweets (achieving 94.47% accuracy), categorizing disaster types (earthquake, flood, and non-disaster) with 91.78% accuracy, and detecting aid requests such as food, donations, and medical assistance (94.64% accuracy). By combining the statistical strengths of TF-IDF with the relational learning capabilities of GCNs, the model attains high accuracy while maintaining computational efficiency and interpretability. The results demonstrate the framework's strong potential for real-time disaster response, offering valuable insights to support emergency management systems and humanitarian decision-making.

KEYWORDS: Natural language processing; tweet classification; graph neural networks; deep learning

1 Introduction

Social media has become a crucial medium for communication during emergencies, with X (formerly Twitter) enabling users to share real-time updates during crises such as floods and earthquakes [1]. These posts often contain vital information for emergency responders; however, their volume, brevity, informality, and noise make classification challenging [2,3]. Prior studies in crisis informatics and emotion prediction highlight the need for intelligent systems to process such data efficiently [4,5]. Effective disaster-response

pipelines typically require identifying disaster-related tweets, categorizing them by disaster type, and extracting urgent support needs such as food, medicine, donations, and infrastructure assistance.

Traditional machine learning (ML) techniques—such as SVM, CRF, and Naive Bayes—paired with representations like BoW [6] and TF-IDF [7] have been widely applied to classify disaster-related tweets [8–10]. These methods rely on hand-crafted features (e.g., unigrams, bigrams, POS tags, hashtags, tweet length) and have shown effectiveness with improvements reported through feature enhancement techniques [8,10]. However, they tend to produce sparse high-dimensional representations and struggle with noisy and dynamic data. Deep learning (DL) models, including CNN [11], LSTM [12], and hybrid CNN-LSTM architectures [13], have demonstrated improved performance; for example, CNNs have outperformed earlier ML methods for disaster tweet detection [14]. Transformer-based models such as BERT [15] and CrisisBERT based on DistilBERT [16] further enhance classification but may be limited in capturing deeper structural dependencies.

Graph neural networks (GNNs) have recently gained interest due to their ability to represent unstructured text through relational structures [17–19]. GNNs have been successfully applied to various NLP tasks including sequence labeling, translation, and text categorization [20–23]. Models such as GCN-based TextGCN [24] and related work [25] conceptualize text classification as node classification using corpus-level graphs constructed from TF-IDF and PMI. Other contributions include SHINE [26], which employs heterogeneous hierarchical graphs; TEXTING [27], which creates inductive per-text graphs; InduT-GCN [28], which builds training-based graphs with unidirectional GCN propagation; and STGCN [29], which introduces thematic corpus-level graphs enhanced with BiLSTM embeddings. Attention-enabled GNNs—such as GAT [30], HGAT [31], and diffusion-based GNNs [32] further improve structural modeling. Additional models combine GNNs with LSTM or transformer features [33] or integrate BERT embeddings with GCNs, such as VOCABGCN-BERT [34]. While these methods (a detailed overview of all models is given in Table 1) advance short-text representation, they predominantly focus on either structural or contextual signals, rarely integrating statistical representations like TF-IDF with graph-based relational learning for disaster tweet classification and support-need extraction.

Table 1: Summary of representative models for disaster tweet classification. [Note: M = manual, A = automatic, SA = aemi automatic, S = sparse, D = dense].

Approach Type	Model	Learning Type	Key Features	Feature Extraction	Representation	Task
Traditional ML	SVM, Naive-Bayes, CRF	NA	Hand Crafted features (Unigrams, POS, tags, hash tags)	M	S	Tweet Classification
Deep Learning (DL)	CNN, LSTM	NA	Automatically learned spatial and temporal features	A	D	Disaster tweet detection
Hybrid DL	CNN+LSTM	NA	Combines spatial and sequential dependencies	A	D	Tweet categorization

(Continued)

Table 1 (continued)

Approach Type	Model	Learning Type	Key Features	Feature Extraction	Representation	Task
Transformer Models	BERT, DistilBERT, CrisisBERT	NA	Contextual embeddings with attention	A	D	Disaster tweet identification
Graph-Based Models (GNN)	TextGCN, SHINE, STGCN	Transductive/ Inductive	Word and document graphs capturing structure	A	D	Short text classification
Attention-Based GNNs	GAT, HGAT	Transductive	Self-attention and hierarchical graph features	A	D	Tweet classification
Proposed Hybrid Model	TF-IDF + GCN	Inductive	Combines statistical and structural features	SA	D	Context-aware tweet classification

To address this gap, the present work introduces a hybrid approach that integrates TF-IDF with a GCN-based architecture to jointly capture statistical term significance and relational word dependencies. TF-IDF vectors are combined with embeddings learned from an NPMI-based tweet–word co-occurrence graph, and the fused representation is used by an FCNN for classification. The study addresses three tasks: (1) classifying tweets as disaster or non-disaster; (2) categorizing disaster tweets by type (flood, earthquake); and (3) identifying specific support needs in earthquake-related tweets. The study is guided by the following research questions:

RQ1 Can the integration of TF-IDF and GCN outperform TF-IDF-only or GCN-only approaches?

RQ2 Can the proposed model maintain strong performance across multiple levels of disaster-related classification?

RQ3 Does combining statistical and relational cues enhance classification, especially for nuanced need extraction?

To explore these questions, a multi-stage framework is developed: tweets are preprocessed and converted into TF-IDF vectors. At the same time, an NPMI-based co-occurrence graph is processed through a GCN to generate structural embeddings. A pooling operation is then deployed to generate fixed-length vectors that are concatenated with TF-IDF features. Finally, an FCNN performs classification across all tasks. This combined strategy offers a more context-aware and structurally informed model for disaster tweet analysis.

Section 2 provides the description of the proposed model, including TF-IDF, GCN vectorization, and training the model. The results of the experiments are presented in Section 3. Finally, the concluding remarks and ideas for future research are in the Section 4.

2 Description of the Proposed Model

The proposed framework is composed of seven interconnected stages organized within a cascaded multi-task learning architecture. In steps 1 and 2, tweets are collected and preprocessed. Step 3 converts the cleaned text into numerical representations using TF-IDF vectorization. In steps 4 and 5, a GCN is applied to a word co-occurrence graph constructed using NPMI to learn graph-based embeddings. These learned representations are merged and provided as input to a shared FCNN in step 6. Finally, step 7 applies a sequential three-level classification process consisting of the following tasks:

- Task 1 uses a binary classification, to ascertain whether or not a tweet is related to a disaster.
- Task 2 identifies the type of disaster (such as non-disaster, earthquake or flood).
- Task 3 extracts additional information about support from tweets that have been classified according to the disaster type. The aid may be food, medicine, donations, or infrastructure assistance.

The following subsections provide a full description of each component's role and purpose, as well as how each one aids in effective tweet analysis related to disasters.

2.1 Tweet Gathering/Dataset Description

This study uses two dependable datasets: CrisisNLP [35] and CrisisLex [36]. These data sources include annotated tweets on a variety of real-world tragedies. A total of 33,370 tweets connected to disaster and non-disaster messages were collected. Task 1 is a binary classification, in which entire tweets are divided into two classes: disaster (Target = 1) and non-disaster (Target = 0), as depicted in Table 2. Furthermore, Task 2 involves creating a dataset that contains 15,435 tweets pulled from the dataset used in Task 1, presented in Table 3. After a tweet has been selected as disaster-related in Task 1, then it moves to Task 2 to be further categorized as a particular disaster type, i.e., earthquake or flood. Nonetheless, a balanced subset of non-disaster tweets is inserted to enhance textual discrimination. This approach avoids the model confusing general tweets with disaster-related tweets that might bear similar linguistic indicators. In addition, in Task 1, few tweets can be mistakenly registered as disaster-related (false positives). Therefore, their inclusion in Task 2 will enable the framework to reevaluate the ambiguous cases and ensure that it is robust enough to separate the real disaster-related and irrelevant text. As a result, Task 2 comes with earthquake, flood, and non-disaster categories in order to improve classification accuracy and model extrapolation. Consequently, Task 2 comprises three approximately balanced groups: earthquake (5779 tweets), flood (4873 tweets), and non-disaster (4783 tweets).

Table 2: Task 1 dataset.

Tweet_ID	Tweet_text	Target
'591903085670215681'	RT @USER: These Baltimore niggers should move to LOCATION	0
'591903104276234242'	Itvnews: URL to LOCATION #HASHTAG tells itvnews: 'It was terrifying'	1
'591903131505659904'	Absolutely#@ devastated by the destruction to my old home #HASHTAG	1

The resulting balanced multiclass dataset is organized and distributed, as shown in Table 3. Task 3 is based on 5779 earthquake-related tweets obtained as a result of Task 2, which is labelling them within aid categories, including food, medical help, infrastructure, and utilities, as presented in Table 4. Additional categories, i.e., sympathy, and support, and general information, are non-aid content commonly noticed

during actual disaster communication. These categories assist the model in differentiating actionable requests from emotional expressions or general discourse. Several examples of relevant tweets are in Table 4. To protect the privacy of the users, usernames, URLs, locations, and hashtags in all the tweets were anonymized by substituting them with generic tokens. Tasks 1 and 2 include Tweet_ID, Tweet_text, and Target, whereas Task 3 includes an additional feature, Help, which identifies the kind of help that is requested.

Table 3: Task 2 dataset.

Tweet_ID	Tweet_text	Target
'297111336012369921'	Yum in so many ways beleza espresso bar	0
'297191572838178816'	LOCATION is flood damage by the numbers	1
'297194979191828480'	Itvnews witness LOCATION earthquake tell itvnews terrifying	2

Table 4: Task 3 dataset about earthquake.

Tweet_ID	Tweet_text	Target	Help
'600463106431528960'	Prayer for LOCATION	2	Sympathy and support
'592126024437125121'	Thought prayer everyone LOCATION	2	Sympathy and support
'593819547372638208'	Obtains powerful image baby rescued least hour LOCATION earthquake hit	2	Food and Medicine

2.2 Preprocessing Tweets

Before extracting features from the data, we performed preprocessing in the following steps: **Text normalization:** For consistency, all tweet text is converted to lowercase. To reduce noise that could negatively impact model performance, punctuation marks, special characters, and numerical values are removed. **Tokenization:** Tokenization involves splitting each tweet into individual units, referred to as tokens. This step breaks sentences into smaller components, allowing the model to analyze textual data more effectively. **Avoiding Stop Words:** Common words such as "and", "the", and "is" which do not contribute meaningful information for tweet classification, are removed. Eliminating these terms enables the model to concentrate on more informative and discriminative words. **Word Stemming:** To normalize variations of the same word, the Porter stemming algorithm is applied. This process reduces words to their root forms (e.g., "flooding" to "flood"), helping the model capture semantic similarity across different word inflections. Table 5 presents examples of original tweets alongside their cleaned versions and corresponding target labels.

Table 5: Tweets with cleaned tweets.

Tweet_ID	Tweet_Text	Cleaned_Tweet	Target
'591903085670215681'	RT @user: These Baltimore niggers should move to L...	rt user baltimor nigger move LOCATION	0
'591903104276234242'	Itvnews: URL to LOCATION #hashtag tells itvnews:...	Itvnew url LOCATION hashtag tell itvnew terrifi	1

(Continued)

Table 5 (continued)

Tweet_ID	Tweet_Text	Cleaned_Tweet	Target
'591903131505659904'	Absolutely#@ devastated by the destruction to my o...	Absolutely#@ devastated by the destruction to my o..	1

Table 6 shows a comparison of tweets before and after the cleaning process, along with their lengths. It clearly shows the importance of removing unnecessary features like, links, symbols, and special characters, which make tweets shorter and more organized. Tweets containing numerical values may include useful information; however, such values often appear inconsistently or in unusual contexts, which can confuse the model. Therefore, numerical values were replaced with a generic token to preserve their presence while improving textual consistency. The Porter stemming algorithm was selected because it is simple to implement and performs well on short, informal Twitter text. Although lemmatization preserves more semantic meaning, it requires accurate part-of-speech tagging, which is difficult to achieve with noisy and unstructured Twitter data. Consequently, stemming provides a practical and efficient alternative.

Table 6: Length of tweets before and after cleaning.

Tweet_ID	Tweet_Text	Tweet Length	Cleaned Tweet	Cleaned Tweet Length	Target
'297111336012369921'	Yum in so many ways beleza espresso bar	39	Yum mani way beleza espresso bar	6	0
'297191572838178816'	LOCATION is flood damage by the numbers	41	LOCATION flood damage number	4	1
'297194979191828480'	Itvnews witness LOCATION earthquake tell itvnews terrifying	57	Itvnew wit LOCATION earthquake tell itvnew terrifi	7	2

2.3 TF-IDF Vectorization of Words

The textual data undergoes a pre-processing and cleaning; then, data are converted into numerical format that can be used in machine learning. In order to reflect the contextual relevance and meaning of words, this research opts to apply the TF-IDF methodology as a hybrid embedding model with Graph Convolutional Networks (GCN). TF-IDF is a measure of the level of significance of a word within a particular tweet in comparison to the entire corpus. It emphasizes distinctive and informative terms while reducing the impact of frequent but less meaningful words. Term Frequency (TF) is the frequency of the occurrence of a word within a tweet whereas the Inverse Document Frequency (IDF) is the frequency of a word in all the tweets. Mathematically, TF-IDF is written as:

$$TF - IDF(t_i, d_j) = TF(t_i, d_j) x IDF(t_i)$$

where

$$TF(t_i, d_j) = \frac{f_{i,j}}{\sum_k f_{k,j}}$$

and

$$IDF(t_i) = \log\left(\frac{N}{\mathrm{df}_i + 1}\right)$$

Here $f_{(i,j)}$ denotes the raw frequency of the term t_i in tweet d_j, df_i represents the number of tweets in which the term t_i occurs, and N is the total number of tweets. This representation allows the model to emphasize informative words, thereby improving its ability to identify critical disaster-related content. As a result, each tweet is represented as a fixed-length sparse vector, where non-zero entries correspond to highly ranked terms and their associated relevance scores. The top twenty terms in the dataset are illustrated in Fig. 1, which is based on the average TF-IDF scores assigned to each word across the entire corpus. Examples of terms with higher average TF-IDF values include "nepal," "earthquake," and "rubyph," indicating their strong association with tweet-specific content. The objective of this analysis is to provide initial insight into prominent topics by highlighting statistically significant terms that effectively distinguish tweets within the dataset. The selection parameter $k = 20$ was determined empirically through experimentation with values of $k = 20, 50, 100$, and 200 features. Increasing k beyond 20 resulted in a classification accuracy improvement of less than 1%, while considerably increasing model complexity and training time. Given the limited length of tweets, a compact set of representative terms was sufficient to capture meaningful vocabulary patterns. Consequently, selecting $k = 20$ achieves a suitable balance between accuracy, computational efficiency, and representational adequacy. Sensitivity analysis confirms that the reduced feature space retains essential lexical information without introducing redundancy.

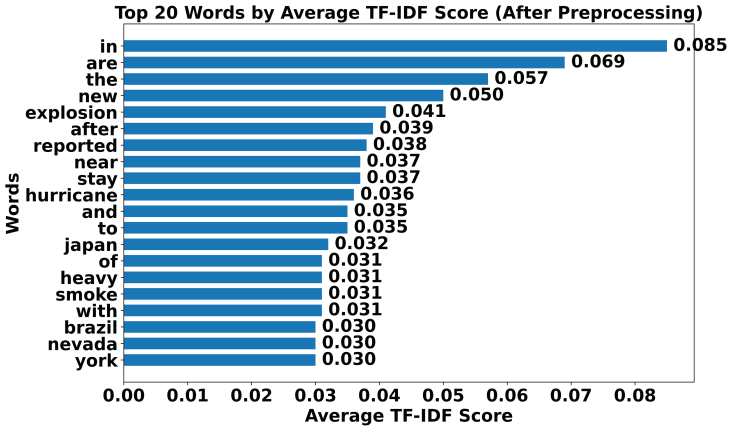

Figure 1: Distribution of TF-IDF terms in disaster tweets.

While TF-IDF effectively highlights the importance of individual words, it does not account for word co-occurrence patterns, which may result in a loss of contextual information. To address this limitation, a graph-based embedding approach is introduced that constructs a word co-occurrence graph to model contextual relationships between terms. This allows word dependencies to be represented more comprehensively before classification.

2.4 Using NPMI to Build a Graph

We use NPMI to make a graph of word co-occurrences so that our embeddings can include a global perspective. This process includes:

2.4.1 Making Co-Occurrence Matrices

A co-occurrence matrix records how frequently words appear together within a given context. The co-occurrence relationships are identified using a sliding window approach with a window size of five. Accordingly, each word is analyzed together with its four neighboring words on both sides. This ensures that meaningful term relationships are captured beyond individual tweet boundaries. Let W denote the vocabulary size, representing the number of unique words in the dataset, and let D denote the number of tweets. The co-occurrence matrix $M \in R^{W \times W}$ is defined as follows:

$$M_{ij} = \sum_{t \in D} \prod (\omega_i \in t \quad \wedge \quad \omega_j \in t) \tag{1}$$

where: M_{ij} tells us how many times the words ω_i and ω_j appear together in one tweet. $\prod \left(\omega_i \cdot \omega_j\right) \in t$, indicator function which gives a value of 1 if both words appear in tweet t and 0 if either is absent. Each tweet t in dataset D is taken into account in the summation. Although TF-IDF effectively captures important terms, it is unable to model word dependencies within tweets. To overcome this limitation, normalized pointwise mutual information (NPMI) is employed to quantify the strength of association between words based on their co-occurrence frequencies across the corpus. This enhancement is crucial because contextual word relationships play a significant role in disaster tweet classification. For example, closely related terms such as "earthquake" and "Nepal" may not be sufficiently represented by TF-IDF alone. NPMI facilitates the construction of a word association graph, enabling a richer and more comprehensive representation of textual data.

2.4.2 Calculating NPMI for Edge Weights

After making the co-occurrence matrix, this study uses NPMI to measure the strength of connection between word pairs. NPMI is different from basic co-occurrence counts because it looks at how often two words occur together compared to how often they occur separately. This stops the graph from being filled with terms that are used a lot. The NPMI score for the two words, w_i and w_j is calculated as:

$$NPMI(\omega_i, \omega_j) = \frac{\log\left(\frac{P(\omega_i, \omega_j)}{P(\omega_i) \cdot P(\omega_j)}\right)}{-\log P(\omega_i, \omega_j)} \tag{2}$$

where the probability that the words ω_i and ω_j will appear together in a single tweet is $P(\omega_i, \omega_j)$.

The probability of the word ω_i is $P(\omega_i)$ and ω_j is $p(\omega_j)$ in the dataset.

The NPMI can be anywhere from −1 to 1. If NPMI is zero, it means that ω_i and ω_j are not related to each other. NPMI = 1 means that the two words are always present together, while NPMI = −1 means that the two words don't exist together as often as expected. Using NPMI helps to keep only the word pairs with strong connections, which makes sure that the edges in the graph show semantic relationships instead of just random co-occurrences.

2.4.3 Setting a Threshold for Graph Sparsity

After calculating NPMI scores for word pairs, the next step is to use a thresholding mechanism to keep only the important connections in the graph. If there is no threshold, the adjacency matrix can become very dense, which can make calculations more expensive and lead to overfitting in later graph-based models like GCNs. To make a sparse adjacency matrix, a threshold T is set to get rid of low NPMI values. Formally, this is expressed as:

$$A_{ij} = \begin{cases} 1, & \text{if NPMI}(\omega_i, \omega_j) > T \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Here A_{ij} means that there is an edge between the words ω_i and ω_j and T is the threshold value used to get rid of weak or unimportant connections. Thresholding makes the graph to show only the most important word connections. Through testing at different threshold values, the value of T was determined to be 0.2. Lower threshold values, such as 0.1, produced graphs with dense co-occurrences and excessive noise. In contrast, higher values, starting from 0.3 and above, resulted in overly sparse graphs, making it difficult to identify meaningful relationships. The selected threshold of 0.2 provided an effective balance between graph density and lexical richness, thereby improving both model performance and training efficiency.

2.4.4 Making a Graph

Once thresholding produces a sparse adjacency matrix, a graph is constructed where words serve as nodes and meaningful relationships between them are represented as edges. The resulting graph is then used to generate graph-based word embeddings, allowing the GCN model to learn richer word representations that extend beyond individual occurrences. Unlike TF-IDF, the graph structure preserves both local (within tweets) and global (across tweets) relationships. Word embeddings are further refined through message passing between related nodes in the graph. High-dimensional TF-IDF vectors are not encoded within the graph; instead, only essential word relationships are retained. The mathematical definition of the graph $G = (V, E)$ is given as follows: Nodes (V): each node represents a unique word in the vocabulary. Edges (E): the adjacency matrix A defines the connections between words, and if $A_{ij} = 1$, there is an edge between words ω_i and ω_i.

$$\begin{aligned} G(V, E), \quad \text{with} \quad & V = \{\omega_1, \omega_2, \omega_3, \ldots, \omega_n\}, \\ & E = \{(\omega_i, \omega_j) \mid A_{ij} = 1\} \end{aligned} \tag{4}$$

We make the dataset more informative and efficient for classification tasks by transforming it into this network representation, which allows the model to embed words in a context-sensitive space.

2.5 GCN Based Vectorization

After creating a co-occurrence graph, we create word embeddings using GCN. Unlike traditional text representations like TF-IDF, which treat words as distinct features, GCN uses the graph structure to capture both local dependencies (within individual tweets) and global dependencies (across multiple tweets). This enables the model to enhance word connections based on graph connections, resulting in more context-aware embeddings that are useful for classifying tweets related to disasters. Fig. 2 illustrates the two-step GCN-based graph embedding process.

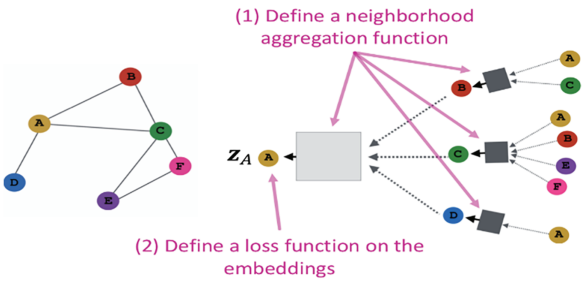

Figure 2: Graph convolutional network for word embedding.

- **Step 1: Aggregation of the neighborhood (message passing):** By combining information from its linked neighbours, each word (or node) in the network modifies its embedding. Node A in the figure receives data from its neighbours B, C, D, E, and F as part of the message-passing process. This step makes sure that each word's representation acquires more context awareness by incorporating semantic information from other words in the graph. This aggregation step is represented mathematically as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right) \tag{5}$$

The graph's adjacency matrix is represented by $\tilde{A}$, which includes self-loops to preserve some of the original semantic meaning of each word. The degree matrix, or $\tilde{D}$, is used for normalization in order to reduce bias from highly connected words. The feature matrix at layer l_1, which includes the existing word embeddings, is represented by $H^{(l)}$. A learnable weight matrix that is optimized during training is represented by $W^{(l)}$. $W^{(l)}$ represents a learnable weight matrix that is optimized throughout the training process. σ is an activation function, like ReLU, that makes the model non-linear. This step makes sure that words with strong co-occurrence relationships have an impact on each other's embeddings.

- **Step 2: Loss function for learning embeddings:** After numerous levels of message transmission in the GCN, each word is given a final embedding that more accurately represents its contextual meaning. For example, the representation of node A, indicated as Z_A in Fig. 2, is created by numerous rounds of aggregation of its adjacent nodes. To make sure that these embeddings are relevant in downstream applications like catastrophe tweet categorization, we create a loss function to direct the learning process. The most prevalent option for node classification tasks is the cross-entropy loss, which evaluates the difference between predicted and true labels. It is represented mathematically as:

$$L = \sum_{i \in Y} \text{CrossEntropy}(Z_i, Y_i) \tag{6}$$

where Y represents the set of labelled words or nodes. Z_i is the likelihood that word i belongs to the predicted class. The word's actual label is represented by Y_i. Lowering this loss function helps the model change the embeddings so that terms with similar meanings have similar representations.

2.6 Integration with the FCNN

Fig. 3 shows the hybridisation of TF-IDF and GCN. First, a method called TF-IDF converts each word in the dataset to a numerical vector. This makes words unique to a tweet more important while making frequently used words less important. In parallel, a graph representation is created, with words $(\omega_1, \omega_2, \omega_3, \ldots, \omega_n)$ acting as nodes and their co-occurrence associations acting as edges. NPMI, which aids in identifying meaningful word correlations, is used to find these relationships. This word graph is analysed by the GCN, which gathers structural connections between words as well as contextual meaning. This aids

the model in comprehending the relationships between words in different tweets. The final feature vector incorporates both global and local word associations by integrating the GCN-generated vectors with the TF-IDF feature vectors. The computed embeddings of TF-IDF and GCN are fed into the classification model once they have been concatenated into a single feature vector:

$$Z = Z_{\text{TF-IDF}} \oplus Z_{\text{GCN}}$$

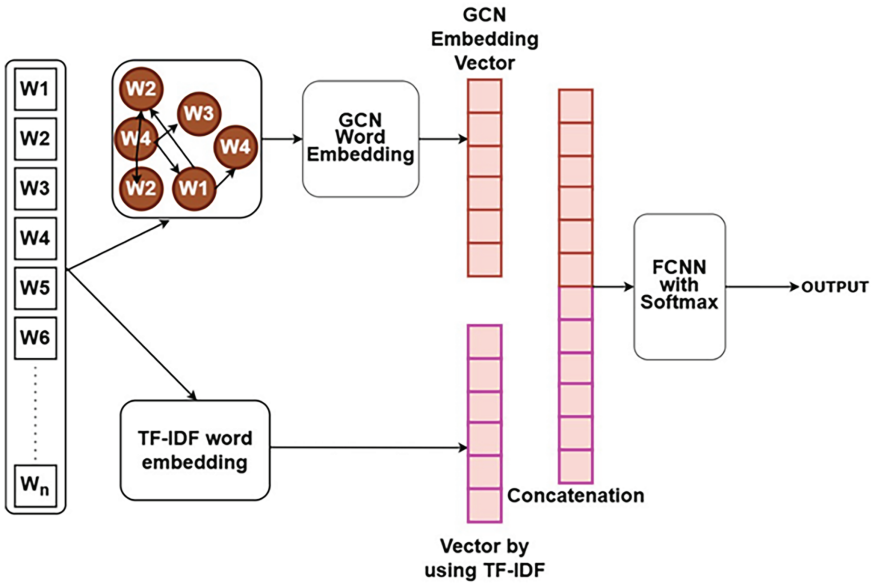

Figure 3: A hybrid model for classifying disaster tweets.

By using $\oplus$ to represent the concatenation operation, the final representation retains both local word importance (TF-IDF embeddings) and global contextual information (GCN embeddings). In order to extract high-level features, the concatenated vector Z is then fed into a FCNN, which has numerous hidden layers and ReLU activation functions. The output probabilities are guaranteed to add up to 1 by a softmax function:

$$Y = \text{Softmax}(W_{\text{FCNN}} \cdot Z + b)$$

Here b represents the FCNN's bias term, and WFCNN represents the weight matrix.

The model combines TF-IDF and GCN embeddings to find a balance between statistical word importance and structural contextual understanding. This makes it a strong and reliable way to classify tweets about disasters.

2.7 Sequential Multi-Task Classification

In the concluding phase of the suggested model, a FCNN is used to execute three sequential classification tasks. This multi-task framework enables the model to incrementally enhance predictions, beginning with general categorization and advancing to more precise insights. The tasks are structured in a hierarchical cascade, whereby the outcome of one task dictates the initiation of the subsequent activity. The input for all three tasks is the same: the concatenated vector of TF-IDF and GCN embeddings that the FCNN processes. There is a different output layer for each task, and the softmax activation for each one is based on the number of classes in that task. By letting the model learn generic patterns once and utilize them for many goals,

this shared representation increases efficiency by eliminating duplication and enhancing generalization. This step-by-step design is like how people really analyse tweets: first figure out whether they are relevant, then what kind they are, and lastly what kind of help they need. Algorithm 1 presents all the steps of our proposed model.

Algorithm 1: Hybrid algorithm of TF-IDF and GCN

Require: Dataset D of disaster-related tweets

Ensure: Classification of tweets into multiple categories depending on the task

for Each tweet $t = [w_1, w_2, \ldots, w_n]$ in dataset D, where each w_i denotes individual terms **do**
    Execute preprocessing steps: convert to lowercase, remove punctuation, delete stop words, and conduct stemming operations
    Produce a fixed-dimensional TF-IDF vector representation for tweet t
end for
Generate a word co-occurrence matrix via a sliding window technique with a size of 5
Calculate the NPMI scores for all combinations of words
Construct a graph structure $G = (V, E)$ where V comprises vocabulary phrases and E denotes connections determined by an NPMI threshold ($T = 0.2$)
Allocate initial node attributes in graph G using calculated TF-IDF representations
Implement the first GCN layer on graph G to get intermediate representations H^1
Utilize the succeeding GCN layer to produce the final word embeddings H^2 using H^1
for Each tweet t in dataset D **do**
    Merge the TF-IDF representation TFIDF_t with the appropriate GCN embeddings H_t^2 to create a unified feature vector X_t
    Input X_t into a FCNN with specified output dimensions
    Task-specific neurons:
        Two neurons for Task 1
        Three neurons for Task 2
        Five neurons for Task 3
    Utilize the softmax function to get the probability distribution $\hat{Y}_t$
end for
Enhance model parameters with the cross-entropy goal and the Adam optimization technique
Assess system performance with common categorization metrics: Accuracy, Precision, Recall, and F1-Score across all tasks

3 Results and Discussions

In order to evaluate the performance of the proposed hybrid TF-IDF + GCN model for disaster tweet classification, a series of experiments was conducted across three datasets, with each dataset corresponding to a specific task, using an 80:20 train–test split. Model performance was assessed using accuracy, precision, recall, and confusion matrices. The architecture consists of a dual-layer GCN with 128 and 64 neurons, followed by an FCNN classifier comprising two layers with 512 and 256 neurons. The Adam optimizer was employed with learning rates of 0.01 for the GCN and 0.001 for the FCNN. To mitigate overfitting and improve convergence, dropout (0.5 for the GCN and 0.3 for the FCNN), L2 regularization, batch normalization, and early stopping were applied. All experiments were executed on Google Colab using an NVIDIA Tesla T4 GPU, an Intel Xeon processor, and 12 GB of RAM. The proposed model contains approximately 2.3 M parameters, which is considerably fewer than transformer-based models such as BERT,

which has around 110 M parameters. The model achieves an average processing speed of 217 tweets per second, with an average inference time of 4.6 ms per tweet, making it suitable for real-time disaster response.

For Task 3, the synthetic minority over-sampling technique (SMOTE) was applied to balance aid-related categories, while Tasks 1 and 2 were already balanced. Overall, the hybrid model offers an effective trade-off between classification accuracy and computational efficiency, demonstrating its feasibility for real-time crisis monitoring systems. To further analyze performance, the results are presented on a task-by-task basis, allowing a clearer understanding of the model's behavior as task complexity increases.

3.1 Task 1 Disaster Identification Results

We examined our suggested method with a number of well-known baseline models in order to assess its efficacy. These models' specifics are as follows:

- For identifying sequential patterns in text, article [16] uses a Bi-LSTM that processes tweets both forward and backward.
- To enhance contextual learning, STGCN [29] treats words, topics, and tweets as connected nodes in a graph by combining Bi-LSTM with GCN.
- For classifying short texts better, HGAT [31] links words, topics, and documents using a hierarchical graph attention mechanism over an entire corpus-level graph.
- TextGCN [24] uses graph learning techniques to improve text classification by representing words and tweets as nodes in a graph structure. This article re-implemented this technique to compare with our proposed model because the original version was unavailable, which may lead to minor differences from reported findings.
- LSTM-GAT [33] integrates LSTM with Graph Attention Networks and creates a graph based on word dependencies for each text.

Table 7 provides a summary of the findings from these comparisons. With an accuracy of 94.47%, our hybrid TF-IDF+GCN model outperforms all baselines. The strength of combining statistical and structural information in disaster tweet classification is demonstrated by this significant improvement over individual models like TF-IDF (75.60%) and GCN (80.98%), as well as over other deep learning techniques.

Table 7: Each model's accuracy—Task 1.

Model	Accuracy (%)	Precision	Recall	F1-score
TF-IDF + GCN (Proposed model)	94.47	0.9398	0.9491	0.9444
TF-IDF-Only	75.60	0.7695	0.7560	0.7567
GCN-Only	80.98	0.8094	0.8098	0.8091
Bi-LSTM [16]	87.00	*	*	*
TEXTGCN [24]	80.31	*	*	*
STGCN [29]	82.31	*	*	*
HGAT [31]	80.45	*	*	*
LSTM-GAT [33]	88.40	*	*	*

Note: **Not available*.

The accuracy patterns of the GCN and Hybrid models over 50 epochs are depicted in Fig. 4. It shows that the GCN stabilizes at about 80%, and a hybrid model shows improved learning ability, reaching up to 95% accuracy. Confusion matrices showing the classification performance of the TF-IDF, GCN, and Hybrid

models are shown in Fig. 5. While TF-IDF and GCN show slightly higher values of false positives as well as false negatives, the proposed hybrid model has the lowest misclassification rate.

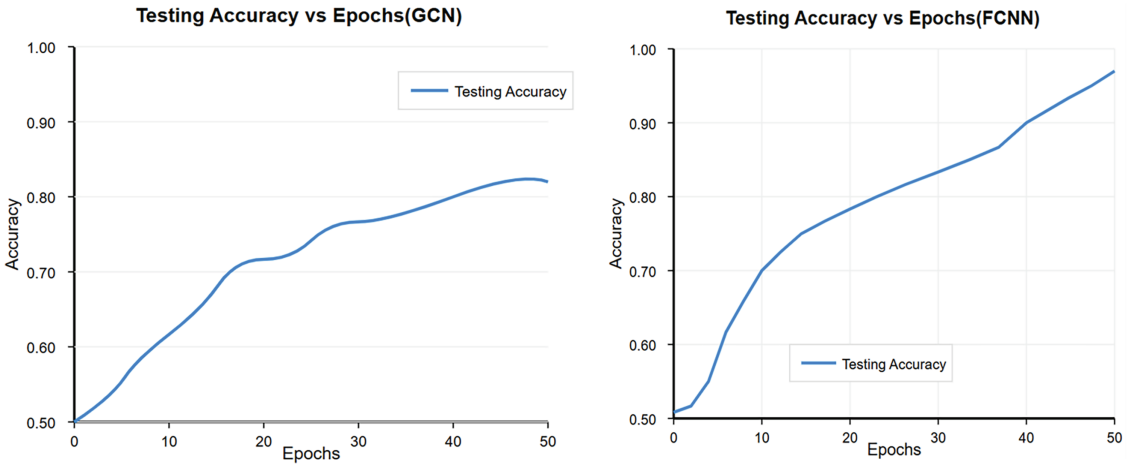

Figure 4: GCN and hybrid model testing accuracy curves.

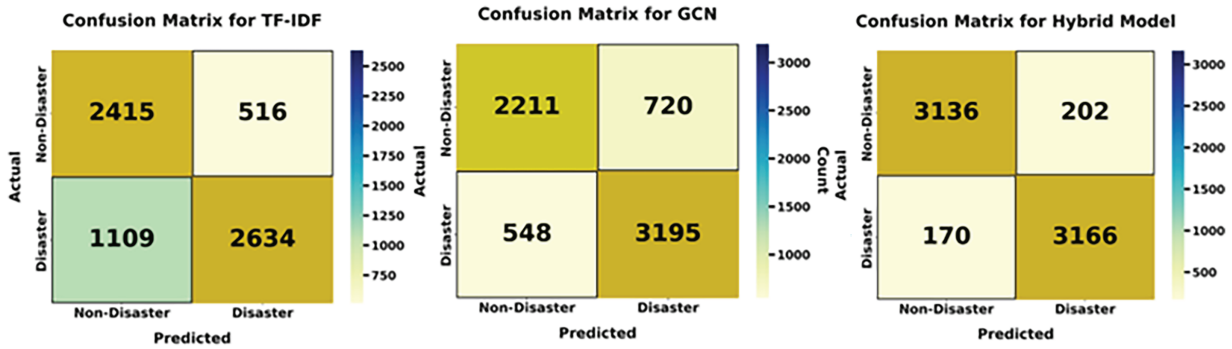

Figure 5: Confusion matrices for different models Task 1.

Based on the obtained results, the TF-IDF + GCN hybrid model emerges as the most effective approach for disaster tweet classification in this study, outperforming baseline TF-IDF, standalone GCN, and LSTM-based models. After establishing baseline performance for binary disaster vs. non-disaster classification (Task 1), the evaluation progressed to a more challenging multi-class scenario. In Task 2, the model was required to distinguish between different disaster types, such as floods and earthquakes, which introduced greater complexity compared to binary classification. This task demanded finer-grained discrimination, as the model needed to capture subtle linguistic differences between disaster categories rather than simply determining the presence or absence of a disaster.

3.2 Classification of Disaster Type Outcomes (Task 2)

The training, validation loss, and accuracy curves over fifty epochs are displayed in Fig. 6. It gives the accuracy vs. loss while the model gets trained. Epochs are displayed on the x-axis, accuracy (%) is shown on the right y-axis, and loss values are shown on the left y-axis. The training and validation loss decreases with increasing epochs, indicating improved learning and generalization of the model. Accuracy increases concurrently, indicating an improvement in classification ability. With a distinct emphasis on striking a

balance between lowering loss and increasing accuracy, the image illustrates how the model's learning has evolved over time.

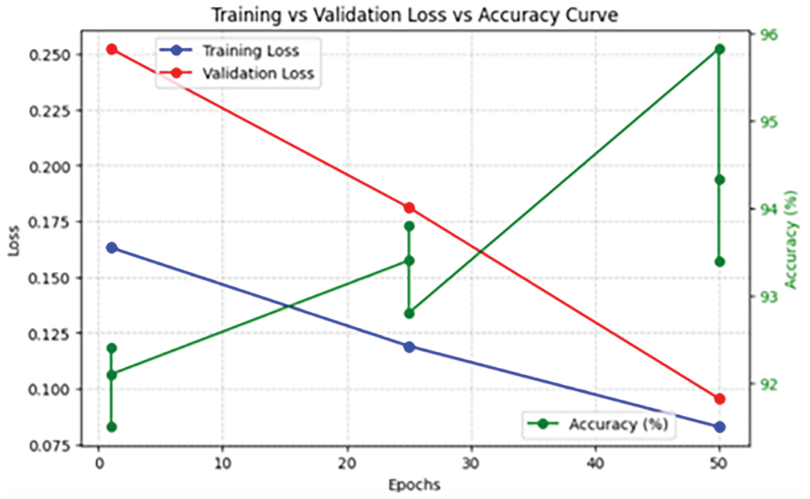

Figure 6: Accuracy curve over epochs and training vs. validation loss.

The confusion matrix heatmaps in Fig. 7 illustrate how well the model detects three disasters. The model generates a large number of accurate predictions, as indicated by the high diagonal values (909, 893, 1031). The framework's accuracy ranges from 93.39% to 95.82%, its precision ranges from 88.80% to 95.92%, and its F1-scores for each class range from 91.00% to 93.37%, according to the metrics table.

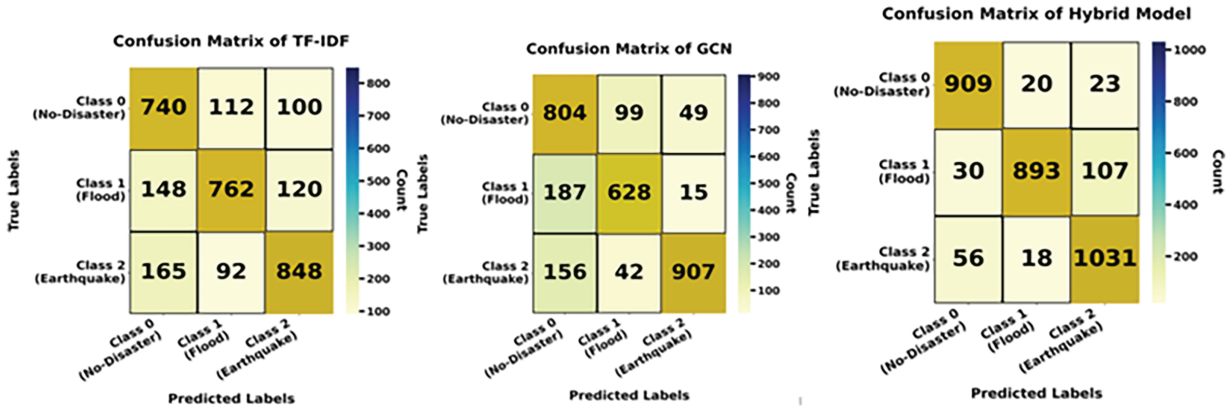

Figure 7: Confusion matrices of different models for Task 2.

Table 8 shows how three different models can be used to predict the type of disaster. The TF-IDF-only model is the worst, but the GCN-only model is a little better because it captures word relationships. Our hybrid model of TF-IDF and GCN gets the best results: 95.49% for No Disaster, 86.70% for Flood, and 93.31% for earthquake, with an overall accuracy of 91.78%. This shows that using both TF-IDF features and GCN embeddings together gives better results for classifying disasters.

Task 3 was about figuring out what kind of disaster it was, and Task 3 makes the analysis even better by sorting tweets into groups based on what kind of help and support they need. This fine-grained classification is harder because tweets often have signals of help that are unclear or overlap.

Table 8: Accuracy of disaster type classification across models.

Model Variant	No Disaster (%)	Flood (%)	Earthquake (%)	Overall Accuracy (%)
TF-IDF Only	77.73	74.08	76.74	76.09
GCN Only	84.50	80.40	82.10	82.30
TF-IDF + GCN	**95.49**	**86.70**	**93.31**	**91.78**

3.3 Findings from Task 3: Help-Seeking Detection

Fig. 8 displays the proposed model's accuracy and loss values with 50 epochs. The test loss drops from 0.1158 to 0.032168, the testing accuracy rises from 91% to 95% in epochs 1 to 50. The figure shows a steady decrease in loss and a steady rise in accuracy, indicating successful model learning and convergence.

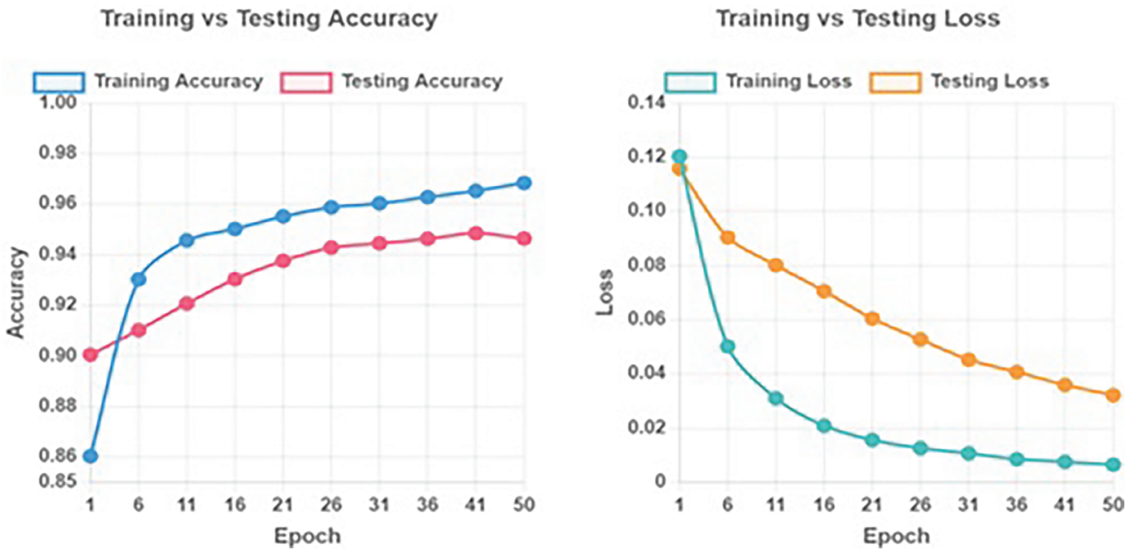

Figure 8: Training and testing accuracy and loss over 50 epochs.

Tables 9 and 10 show how well our suggested hybrid TF-IDF+GCN model works for help categorization tasks. Table 9 shows that the hybrid model consistently gets high precision, recall, and F1-scores across all five help categories. The best results are in Other Useful Information (F1 = 0.9836). The overall accuracy is 94.64%, which shows that it is extremely trustworthy. On the other hand, Table 10 compares several baselines, such as TF-IDF-only, GCN-only, LSTM, Bi-LSTM, and BERT. All of these do much worse across categories, especially when it comes to tweets about infrastructure.

Table 9: Performance of the hybrid model for various help categories.

Category	Precision	Recall	F1-Score
Donation & Volunteering	0.9399	0.9493	0.9468
Food & Medicine	0.9150	0.9459	0.9291
Infrastructure & Utilities	0.8421	0.8533	0.8474
Other Useful Information	0.9676	0.9828	0.9836
Sympathy & Support	0.9356	0.9497	0.9425

(Continued)

Table 9 (continued)

Category	Precision	Recall	F1-Score
Macro Average	**0.9210**	**0.9362**	**0.9278**
Weighted Average	**0.9446**	**0.9599**	**0.9518**
Overall Accuracy		**94.64%**	

Table 10: Performance of various DL models for various help categories.

Help Category	TF-IDF Only (P, R, F1)	GCN Only (P, R, F1)	LSTM (P, R, F1)	Bi-LSTM (P, R, F1)	BERT (P, R, F1)
Donations & Volunteering	(0.45, 0.49, 0.47)	(0.56, 0.40, 0.47)	(0.61, 0.32, 0.42)	(0.55, 0.43, 0.48)	(0.50, 0.40, 0.44)
Food & Medicine	(0.50, 0.53, 0.51)	(0.77, 0.49, 0.66)	(0.77, 0.42, 0.54)	(0.74, 0.49, 0.59)	(0.73, 0.48, 0.58)
Infrastructure & Utilities	(0.34, 0.30, 0.32)	(0.88, 0.26, 0.40)	(0.92, 0.13, 0.22)	(0.91, 0.23, 0.37)	(0.79, 0.13, 0.22)
Other Useful Information	(0.56, 0.57, 0.57)	(0.57, 0.82, 0.68)	(0.55, 0.85, 0.67)	(0.57, 0.79, 0.68)	(0.56, 0.79, 0.65)
Sympathy & Support	(0.65, 0.59, 0.62)	(0.82, 0.58, 0.68)	(0.78, 0.58, 0.67)	(0.76, 0.60, 0.67)	(0.76, 0.56, 0.64)

Fig. 9 demonstrates that the confusion matrix for Task 3 indicates the model's efficacy in categorizing tweets into five assistance-related classifications. Despite the fact that there were only a few small misclassifications between labels that were closely connected to one another, such as Sympathy and Other Useful Information, its great performance is highlighted by the concentration of predictions along the diagonal.

The outcomes from all three tasks indicate that the suggested hybrid model functions effectively. It can be used to find disasters in general and to sort them into different types and help categories. This shows that our model can handle both simple and complicated cases, which is important for real disaster response. To gain a clearer understanding of these results, we will examine the contribution of each model component in Section 3.4. This ablation study looks at the TF-IDF only, the GCN only, and the TF-IDF+GCN model together.

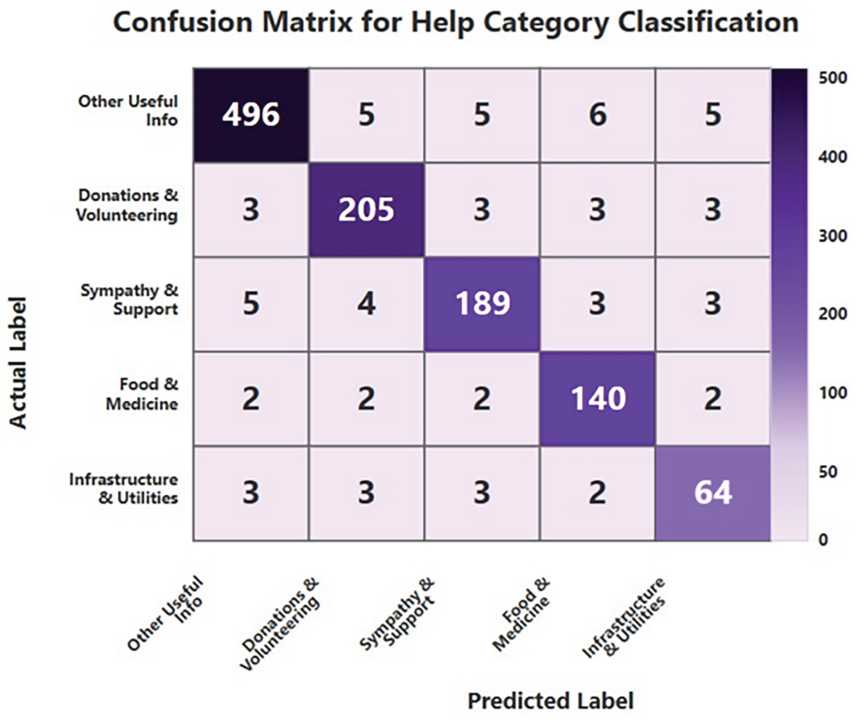

Figure 9: Confusion matrix for help category of Hybrid model.

3.4 Ablation Study and Error Analysis

To illustrate the contributions of TF-IDF, GCN, and their combination, we present the overall accuracy for each of the three tasks in this ablation study. To determine how much each component contributed, we tested three different iterations of the model: TF-IDF-only, GCN-only, and the hybrid TF-IDF+GCN. The results are displayed in Table 11. For Task 1 (Disaster vs. Non-Disaster), the hybrid model achieved 94.47% accuracy, significantly higher than TF-IDF-only (75.60%) and GCN-only (80.98%). This demonstrates that binary detection is improved by combining structural and statistical features. For Task 2 (Disaster Type Classification), the hybrid model received 91.78%, compared to 76.09 percent for the TF-IDF-only model and 82.30% for the GCN-only model. This shows that combining features helps with multi-class classification. Our hybrid model got 94.64% accuracy for Task 3 (help categorization), which is much better than TF-IDF-only (54.01%) and GCN-only (62.89%). The ablation analysis shows that TF-IDF finds out how important words are, while GCN finds out how words are related to each other. Together, they always make things better on all tasks.

Table 11: Ablation study results across all tasks.

Model Variant	Task 1: Disaster vs. Non-Disaster (%)	Task 2: Disaster Type Classification (%)	Task 3: Help Categorization (%)
TF-IDF Only	75.60	76.09	54.01
GCN only	80.98	82.30	62.8
TF-IDF+GCN	94.47	91.78	94.64

Table 12 shows the full error analysis of the suggested hybrid model for all three classification tasks. It shows that the model works well and consistently, even though the tasks are quite challenging. In binary classification (Task 1, 6674 samples), the model achieves 94.47% accuracy with 372 total errors (5.57%), which is acceptable in disaster detection when missing actual catastrophes (FN) should be reduced. False positives (202, 3.03%) outnumber false negatives (170, Effective, bias-free learning is shown by the balanced error distribution. The model achieves 91.78% accuracy in classifying 3-class disasters (Task 2, 3087 samples), with the top issue being Flood*rightarrow*Earthquake misunderstanding (10.39%), accounting for the biggest error in the analysis. Large inter-disaster misunderstanding shows that flood and earthquake tweets share grammatical and contextual characteristics, while confusion with non-disaster content remains modest (43–74 instances). The misunderstanding is asymmetric, with Flood*rightarrow*Earthquake (107) being six times greater than Earthquake*rightarrow*Flood (18), suggesting that flood-related jargon is more commonly misinterpreted as earthquake terms. Despite five semantically overlapping categories and extreme class imbalance (517 to 75 data), the model achieves the maximum accuracy (94.64%) in help category categorization (Task 3, 1156 samples). Most categories had uniform error distributions (0.97%–1.51% every confusion pair), with only the smallest class (Infrastructure, 75 samples) having a 14.67% error rate with 11 absolute mistakes. The model's good discrimination is shown by the continuously low inter-category confusion (2–6 instances per pair) across semantically comparable categories like "Food & Medicine" and "Infrastructure & Utilities". Real semantic ambiguity and linguistic overlap cause errors, not model defects or class imbalance, and the model performs well across all class sizes when accounting for task difficulty.

Table 12: Detailed error analysis of hybrid model.

Task	Error Type	Count	Error Rate (%)	Error Pattern & Analysis
Task 1: Binary Classification (Disaster vs. non-disaster)—Total samples: (6674)				
Binary Classification	False Positives (FP)	202	3.03%	Tweets that weren't about disasters were wrongly labelled as disasters; a small number of false alarms.
	False Negatives (FN)	170	2.55%	Overlooked actual disasters; less than FP, which exhibits a high recall.
Task 2: Disaster Type Classification (Non-disaster, Flood, Earthquake)—Total samples: (3088)				
Three class Classification	Non-Disaster *rightarrow* Flood	20	2.10%	Content that is not a disaster is wrongly la-belled as a flood event.
	Non-Disaster *rightarrow* Earthquake	23	2.42%	Unrelated tweets were mistaken for earth-quake reports.
	Flood *rightarrow* Non-Disaster	30	2.91%	Flood-related tweets are classified as non-disaster content.

(Continued)

Table 12 (continued)

Task	Error Type	Count	Error Rate (%)	Error Pattern & Analysis
	Flood *rightarrow* Earthquake	107	10.39%	Maximum confusion: Flood events erroneously categorized as earthquakes; consider-able inter-disaster ambiguity.
	Earthquake *rightarrow* Non-Disaster	56	5.07%	Tweets about earthquakes are misinterpreted as regular updates, causing some misunderstanding.
	Earthquake *rightarrow* Flood	18	1.63%	Floods and earthquakes mixed up; less uncertainty in the other direction.
Task 3: Help Category Classification (5 Categories)—Total samples: (1156)				
Five class Classification	Other Useful Information Errors.	21	4.06%	Donations-5, Sympathy-5, Food-6, Infrastructure-5; well-distributed errors.
	Donations & Volunteering Errors.	12	5.53%	Other Information-3, Sympathy-3, Food-3, Infrastructure-3; uniform distribution.
	Sympathy & Support Errors.	10	5.03%	Other Information-3, Donations-3, Food-2, Infrastructure-2; Low error count.
	Food & Medicine Errors.	8	5.41%	Other Information-2, Donations-2, Sympathy-2, Infrastructure-2; Balanced confusion.
	Infrastructure & Utilities Error.	11	14.67%	Other Information-3, Donations-3, Sympathy-3, Food-2; Highest percentage due to smallest class (75 samples).

4 Conclusion

This study presents a comprehensive multi-stage hybrid framework that integrates TF-IDF with Graph Convolutional Networks (GCNs) to address three core challenges in disaster-related tweet classification. In Task 1, the proposed model successfully distinguishes disaster-related tweets from non-disaster content, achieving an accuracy of 94.47%. For Task 2, the framework categorizes tweets into three classes: No Disaster, Flood, and Earthquake, with class-wise accuracies of 95.49%, 86.70%, and 93.31%, respectively, resulting in an overall accuracy of 91.78%. In Task 3, the model identifies various categories of aid and support, including donations, food, medical assistance, and other forms of help, achieving an accuracy of 94.64%.

Ablation experiments indicate that while TF-IDF and GCN independently contribute valuable features, their integration within a unified hybrid architecture consistently produces superior performance across all tasks.

Future work will focus on extending the proposed framework to multilingual and code-mixed datasets to enhance its linguistic diversity and geographic applicability. In addition, comparative evaluations involving large-scale AI models, including OpenAI GPT-based classifiers, will be conducted to examine the scalability and adaptability of the approach. Incorporating real-time data streaming, geolocation-aware analysis, and Explainable AI (XAI) techniques with dynamic graph construction is expected to further improve model transparency, interpretability, and operational usefulness in emergency response and humanitarian decision-making.

Acknowledgement: Not applicable.

Funding Statement: Not applicable.

Author Contributions: Conceptual design, workflow creation, Basudev Nath and Sudhansu Shekhar Patra; modeling, implementation and supervision, Subrata Chowdhury, Hassan Alkhiri and Sheraz Aslam; software, Basudev Nath and Kainat Mustafa; data collection and preprocessing, Deepak Sahoo and Kainat Mustafa; interpretation of result analysis, Deepak Sahoo, Basudev Nath and Sudhansu Shekhar Patra; manuscript draft preparation, review, and editing, Basudev Nath, Sheraz Aslam, Hassan Alkhiri and Sudhansu Shekhar Patra; project administration, Sheraz Aslam and Hassan Alkhiri. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are given in the reference.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Vieweg S, Hughes AL, Starbird K, Palen L. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10; 2010 Apr 10–15; Atlanta, GA, USA. New York, NY, USA: Association for Computing Machinery; 2010. p. 1079–88. doi:10.1145/1753326.1753486.
2. Gao H, Barbier G, Goolsby R. Harnessing the crowdsourcing power of social media for disaster relief. IEEE Intell Syst. 2011;26(3):10–4. doi:10.1109/mis.2011.52.
3. Khare P, Burel G, Alani H. Classifying crises-information relevancy with semantics. In: Gangemi A, Navigli R, Vidal ME, Hitzler P, Troncy R, Hollink L et al., editors. The semantic web. Cham, Switzerland: Springer International Publishing; 2018. p. 367–83. doi:10.1007/978-3-319-93417-4_24.
4. Yin M, Wan M, Lin Z, Jiang J. Moralization-aware identity fusion for detecting violent radicalization in social media. Inform Process Manage. 2026;63(2):104413. doi:10.1016/j.ipm.2025.104413.
5. Meng T, Shou Y, Ai W, Du J, Liu H, Li K. A multi-message passing framework based on heterogeneous graphs in conversational emotion recognition. Neurocomputing. 2024;569:127109. doi:10.1016/j.neucom.2023.127109.
6. Blei D, Ng A, Jordan M. Latent dirichlet allocation. J Mach Learn Res. 2003;3:993–1022. doi:10.7551/mitpress/1120.003.0082.
7. Kowsari K, Jafari Meimandi K, Heidarysafa M, Mendu S, Barnes L, Brown D. Text classification algorithms: a survey. Information. 2019;10(4):150. doi:10.3390/info10040150.
8. Stowe K, Paul MJ, Palmer M, Palen L, Anderson K. Identifying and categorizing disaster-related tweets. In: Proceedings of the Fourth International Workshop on Natural Language Processing for Social Media; 2016 Nov 1; Austin, TX, USA. p. 1–6.

9. Verma S, Vieweg S, Corvey W, Palen L, Martin J, Palmer M, et al. Natural language processing to the rescue? extracting "situational awareness" tweets during mass emergency. Proc Int AAAI Conf Web Soc Media. 2011;5(1):385–92. doi:10.1609/icwsm.v5i1.14119.
10. Imran M, Elbassuoni S, Castillo C, Diaz F, Meier P. Practical extraction of disaster-relevant information from social media. In: Proceedings of the 22nd International Conference on World Wide Web; 2013 May 13–17; Rio de Janeiro, Brazil. p. 1021–4.
11. Chen Y. Convolutional neural network for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014 Oct 25–29; Doha, Qatar. p. 1746–51.
12. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80. doi:10.1162/neco.1997.9.8.1735.
13. Bhoi A, Pujari SP, Balabantaray RC. A deep learning-based social media text analysis framework for disaster resource management. Soc Netw Anal Min. 2020;10:1–14. doi:10.1007/s13278-020-00692-1.
14. Caragea C, Silvescu A, Tapia AH. Identifying informative messages in disaster events using convolutional neural networks. In: The 13th International Conference on Information Systems for Crisis Response and Management; 2016 May 22–25; Rio de Janeiro, Brazil. p. 137–47.
15. Devlin J, Chang MW, Lee K, Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers); 2019 Jun 2–7; Minneapolis, MN, USA. p. 4171–86.
16. Liu J, Singhal T, Blessing LT, Wood KL, Lim KH. Crisisbert: a robust transformer for crisis classification and contextual crisis embedding. In: Proceedings of the 32nd ACM Conference on Hypertext and Social Media; 2021 Aug 30–Sep 2; Virtual. p. 133–41.
17. Bastings J, Titov I, Aziz W, Marcheggiani D, Sima'an K. Graph convolutional encoders for syntax-aware neural machine translation. arXiv:1704.04675. 2017.
18. Zhang Y, Liu Q, Song L. Sentence-state LSTM for text representation. arXiv:1805.02474. 2018.
19. Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: Proceedings of the 30th International Conference on Neural Information Processing Systems; 2016 Dec 5–10; Barcelona, Spain. p. 3844–52.
20. Zheng W, Lu S, Cai Z, Wang R, Wang L, Yin L. PAL-BERT: an improved question answering model. Comput Model Eng Sci. 2024;139(3):2729–45. doi:10.32604/cmes.2023.046692.
21. Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst. 2020;32(1):4–24. doi:10.1109/tnnls.2020.2978386.
22. Zhang Z, Cui P, Zhu W. Deep learning on graphs: a survey. IEEE Trans Knowl Data Eng. 2020;34(1):249–70. doi:10.1109/tkde.2020.2981333.
23. Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, et al. Graph neural networks: a review of methods and applications. AI Open. 2020;1:57–81. doi:10.1016/j.aiopen.2021.01.001.
24. Yao L, Mao C, Luo Y. Graph convolutional networks for text classification. Proc AAAI Conf Artif Intell. 2019;33:7370–7. doi:10.1609/aaai.v33i01.33017370.
25. Meng Q, Song Y, Mu J, Lv Y, Yang J, Xu L, et al. Electric power audit text classification with multi-grained pre-trained language model. IEEE Access. 2023;11:13510–8. doi:10.1109/access.2023.3240162.
26. Wang Y, Wang S, Yao Q, Dou D. Hierarchical heterogeneous graph representation learning for short text classification. arXiv:2111.00180. 2021.
27. Zhang Y, Yu X, Cui Z, Wu S, Wen Z, Wang L. Every document owns its structure: inductive text classification via graph neural networks. arXiv:2004.13826. 2020.
28. Wang K, Han SC, Poon J. InducT-GCN: inductive graph convolutional networks for text classification. In: 2022 26th International Conference on Pattern Recognition (ICPR); 2022 Aug 21–25; Montreal, QC, Canada. p. 1243–9.
29. Ye Z, Jiang G, Liu Y, Li Z, Yuan J. Document and word representations generated by graph convolutional network and bert for short text classification. In: Frontiers in artificial intelligence and applications. Amsterdam, The Netherlands: IOS Press; 2020. p. 2275–81. doi:10.3233/faia200355.

30. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. arXiv:1710.10903. 2017.
31. Linmei H, Yang T, Shi C, Ji H, Li X. 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); 2019 Nov 3–7; Hong Kong, China. p. 4821–30.
32. Liu Y, Guan R, Giunchiglia F, Liang Y, Feng X. Deep attention diffusion graph neural networks for text classification. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing; 2021 Nov 7–11; Virtual. p. 8142–52.
33. Wang H, Li F. A text classification method based on LSTM and graph attention network. Connect Sci. 2022;34(1):2466–80. doi:10.1080/09540091.2022.2128047.
34. Paul NR, Sahoo D, Balabantaray RC. VocabGCN-BERT: a hybrid model to classify disaster related tweets. In: 2022 OITS International Conference on Information Technology (OCIT); 2022 Dec 14–16; Bhubaneswar, India. p. 55–60.
35. DATASET 2. [cited 2026 Jan 11]. Available from: https://crisisnlp.qcri.org/crisis_datasets_benchmarks.html.
36. DATASET 1. [cited 2026 Jan 11]. Available from: https://crisislex.org/.