BAREKENG: Journal of Mathematics and Its Applications
September 2026 Volume 20 Issue 3 Page 2613-2630
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol20iss3pp2613-2630

NATURAL DISASTER REPORT ON SOCIAL MEDIA CLASSIFICATION METHOD BASED ON WORD EMBEDDING AND GRAPH ATTENTION NETWORK

Mohammad Reza Faisal[1*], **Irwan Budiman**[2], **Dodon Turianto Nugrahadi**[3], **Muhammad Rafi**[4], **Mera Kartika Delimayanti**[5], **Luu Duc Ngo**[6], **Moses Okechukwu Onyesolu**[7]

[1,2,3,4]Department of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Lambung Mangkurat
Jln. Jend. Achmad Yani Km. 35,5, Simpang Empat Banjarbaru, 70714, Indonesia

[5]Department of Computer and Informatics Engineering, Politeknik Negeri Jakarta
Jln. Prof. DR. G.A. Siwabessy Depok, 16425, Indonesia

[6]Faculty of Information Technology, Bac Lieu University
Bac Lieu City, 260000, Vietnam

[7]Department of Computer Science, Faculty of Physical Sciences, Nnamdi Azikiwe University
Anambra State, 435115, Nigeria

*Corresponding author's e-mail: * **reza.faisal@ulm.ac.id***

Article Info

Article History:

Received: 7th October 2025
Revised: 2nd December 2025
Accepted: 17th March 2026
Published:8th April 2026

Keywords:

GAT;
GGN;
Natural disaster;
Social media;
Text Classification;
Word Embedding.

ABSTRACT

Natural disasters frequently occur unexpectedly and seriously threaten human safety and infrastructure. Traditional detection systems rely heavily on IoT sensors and satellite monitoring, which are often costly and less accessible in resource-limited or remote areas. In contrast, social media provides a rich and real-time source of information, as users frequently post eyewitness reports during disaster events. However, automatically classifying these posts into relevant disaster categories remains challenging due to the short and informal nature of the text. The research aims to develop a high-performing classification model for disaster-related tweets using graph-based neural architectures and structured word embedding representations. The method used is a comparative implementation of Graph Convolutional Network (GCN) and Graph Attention Network (GAT) models, with input constructed by concatenating vectors from three word embedding techniques—Word2Vec, FastText, and GloVe—across seven multilingual datasets, consisting of four English datasets covering wildfires, earthquakes, hurricanes, and floods, and three Indonesian datasets covering earthquakes, floods, and forest fires. The result of this study is that GAT outperformed GCN in all scenarios, with FastText embeddings yielding the highest individual performance. In contrast, combined embeddings sometimes led to performance degradation due to redundancy. The average F1-score for GCN is 0.749, while GAT achieves 0.915. The main contribution of this study lies in demonstrating that attention-based graph models combined with structured embeddings can significantly enhance multilingual disaster message classification, offering a more robust foundation for AI-based disaster informatics and early-warning systems. Nevertheless, model training exhibited several limitations, including early-epoch accuracy fluctuations, signs of mild overfitting, and instability in validation accuracy, indicating that further optimization is needed to achieve more stable convergence.

How to cite this article:

M. R. Faisal, I. Budiman, D. T. Nugrahadi, M. Rafi, M. K. Delimayanti, L. D. Ngo and M. O. Onyesolub., "NATURAL DISASTER REPORT ON SOCIAL MEDIA CLASSIFICATION METHOD BASED ON WORD EMBEDDING AND GRAPH ATTENTION NETWORK", *BAREKENG: J. Math. & App.*, vol. 20, no. 3, pp. 2613-2630, Sep, 2026.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

Natural disasters are events that often occur unexpectedly and have significant impacts on human safety and infrastructure. Traditionally, disaster detection systems have relied heavily on Internet of Things (IoT)-based sensors and satellite monitoring. Although accurate, these approaches require substantial investment in installing sensors in vulnerable areas and the high operational costs associated with satellite use. This poses a challenge, particularly in remote or resource-limited regions. Social media has become a widely used communication platform globally in the current digital era. Many individuals spontaneously share information about events they witness, including natural disasters. Such posts often contain real-time cues regarding time, location, and environmental conditions [1]. Therefore, text data from social media holds considerable potential as an alternative data source for disaster detection systems.

In general, social media messages related to natural disasters can be categorized into three types: eyewitness accounts, third-party sources such as quoted news or reposted content, and messages unrelated to natural disasters that still contain similar keywords. Manually filtering and verifying these messages is highly challenging due to the large volume of data and time constraints [2]. Therefore, artificial intelligence-based approaches, particularly automated text classification, offer a promising solution to efficiently and accurately filter information. This can support more cost-effective disaster response efforts than relying solely on sensor and satellite-based systems.

In general, text classification research aims to improve accuracy by combining methods at the feature extraction and classification stages. Commonly employed feature extraction methods include frequency-based representations, such as Term Frequency-Inverse Document Frequency (TF-IDF), and word embedding-based representations, such as word2vec, GloVe, and fastText, which are typically integrated with traditional machine learning or deep learning algorithms. A previous study on identifying eyewitness reports of natural disasters in English demonstrated that the combination of TF-IDF and Random Forest yielded good performance, with F1-scores of 0.74, 0.82, 0.83, and 0.81 for flood, hurricane, earthquake, and wildfire events, respectively [1]. Another model was developed by applying a word embedding-based feature extraction technique, specifically word2vec, where structured data were generated by summing word vectors within a sentence, resulting in a compact feature space of 100 dimensions with comparable performance to the previous study. This approach proved effective, as the resulting structured data retained semantic content by aggregating all word vectors in a message, unlike TF-IDF features, which capture only word frequency [2]. Another study [3] generated structured data by concatenating word vectors to preserve semantic meaning and word order, using three word embedding techniques: word2vec, fastText, and GloVe. Classification models were then developed using combinations of these structured representations with deep learning algorithms, namely Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM), and the Bidirectional Encoder Representations from Transformers (BERT). The average F1-scores achieved by these models-1D CNN, 2D CNN, 3D CNN, LSTM, and BERT—were 0.63, 0.63, 0.66, 0.60, and 0.57, respectively. These findings indicate that structured data formed through word embedding techniques enables better performance than BERT-based models. Furthermore, CNN outperformed LSTM and BERT in this context, likely due to its advantages in processing short texts such as social media messages. At the same time, LSTM and BERT are more effective for longer textual inputs.

Other studies have employed social media messages in Indonesian, focusing on disasters common in Indonesia—namely floods, earthquakes, and wildfires. Flood-related disaster messages were classified using TF-IDF feature extraction and Support Vector Machine (SVM) classification, yielding accuracies of 77.87% and 77.5% [4], [5]. Another study utilized reports of the three disaster types and structured one-dimensional data through concatenated word vectors based on three embedding techniques, combined with 1D CNN classification [6]. One challenge with this approach was the varying number of words per message, which was addressed by truncating words to equalize input dimensions. The classification performance for earthquake, flood, and wildfire messages was 74.33%, 74%, and 81.14%, respectively. This study also showed that combining vectors from the three-word embeddings produced better results than using individual embeddings alone. Another study [7] generated 2D and 3D structured data to be processed using 2D and 3D CNNs, demonstrating accuracy improvements: 78.33%, 78.33%, and 81.97% for earthquake, flood, and wildfire identification, respectively. In a subsequent study [8], padding techniques were employed instead of truncation to ensure all word vectors were retained, and structured data were created in 1D, 2D, and 3D formats. To further improve model performance, a hybrid CNN-LSTM model was implemented, leveraging CNN's strength in capturing local features (word patterns) and LSTM's ability to model word-sequence context. The resulting classification accuracies for earthquake, flood, and wildfire messages were 83.38%, 83.72%, and 89.03%, respectively.

Recently, graph-based algorithms such as Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) have emerged and gained adoption in natural language processing (NLP) tasks, particularly in text classification [9], [10]. A study using social media text for cyberbullying detection applied TF-IDF feature extraction with GCN and GAT classifiers, achieving accuracies of 94.87% and 87.79%, respectively [11]. Several other studies also employed TF-IDF features as input for GCN and GAT models [12], [13]. One study [14] explored word-embedding-based features by constructing graphs from summed word2vec vectors per sentence and achieved an accuracy of 85.58%. Another study used GloVe embeddings and GCN to classify long texts in datasets such as Movie Review, Reuters-8, and Stanford Sentiment Treebank, obtaining accuracies of 82.3%, 96.8%, and 49.8%, respectively [15]. To date, no studies have utilized fastText with GCN, nor have any explored the integration of word embedding-based feature extraction with GAT.

Given these research gaps, there remains ample opportunity to develop novel methods to achieve higher-performing classification models for natural disaster messages. This study utilizes four English-language datasets of disaster reports [1] and three Indonesian-language datasets [6]. Feature extraction is based on three word embedding techniques—word2vec, fastText, and GloVe—and their combination, to enrich the resulting structured data. To address the identified gaps, this study proposes a GNN-based classification model trained on structured data derived from a combination of three-word embeddings. In contrast to prior studies that predominantly use TF-IDF or a single embedding technique, this research systematically evaluates all combinations of embeddings across two GNN architectures for natural disaster message classification. Furthermore, this study is the first to implement GAT with word-embedding inputs and to evaluate the model across seven multilingual datasets. The novelty of this work lies in integrating multi-embedding representations into graph structures and the expanded application of GNNs within the domain of social text-based disaster detection.

2. RESEARCH METHODS

2.1 Dataset

The datasets used in this study are presented in Table 1. Each dataset consists of three classes: direct eyewitness, non-eyewitness, and don't know. Messages in the eyewitness category contain natural disaster-related keywords posted by users who directly witnessed the events. Messages in the non-eyewitness category also report natural disasters, but are not written by actual eyewitnesses. Meanwhile, messages classified under the don't know category include natural disaster-related keywords, but the content is not actually about a disaster event [8].

Table 1. Dataset

Language	Natural Disaster	Class Label			Notation	Ref
		direct eyewitness	none eyewitness	don't know		
English	Wildfires	189	432	1379	$t_{en,wildfires}$	[1]
	Earthquakes	1600	200	200	$t_{en,earthquakes}$	[1]
	Hurricanes	465	336	1199	$t_{en,hurricanes}$	[1]
	Floods	627	822	551	$t_{en,floods}$	[1]
Indonesia	Earthquakes	1000	1000	1000	$t_{id,earthquakes}$	[6]
	Floods	1000	1000	1000	$t_{id,floods}$	[5], [6]
	Forest Fires	1000	1000	1000	$t_{id,forest\ fires}$	[6]

Each class contains an equal number of messages for the Indonesian-language datasets, resulting in a balanced classification dataset. In contrast, the English-language datasets are imbalanced due to differences in the number of messages per class. The English-language dataset used in this study can be downloaded from https://crisisnlp.qcri.org, specifically under Resource #12. The original dataset contains approximately 14,000 raw messages collected during natural disaster events, including hurricanes, earthquakes, floods, and wildfires (forest fires). After annotation, two independently labeled versions were produced: one annotated by the original authors and another by crowdflower-paid workers. In this research, we employ the version annotated by crowdflower paid workers, consisting of 2,000 labeled messages for each disaster type, with the distribution across class labels presented in Table 1.

2.2 Methods

Fig. 1 illustrates the stages of this study: (1) preprocessing, (2) feature extraction, (3) classification, and (4) model performance.

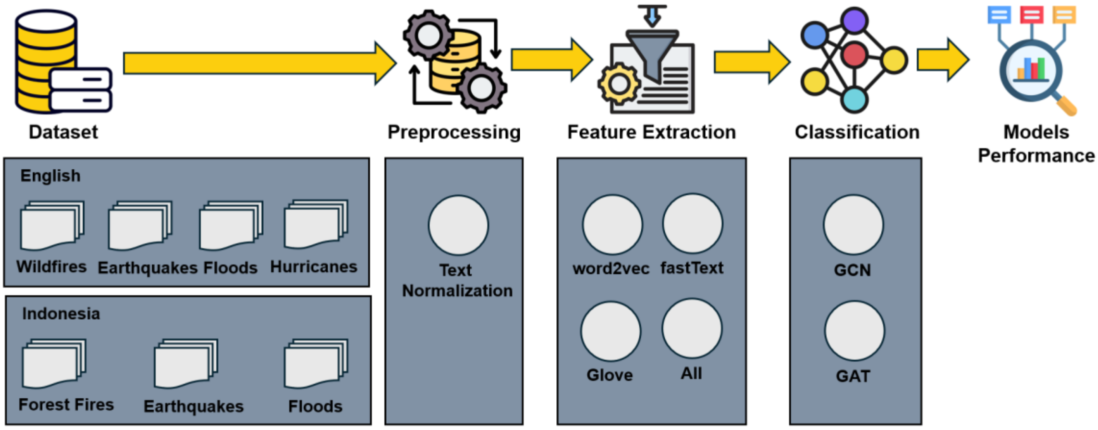


Figure 1. Research flow.

2.2.1 Preprocessing

Fig. 1 shows that the first stage after collecting the datasets is text preprocessing, which involves text normalization to simplify the text, ensuring consistency and facilitating further processing [16], [17]. The text normalization functions applied in this study are as follows:

$caseFolding(t_i)$: converts all characters to lowercase,

$remUser(t_i)$: removes usernames (e.g., @username),

$remURL(t_i)$: removes hyperlinks (URLs),

$remDigit(t_i)$: removes numerical digits,

$remPunct(t_i)$: removes punctuation marks,

$remNA(t_i)$: removes non-alphabetic characters, such as symbols or emojis.

Each function can be executed in a pipeline, resulting in the normalized text function $f_n(t_i)$ as shown in Eq. (1), where t_i is the original dataset as listed in Table 1, and t_i^n is the normalized result.

$$f_n(t_i) = remNA\left(remPunct\left(remDigit\left(remURL\left(remUser\left(caseFolding(t_i)\right)\right)\right)\right)\right) = t_i^n. \quad (1)$$

2.2.2 Feature Extraction

The second stage, which is shown in Fig. 1 is feature extraction, which aims to transform unstructured text data into structured numerical representations that can be processed by classification algorithms [2], [3], [18]. A word embedding model must be constructed before feature extraction using word embedding. In this study, three embedding models were developed: word2vec, fastText, and GloVe. The data used to train these embedding models are represented by Eqs. (2) and (3):

$$D_{en} = \{D_{en,wiki}, t^n_{en,wildfires}, t^n_{en,earthquakes}, t^n_{en,hurricanes}, t^n_{en,floods}\}, \quad (2)$$

$$D_{id} = \{D_{id,wiki}, t^n_{id,earthquakes}, \quad t^n_{id,floods}, t^n_{id,forest\ fires}\}, \quad (3)$$

where $D_{en,wiki}$ denotes the English Wikipedia corpus and $D_{id,wiki}$ denotes the Indonesian Wikipedia corpus, both of which contain extensive textual data. D_{en} and D_{id} represent the merged corpora of Wikipedia texts and the normalized datasets. Training with large corpora produces better word embedding models. The word embedding training process is defined by Eq. (4):

$$E_{model} = f_{model}(D), E \in \mathbb{R}^{|V|\times d} and\ model \in \{word2vec, fastText, Glove\}, \quad (4)$$

where E is the embedding matrix, $|V|$ is the vocabulary size (number of unique words in the corpus), and d is the embedding dimension, which represents the length of each word vector. In this study, we use $d = 100$. Ketiga matriks embedding yang dihasilkan pada penelitian ini adalah $E_{word2vec}$, $E_{fastText}$, and E_{Glove}.

The next step involves transforming each message t_i^n into a structured representation using the function $f_{we}(S)$, defined in Eq. (5):

$$f_{we}(S) = \|_{i=1}^{n}\ v(w_i), w_i \in S, \quad (5)$$

where S is a sentence in the message, w_i is the word in S, $v(\cdot)$ is a function that retrieves the word vector w_i from the embedding matrix E_{model}, with $we \in \{word2vec, fastText, Glove\}$. The resulting word vectors w_1 to w_n are concatenated.

Since $f_{we}(S)$ produces vectors of varying lengths depending on the sentence length n, this study standardizes the output length to match the longest sentence S_{max}. The padding function is described in Eq. (6):

$$pad(X, L) = \begin{cases} [v_1, \dots, v_T, 0, \dots, 0], if\ T < L \\ [v_1, \dots, v_L], if\ T = L \end{cases}, \quad (6)$$

where $X = f_{we}(S)$, T represents the number of features in X, and L is the number of features in $f_{we}(S_{max})$. As shown in Eq. (6), if $T < L$, then the values from v_{T+1} to v_L are padded with zeros. Thus, the final function used to perform feature extraction on a message is defined in Eq. (7):

$$f'_{we}(S) = pad(f_{we}(S), L) \quad (7)$$

To combine the feature extraction results of all messages within a dataset, Eq. (8) is employed.

$$f_{fe,we}(t^n_{lang,dataset}) = \bigcup_{j=1}^{m} f'_{we}(S_{lang,dataset,j}) = t'_{lang,dataset,we}, \quad (8)$$

where:

$S_j \in t^n_{lang,dataset}$,

$t^n_{lang,dataset}$ represents the dataset that has been processed using the text normalization function $N(\cdot)$,

$lang \in \{en, id\}$,

$dataset \in \{wildfires, forest\ fires, earthquakes, floods, hurricanes\}$,

$we \in \{word2vec, fastText, Glove\}$.

This study also investigates the integration of feature extraction results from the three word embedding techniques. The process of combining the three structured data representations is illustrated in Eq. (9):

$$t'_{lang,dataset,all} = \{t'_{lang,dataset,word2vec}, t'_{lang,dataset,fastText}, t'_{lang,dataset,Glove}\}. \quad (9)$$

2.2.3 Classification

The third stage, as shown in Fig. 1, involves classification. The classification algorithm used in this study is based on a Graph Neural Network (GNN). Graph Neural Networks are deep learning models that process data structured as graphs. A graph is a data structure composed of nodes and edges that connect those nodes [19]. One of the fundamental architectures within GNNs is the Graph Convolutional Network (GCN), which extends traditional convolution operations to graph structures. An advanced variant of GCN is the Graph Attention Network (GAT). GCN and GAT generally follow similar core computational steps, as outlined in Table 2 [10], [20]. This stage uses 80% of the feature-extracted dataset for each natural disaster category.

Table 2. Comparison Between GCN and GAT

No	Step	GCN	GAT
1	Input Data	Node feature matrix and adjacency matrix A.	Same as GCN: node features and adjacency matrix (or edge list).
2	Graph Normalization	Adjacency matrix A is added with self-loops and normalized using node degrees: $\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$.	Does not explicitly normalize the adjacency matrix.
3	Feature Aggregation	Aggregates neighbor features using a linear operation based on the adjacency matrix: $H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)})$.	Aggregates using an attention-weighted sum between nodes: $h_i' = \sigma(\sum_{j\in N(i)} \alpha_{ij} W h_j)$.
4	Inter-Node Weighting	Connection weights between nodes are determined by the graph structure (static).	Connection weights are computed dynamically using the attention mechanism: $\alpha_{ij} = softmax\left(LeakyReLU(\alpha^T[Wh_i \vert\vert Wh_j])\right)$.
5	Training & Optimization	Trained like a standard neural network using cross-entropy or other loss functions.	Same, but includes additional parameters from the attention vector.
6	Output Layer	Node representations from graph convolution, usually followed by a softmax.	Node representations after the attention mechanism are also typically passed through a softmax.

2.2.4 Model's Performance

The last stage which is shown in in Fig. 1, is the classification model's performance is evaluated using accuracy and F1 score, each calculated based on the values from the confusion matrix [21]. This stage uses 80% of the feature-extracted dataset for each natural disaster category. A confusion matrix is a tabular representation that compares the model's predicted labels with the actual ground truth labels. This study addresses a multiclass classification problem, as the datasets used consist of three classes. For instance, if the dataset contains classes A, B, and C, the confusion matrix structure is shown in Table 3.

Table 3. Confusion matrix multiclass

	Predicted A	Predicted B	Predicted C
Actual A	a_{11}	a_{12}	a_{13}
Actual B	a_{21}	a_{22}	a_{23}
Actual C	a_{31}	a_{32}	a_{33}

The prediction results and actual labels from the confusion matrix are used to calculate accuracy, which is the percentage of correct predictions over the total number of predictions made [22]. The formula for computing accuracy is presented in Eq. (10).

$$accuracy = \frac{a_{11} + a_{22} + a_{33}}{\sum_{i=1}^{3}\sum_{j=1}^{3} a_{ij}} \times 100\%. \tag{10}$$

In addition to accuracy, this study also utilizes the F1 score as a performance metric. The F1 score is particularly suitable for imbalanced datasets, as it is more sensitive to errors in the minority class. This metric captures the balance between precision and recall in a single measure. Precision is the ratio of true positives to the total number of positive predictions made by the model. At the same time, recall measures the ratio of true positive predictions to the total actual positive instances [22].

To compute the F1 score, the process begins by calculating the precision for each class i, where $i \in \{A, B, C\}$. The formula for computing precision is given in Eq. (11).

$$precision_i = \frac{TP_i}{TP_i + FP_i} = \frac{a_{ii}}{\sum_{k=1}^{3} a_{ki}}. \tag{11}$$

The second step is to calculate the recall, using the formula presented in Eq. (12).

$$recall_i = \frac{TP_i}{TP_i + FN_i} = \frac{a_{ii}}{\sum_{k=1}^{3} a_{ik}}. \tag{12}$$

Next, the F1 score for each class is computed using the formula provided in Eq. (13).

$$F1_i = 2 \cdot \frac{precision_i \cdot recall_i}{precision_i + recall_i}. \tag{13}$$

The final step involves calculating the macro-average F1 score, which is determined using the formula shown in Eq. (14).

$$F1_i = \frac{F1_A + F1_B + F1_C}{3}. \tag{14}$$

2.3 Baselines

In the experiment, we compared our methods with various state-of-the-art models. The following models have been previously applied to classify English-language disaster-related messages:

1. TF-IDF Random Forest [1]: this model employs the Term Frequency-Inverse Document Frequency (TF-IDF) technique for feature extraction and utilizes the Random Forest classification algorithm.
2. Word Embedding Random Forest [2]: This model uses word embedding-based feature extraction, specifically word2vec, by summing all word vectors in a sentence. The classification algorithm used is Random Forest.
3. Word Embedding LSTM [3]: This model adopts word embedding techniques, including word2vec, fastText, and GloVe, and processes them using the Long Short-Term Memory (LSTM) algorithm.
4. BERT [3]: This model leverages BERT for feature extraction and classification.
5. Previous studies also developed models for classifying disaster-related messages in the Indonesian language:
6. TF-IDF SVM [5]: This model utilizes the TF-IDF feature extraction method combined with the Support Vector Machine (SVM) classification algorithm. It has been applied to Indonesian disaster-related datasets.
7. Word Embedding Bi-LSTM [23]: This model uses word2vec-based word embedding features, processed with a Bidirectional LSTM (Bi-LSTM) algorithm.
8. Word Embedding CNN [7]: This model uses word2vec, fastText, and GloVe embeddings, structured as one-, two-, and three-dimensional inputs, which are then processed by 1D, 2D, and 3D CNNs, respectively.
9. Word Embedding Hybrid CNN LSTM [8]: This model combines CNN and LSTM architectures. Word embeddings (word2vec, fastText, and GloVe) are structured into 1D, 2D, and 3D input formats and processed by 1D, 2D, and 3D CNN-LSTM architectures, respectively.

The following models have not yet been used for disaster-related message classification in the two language groups above, but have demonstrated strong performance in other text classification tasks. This study evaluates these models for both English and Indonesian disaster-related datasets using the feature extraction and classification techniques described.

1. TF-IDF GCN [24]: This model uses TF-IDF for feature extraction and the Graph Convolutional Network (GCN) for classification.
2. TF-IDF GAT [25]: This model utilizes TF-IDF for feature extraction and Graph Attention Network (GAT) for classification.
3. mBERT [26]: The pre-trained Multilingual BERT (mBERT), developed by Google, is trained on 104 languages simultaneously without language-specific adjustments.
4. XLM [27]: The Cross-lingual Language Model (XLM) is a pre-trained model capable of understanding and processing multiple languages within a single framework.
5. XLM-RoBERTa [28]: This model combines XLM's multilingual training capabilities with RoBERTa's optimizations and training techniques.

3. RESULTS AND DISCUSSION

3.1 Result

This subsection presents the results from each stage of the research flow as illustrated in Figure 1. The results of the preprocessing stage using the function $f_n(t_i)$ are shown in Table 4.

Table 4. Result of Text Normalization

Language	Dataset	Input	Result
English	Wildfires	$t_{en,wildfires}$	$t^n_{en,wildfires}$
	Earthquakes	$t_{en,earthquakes}$	$t^n_{en,earthquakes}$
	Hurricanes	$t_{en,hurricanes}$	$t^n_{en,hurricanes}$
	Floods	$t_{en,flood}$	$t^n_{en,flood}$
Indonesia	Earthquakes	$t_{id,earthquakes}$	$t^n_{id,earthquakes}$
	Floods	$t_{id,floods}$	$t^n_{id,floods}$
	Forest Fires	$t_{id,forest\ fires}$	$t^n_{id,forest\ fires}$

The feature extraction stage produced 21 structured datasets, as summarized in Table 5.

Table 5. Result of Feature Extraction

No	Input	Feature Extraction	Result
1	$t^n_{en,wildfires}$	$f_{fe,word2vec}(input)$	$t'_{en,wildfires,word2vec}$
2		$f_{fe,fastText}(input)$	$t'_{en,wildfires,fastText}$
3		$f_{fe,Glove}(input)$	$t'_{en,wildfires,Glove}$
4		All	$t'_{en,wildfires,all}$
5	$t^n_{en,earthquakes}$	$f_{fe,word2vec}(input)$	$t'_{en,earthquakes,word2vec}$
6		$f_{fe,fastText}(input)$	$t'_{en,earthquakes,fastText}$
7		$f_{fe,Glove}(input)$	$t'_{en,earthquakes,Glove}$
8		All	$t'_{en,earthquakes,all}$
9	$t^n_{en,hurricanes}$	$f_{fe,word2vec}(input)$	$t'_{en,hurricanes,word2vec}$
10		$f_{fe,fastText}(input)$	$t'_{en,hurricanes,fastText}$
11		$f_{fe,Glove}(input)$	$t'_{en,hurricanes,Glove}$
12		All	$t'_{en,hurricanes,all}$
13	$t^n_{en,floods}$	$f_{fe,word2vec}(input)$	$t'_{en,floods,word2vec}$
14		$f_{fe,fastText}(input)$	$t'_{en,floods,fastText}$
15		$f_{fe,Glove}(input)$	$t'_{en,floods,Glove}$
16		All	$t'_{en,floods,all}$
17	$t^n_{id,earthquakes}$	$f_{fe,word2vec}(input)$	$t'_{id,earthquakes,word2vec}$
18		$f_{fe,fastText}(input)$	$t'_{id,earthquakes,fastText}$
19		$f_{fe,Glove}(input)$	$t'_{id,earthquakes,Glove}$
20		All	$t'_{id,earthquakes,all}$
21	$t^n_{id,floods}$	$f_{fe,word2vec}(input)$	$t'_{id,floods,word2vec}$
22		$f_{fe,fastText}(input)$	$t'_{id,floods,fastText}$
23		$f_{fe,Glove}(input)$	$t'_{id,floods,Glove}$
24		All	$t'_{id,floods,all}$
25	$t^n_{id,forest\ fires}$	$f_{fe,word2vec}(input)$	$t'_{id,forest\ fires,word2vec}$
26		$f_{fe,fastText}(input)$	$t'_{id,forest\ fires,fastText}$
27		$f_{fe,Glove}(input)$	$t'_{id,forest\ fires,Glove}$
28		All	$t'_{id,forest\ fires,all}$

The classification results were obtained by developing models using the 21 structured datasets presented in Table 5. Two classification algorithms were employed, namely GCN and GAT. The GCN and GAT architectures used in this study were configured as shown in Table 6, which also lists the software environment used to implement all phases of this study, including preprocessing, feature extraction, classification, and model performance evaluation. Additionally, the table provides information on the hardware specifications used to execute the program.

Table 6. Research Setup

	GCN	GAT
Layer	2	2
Hidden Layer	64	64
Activation	ReLu	ReLu
Output	Softmax	Softmax
Cosine Similarity	0.5	0.5
Epoch	100	100
Learning Rate	0.01	0.01
Optimizer	Adam	Adam
Hardware	Hardware's specification that using in this research: • CPU: Intel Xeon CPU with 2 vCPUs 2.20 GHz • RAM: Approximately 12GB of RAM. • GPU: NVIDIA Tesla K80 with 12GB of GDDR5 VRAM. • TPU: TPUs v2 with 8GB of HBM • Disk Space: 100GB.	
Software	Software that is used in this research is Python with these libraries: • re • NumPy • pandas • Gensim • PyTorch • scikit-learn	

Combining the two algorithms with the input datasets resulted in the construction of 41 models. The classification performance of each model is reported in Table 7, Table 8, Table 9, and Table 10. The evaluation is done using accuracy and F1-score metrics. The best-performing result for each dataset is highlighted in bold.

Table 7 shows that GCN performance varies across the English disaster datasets depending on the embedding method used. For Wildfires, the combined embedding input achieves the highest accuracy (93%) and F1-score (0.79). In Earthquakes, FastText performs best with 81% accuracy and an F1-score of 0.44. For Hurricanes, Word2Vec yields the strongest result (80%, F1 = 0.68), while in Floods, FastText again provides the highest performance (75.5%, F1 = 0.71).

Table 7. Experiment result for the English dataset using GCN

Natural Disaster	Input	Accuracy (%)	F1-score
Wildfires	$t'_{en,wildfires,word2vec}$	92	0.79
	$t'_{en,wildfires,fastText}$	91.5	0.75
	$t'_{en,wildfires,Glove}$	91.5	0.71
	$t'_{en,wildfires,all}$	**93**	**0.79**
Earthquakes	$t'_{en,earthquakes,word2vec}$	80.5	0.33
	$t'_{en,earthquakes,fastText}$	**81**	**0.44**
	$t'_{en,earthquakes,Glove}$	80	0.3
	$t'_{en,earthquakes,all}$	80	0.3
Hurricanes	$t'_{en,hurricanes,word2vec}$	**80**	**0.68**
	$t'_{en,hurricanes,fastText}$	78.5	0.64
	$t'_{en,hurricanes,Glove}$	80	0.67
	$t'_{en,hurricanes,all}$	75.5	0.5
Floods	$t'_{en,floods,word2vec}$	72.67	0.69
	$t'_{en,floods,fastText}$	75.5	0.71
	$t'_{en,floods,Glove}$	69.5	0.63
	$t'_{en,floods,all}$	**76.5**	**0.71**

Table 8 shows that GAT consistently achieves strong performance across all English disaster datasets, with the best results varying by embedding method. For Wildfires, the combined embedding input achieves 97% accuracy and an F1-score of 0.93. In Earthquakes, FastText achieves the best performance (96%, F1 =

0.92), while in Hurricanes, both GloVe and the combined embeddings yield the highest F1-score of 0.94. For Floods, the combined embedding input again delivers the strongest results with 96% accuracy and an F1-score of 0.96.

Table 8. Experiment Result for the English Dataset using GAT

Natural Disaster	Input	Accuracy (%)	F1-score
Wildfires	$t'_{en,wildfires,word2vec}$	96.5	0.92
	$t'_{en,wildfires,fastText}$	96	0.91
	$t'_{en,wildfires,Glove}$	91.5	0.89
	$t'_{en,wildfires,all}$	**97**	**0.93**
Earthquakes	$t'_{en,earthquakes,word2vec}$	80.5	0.65
	$t'_{en,earthquakes,fastText}$	**96**	**0.92**
	$t'_{en,earthquakes,Glove}$	97.5	0.85
	$t'_{en,earthquakes,all}$	91.5	0.72
Hurricanes	$t'_{en,hurricanes,word2vec}$	90.5	0.88
	$t'_{en,hurricanes,fastText}$	84.5	0.78
	$t'_{en,hurricanes,Glove}$	**95**	**0.94**
	$t'_{en,hurricanes,all}$	94.5	0.94
Floods	$t'_{en,floods,word2vec}$	77.67	0.74
	$t'_{en,floods,fastText}$	88	0.86
	$t'_{en,floods,Glove}$	90	0.75
	$t'_{en,floods,all}$	**96**	**0.96**

Table 9 shows that GCN achieves strong performance on the Indonesian datasets, with the best results varying across embedding methods. For Earthquakes, FastText produces the highest accuracy (90.33%) and F1-score (0.90). In the Flood dataset, Word2Vec achieves the best performance with 91.67% accuracy and an F1-score of 0.92. For Forest Fires, Word2Vec again yields the strongest results (95%, F1 = 0.95), outperforming FastText, GloVe, and the combined embeddings.

Table 9. Experiment Result for the Indonesia Dataset using GCN

Natural Disaster	Input	Accuracy (%)	F1-score
Earthquakes	$t'_{id,earthquakes,word2vec}$	85	0.85
	$t'_{id,earthquakes,fastText}$	**90.33**	**0.9**
	$t'_{id,earthquakes,Glove}$	86.33	0.86
	$t'_{id,earthquakes,all}$	84.67	0.85
Flood	$t'_{id,floods,word2vec}$	**91.67**	**0.92**
	$t'_{id,floods,fastText}$	88.67	0.88
	$t'_{id,floods,Glove}$	88.33	0.88
	$t'_{id,floods,all}$	85.33	0.85
Forest fires	$t'_{id,forest\ fires,word2vec}$	**95**	**0.95**
	$t'_{id,forest\ fires,fastText}$	94.33	0.94
	$t'_{id,forest\ fires,Glove}$	92	0.92
	$t'_{id,forest\ fires,all}$	94.33	0.94

Table 10 demonstrates that GAT achieves near-perfect performance across all Indonesian disaster datasets. For Earthquakes, both Word2Vec and FastText yield the highest results with 99.67% accuracy and an F1-score of 0.997. In the Floods dataset, FastText and GloVe each achieve 99.33% accuracy and an F1-score of 0.993. For Forest Fires, the best performance is achieved with Word2Vec and the combined embedding input, both achieving 99.67% accuracy and an F1-score of 0.997.

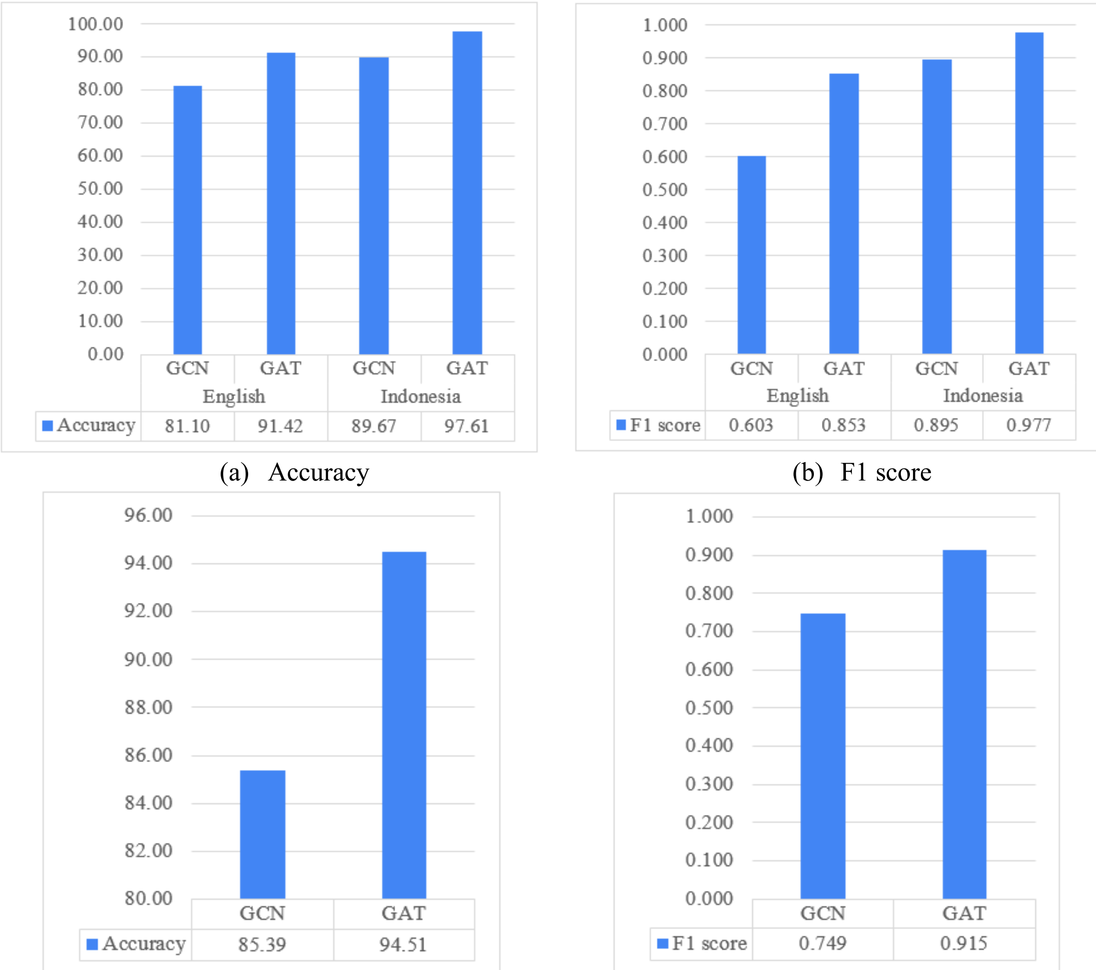
Table 10. Experiment Result for the Indonesia Dataset using GAT

Natural Disaster	Input	Accuracy (%)	F1-score
Earthquakes	$t'_{id,earthquakes,word2vec}$	**99.67**	**0.997**
	$t'_{id,earthquakes,fastText}$	**99.67**	**0.997**
	$t'_{id,earthquakes,Glove}$	97	0.97
	$t'_{id,earthquakes,all}$	92.67	0.93
Floods	$t'_{id,floods,word2vec}$	96.33	0.96

Natural Disaster	Input	Accuracy (%)	F1-score
Forest fires	$t'_{id,floods,fastText}$	**99.33**	**0.993**
	$t'_{id,floods,Glove}$	**99.33**	**0.993**
	$t'_{id,floods,all}$	92.33	0.92
	$t'_{id,forest\ fires,word2vec}$	**99.67**	**0.997**
	$t'_{id,forest\ fires,fastText}$	97.67	0.98
	$t'_{id,forest\ fires,Glove}$	98	0.98
	$t'_{id,forest\ fires,all}$	**99.67**	**0.997**

3.2 Discussion

Fig. 2 presents a comparative analysis of GCN and GAT performance in classifying natural disaster-related messages, evaluated using accuracy and F1-score metrics.



(a) Accuracy (b) F1 score

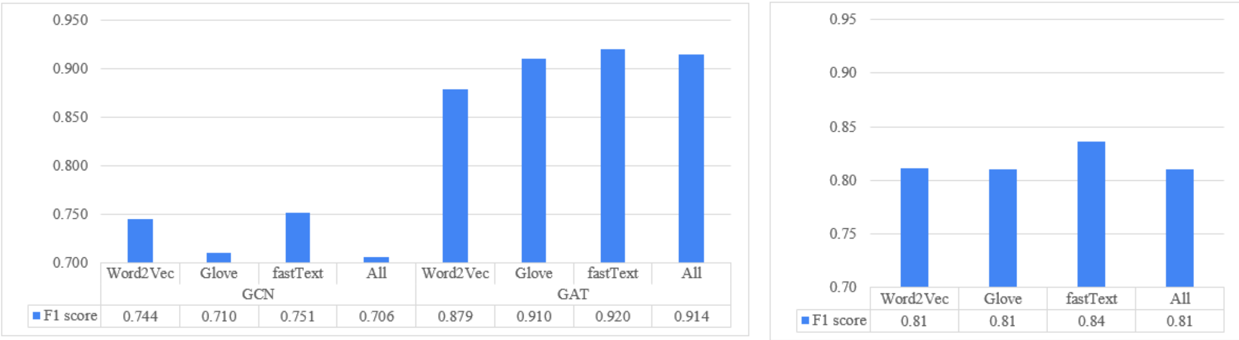
(c) Accuracy **(d)** F1 score

Figure 2. Comparison of GCN and GAT Models' Performance.

Fig. 2 (a) and Fig. 2 (b) depict the average performance of GCN and GAT models based on the language used in the dataset, while Fig. 2 (c) and Fig. 2 (d) show the overall average performance of both models. In general, GAT consistently outperforms GCN across all scenarios. This advantage is attributed to GAT's adaptive capability to determine the importance of weights between nodes through a self-attention mechanism. Unlike GCN, which relies on static adjacency matrices and linear aggregation, GAT dynamically computes the contribution of each neighboring node using attention coefficients. This enables GAT to selectively emphasize more relevant nodes, particularly when processing word embedding-based data with high semantic variability, producing more informative feature representations and improving classification accuracy.

Fig. 2 (a) and Fig. 2 (b) also reveal that English-language datasets exhibit lower F1 scores than accuracy. This discrepancy is due to the imbalanced class distribution in those datasets. While accuracy measures the proportion of correct predictions over all predictions regardless of class distribution, the F1-score — which combines precision and recall — is a more sensitive measure of model performance across classes. A lower F1-score than accuracy indicates that, although the model performs well overall, its ability to detect minority classes remains limited.

Fig. 3 illustrates the average performance of models based on the word embedding techniques used to construct structured input data for the classification algorithms.



(a) (b)

Figure 3. Comparison of the Performance of Models by Word Embedding.

Fig. 3 (a) shows the performance of GCN and GAT models based on the input derived from different word embedding methods. The results demonstrate that structured data based on fastText yields the highest-performing models compared to other embedding techniques. Fig. 3 (b) presents the average F1-scores across all model combinations, with fastText consistently achieving the best results compared to Word2Vec and GloVe. The superior performance of fastText-based models is due to their subword-level modeling via character n-grams, which allows them to better capture morphological structures and handle rare, misspelled, or out-of-vocabulary words. In contrast, Word2Vec and GloVe only generate representations for words seen during training. As a result, fastText provides richer, more complete word vectors, enabling better-structured sentence-level representations.

Previous studies have also attempted to combine sentence vectors from multiple embedding techniques, motivated by the complementary semantic information that each method can offer, potentially improving model performance [7], [8], [29]. In this study, combined input data using all three embedding techniques were also tested, with performance results presented under in the "All" columns of Fig. 3 (a) and Fig. 3 (b). However, the combination of embeddings did not consistently yield superior performance, and in most cases, the average results were lower than those obtained using individual embedding methods. This suggests that naive concatenation of multiple embeddings without additional processing can lead to redundancy and increased feature dimensionality, thereby degrading model performance. Therefore, more sophisticated fusion approaches are necessary to fully leverage the benefits of multi-embedding integration.

The proposed method, which integrates word embedding-based inputs with the GAT architecture, outperforms previous state-of-the-art models for classifying natural disaster reports. A comparison of the best-performing models in this study with those from prior work is presented in Table 11.

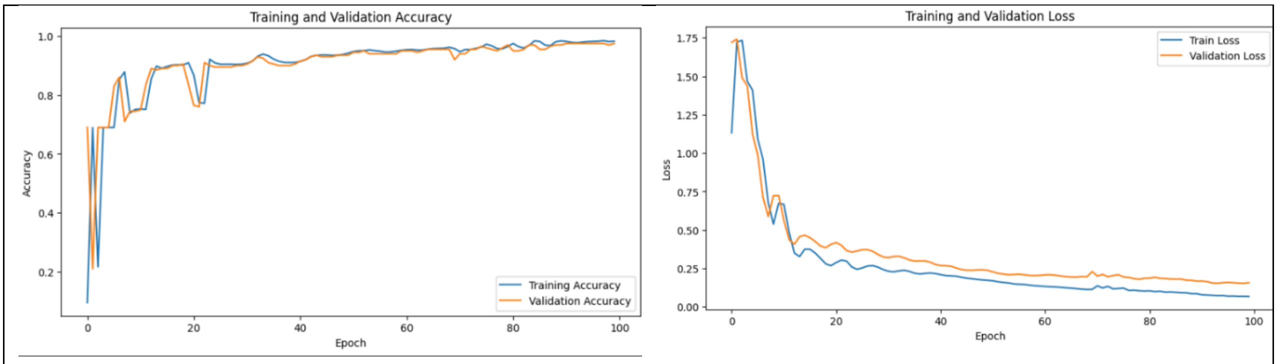
Table 11. Comparison of our Research to Previous Research.

Language	Dataset	Method	Accuracy (%)	F1 score
English	Wildfires	All + GAT (Our research)	97	0.93
		TF-IDF Random Forest [1]		0.813
		Word Embedding Random Forest [2]		0.835
		Word Embedding CNN [3]		0.722
		Word Embedding LSTM [3]	0.82	0.676
		BERT [3]	0.785	0.671
		TF-IDF GCN [24]	81	0.68
		TF-IDF GAT[25]	84.5	0.73
		mBERT [26]	84	0.8372
		XLM [27]	84	0.8312
		XLM-RoBERTa [28]	83.5	0.8325

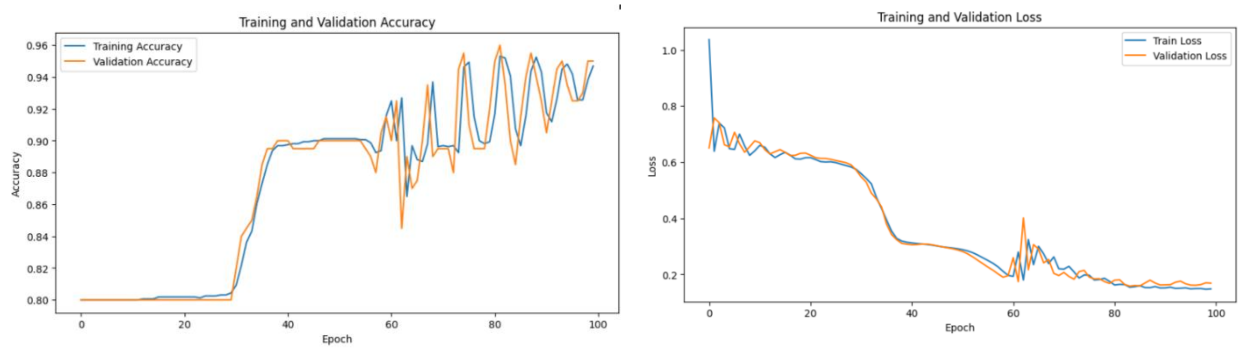
Language	Dataset	Method	Accuracy (%)	F1 score
	Earthquakes	fastText + GAT (Our research)	96	0.92
		TF-IDF Random Forest [1]		0.834
		Word Embedding Random Forest [2]		0.84
		Word Embedding CNN [3]		0.707
		Word Embedding LSTM [3]	0.845	0.673
		BERT [3]	0.845	0.638
		TF-IDF GCN [24]	85	0.6
		TF-IDF GAT[25]	88.5	0.71
		mBERT [26]	87	0.8685
		XLM [27]	84	0.8231
		XLM-RoBERTa [28]	82	0.7929
	Hurricanes	Glove + GAT (Our research)	95	0.94
		TF-IDF Random Forest [1]		0.819
		Word Embedding Random Forest [2]		0.84
		Word Embedding CNN [3]		0.691
		Word Embedding LSTM [3]	0.72	0.651
		BERT [3]	0.68	0.576
		TF-IDF GCN [24]	72.5	0.64
		TF-IDF GAT[25]	73	0.67
		mBERT [26]	76.5	0.7653
		XLM [27]	74.5	0.7479
		XLM-RoBERTa [28]	75.5	0.7436
	Floods	All + GAT (Our research)	96	0.96
		TF-IDF Random Forest [1]		0.735
		Word Embedding Random Forest [2]		0.83
		Word Embedding CNN [3]		0.759
		Word Embedding LSTM [3]	0.585	0.585
		BERT [3]	0.72	0.72
		TF-IDF GCN [24]	65	0.64
		TF-IDF GAT[25]	71.5	0.71
		mBERT [26]	68.5	0.6821
		XLM [27]	65.5	0.6542
		XLM-RoBERTa [28]	73	0.7298
Indonesia	Earthquakes	Word2Vec + GAT (Our research)	99.67	1
		TF-IDF GCN [24]	85.67	0.86
		TF-IDF GAT[25]	89.67	0.9
		mBERT [26]	78.67	0.7882
		XLM [27]	74	0.7433
		XLM-RoBERTa [28]	80	0.8008
		Glove + 2D CNN [7]	78.33	
	Floods	Glove + GAT (Our research)	99.33	0.99
		TF-IDF GCN [24]	84.33	0.86
		TF-IDF GAT[25]	88.33	0.88
		mBERT [26]	78.33	0.7828
		XLM [27]	74.67	0.7477
		XLM-RoBERTa [28]	84.33	0.8435
		All + 2D CNN [7]	78.33	

Language	Dataset	Method	Accuracy (%)	F1 score
	Forest Fires	Glove + GAT (Our research)	99.67	1
		TF-IDF GCN [24]	88.33	0.88
		TF-IDF GAT[25]	96	0.96
		mBERT [26]	86.67	0.8668
		XLM [27]	85.67	0.8553
		XLM-RoBERTa [28]	88.67	0.8867
		Glove + 2D CNN [7]	81.97	

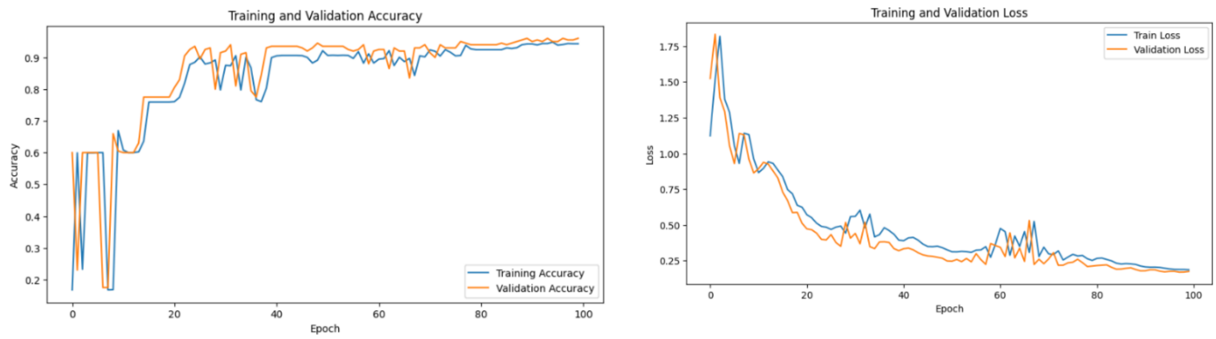
This research produced seven top-performing models, corresponding to the seven datasets utilized. The training results of the best models using English-language datasets are illustrated in Fig. 4. In Fig. 4 (a), the accuracy on the Wildfires dataset increases rapidly over the first 20 epochs and then stabilizes at 0.95–1.00, with closely aligned training and validation curves and a steadily decreasing loss, indicating stable convergence and minimal overfitting. Fig. 4 (b) shows slower, less stable learning for earthquakes, with accuracy remaining low until around epoch 25 and exhibiting strong fluctuations thereafter; the loss curves also oscillate, suggesting sensitivity to class imbalance or learning rate settings. In Fig. 4 (c), the Hurricanes dataset shows faster, more stable convergence, with accuracy rising quickly and stabilizing around 0.90–0.95, while the loss decreases smoothly with minor fluctuations. Fig. 4 (d) shows that the Flood dataset converges more gradually and exhibits greater variability in both accuracy and loss, indicating noisier samples and a more challenging classification task.



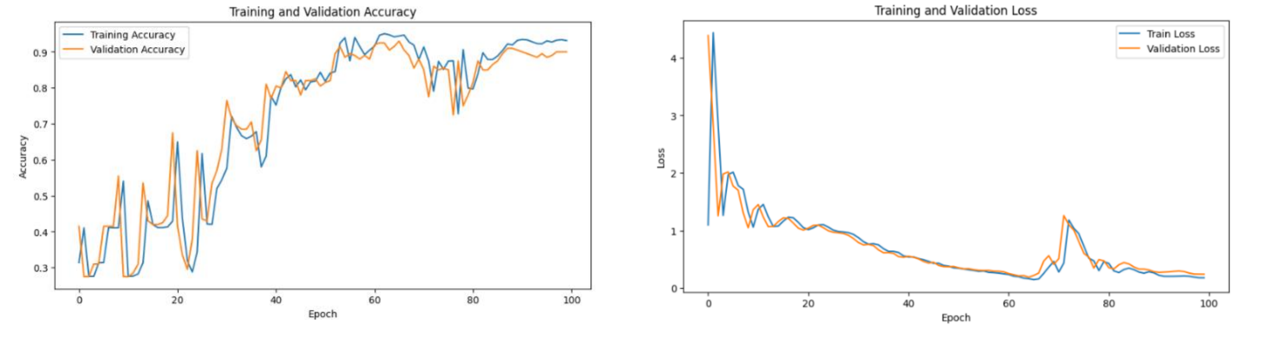
(a) Wildfires



(b) Earthquakes



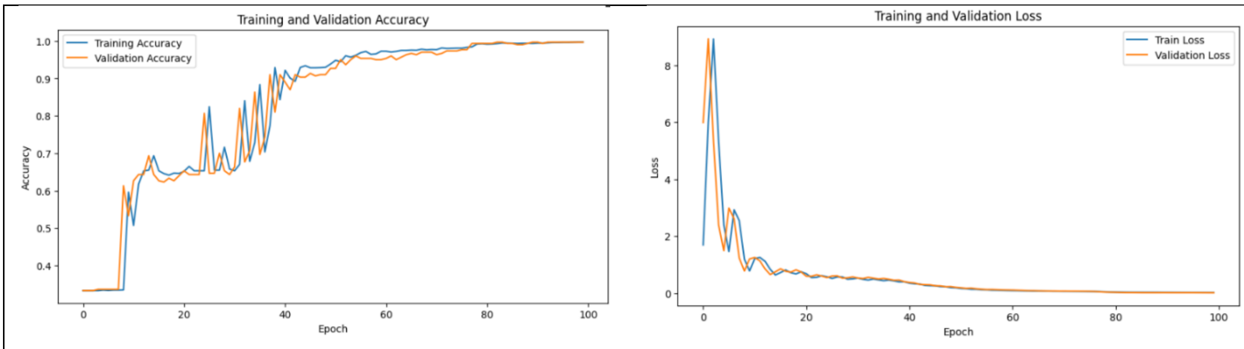
(c) Hurricanes



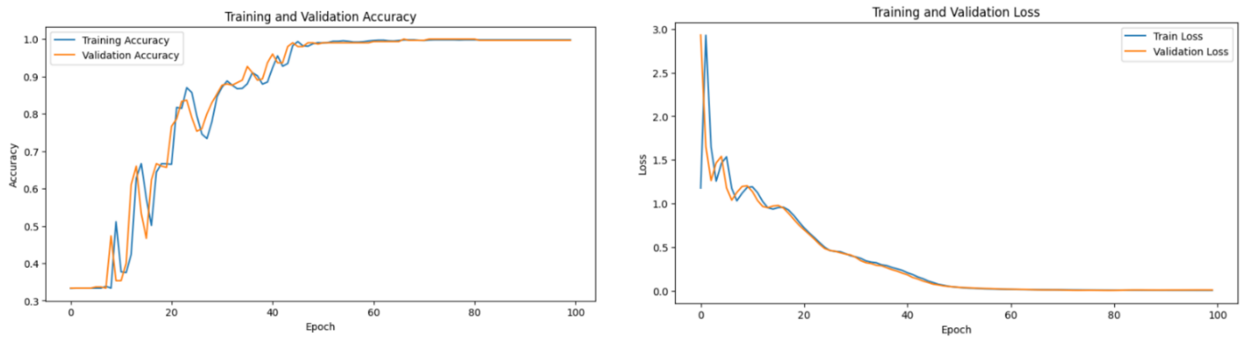
(d) Flood

Figure 4. Accuracy and Loss of Models with the English dataset

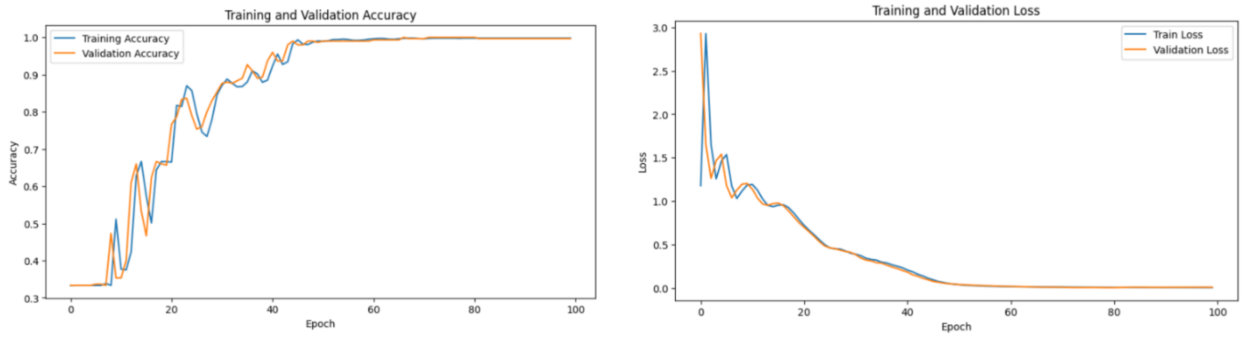
In contrast, the training results of the best models on Indonesian-language datasets are shown in Fig. 5. Fig. 5 (a) shows rapid accuracy improvement during the first 20 epochs, followed by stable convergence near 1.0. Training and validation curves closely overlap, and the loss decreases smoothly, indicating strong generalization and minimal overfitting. In Fig. 5 (b), accuracy steadily increases and stabilizes at near-perfect levels, with the training and validation curves almost identical. Fig. 5 (c) exhibits a similar pattern, with accuracy rising sharply and quickly approaching near-perfect performance. Training and validation losses converge smoothly with minimal fluctuation, indicating highly stable training dynamics and excellent generalization on the Forest Fire dataset.



(a) Earthquakes



(b) Floods



(c) Forest Fires

Figure 5. Accuracy and Loss of Models with the dataset in the Indonesian language

Fig. 4 and Fig. 5 demonstrate that models built with the Graph Attention Network (GAT) architecture perform well at classifying text related to natural disasters. This is evidenced by the consistently decreasing trends in the training and validation loss curves and the high, stable accuracy scores. No signs of overfitting are observed, as the validation loss does not increase significantly and instead follows the downward trend of the training loss. Moreover, the training and validation accuracy curves support the conclusion that the models do not overfit and can classify data with consistently high accuracy.

Despite the overall strong performance, the results in Fig. 4 and Fig. 5 also highlight several potential limitations. One indication is the fluctuation in accuracy during the early and middle epochs, which may be due to a high learning rate or class imbalance in the datasets. This suggests that the models require more time to converge and learn consistent patterns. Additionally, slight overfitting may be present, as evidenced by validation loss being marginally higher than training loss during the mid-to-late training phases. Some models also exhibit instability, particularly in the validation accuracy, which fluctuates sharply across certain epochs. These findings point to opportunities for further research to optimize model performance.

4. CONCLUSION

The conclusions of this research are:

1. This study provides empirical evidence that Graph Neural Network (GNN) architectures—specifically Graph Convolutional Network (GCN) and Graph Attention Network (GAT)— are effective for multilingual disaster tweet classification when combined with structured word embeddings.
2. Experimental evaluations across seven datasets covering multiple disaster types, conducted in English and Indonesian, consistently indicate that GAT surpasses GCN in both accuracy and F1-score. Notably, GAT achieved F1-scores of 0.93 (wildfires), 0.92 (earthquakes), 0.94 (hurricanes), and 0.96 (floods) in English, and 1.00 (earthquakes), 0.99 (floods), and 1.00 (forest fires) in Indonesian.
3. Among 56 evaluated models, FastText embeddings demonstrated the highest standalone performance. In contrast, concatenating multiple embeddings often led to reduced performance, likely due to feature redundancy and the curse of dimensionality.
4. The key contribution of this work lies in the novel application of GAT with structured word embeddings, thereby establishing a new multilingual benchmark for disaster tweet classification. Compared to conventional machine learning and CNN/LSTM-based models, GAT-based approaches more effectively capture contextual and relational information in short, informal text sequences.
5. Despite these strong results, several limitations emerged during model training. Accuracy fluctuations observed in the early and middle epochs suggest that the models require longer convergence times, which may be influenced by learning rate settings or class imbalance in the datasets. Mild overfitting is also indicated by validation loss that remains slightly higher than training loss in certain epochs. Furthermore, several models show instability in validation accuracy, with noticeable fluctuations across training iterations. These findings highlight that, while effective, the proposed models are not yet fully optimized for stability and generalization.
6. Based on these limitations, future research should explore more advanced embedding-fusion mechanisms—such as attention-based fusion, gating techniques, or dimensionality-reduction strategies—to better handle redundancy in multi-embedding representations. Additional improvements may be achieved by applying adaptive learning-rate schedules, stronger regularization techniques, and class-balancing methods such as focal loss or data augmentation. Future work should also validate model robustness on larger, more diverse, and real-time datasets. Incorporating multimodal features (e.g., combining text with images) and temporal patterns from social media streams may further enhance the practical reliability of GNN-based disaster monitoring systems.

Author Contributions

Mohammad Reza Faisal: Conceptualization; Methodology, Writing - Original Draft. Irwan Budiman: Data Curation; Software. Dodon Turianto Nugrahadi: Resources; Investigation; Validation. Muhammad Rafi: Software; Data Curation. Mera Kartika Delimayanti: Supervision; Writing - Review and Editing, Visualization. Luu Duc Ngo: Software; Validation. Moses Okechukwu Onyesolub: Supervision; Writing - Reviewing and Editing. All authors discussed the results and contributed to the final manuscript.

Funding Statement

This research was supported by funding from the DRTPM Research Program of Indonesia's Ministry of Education, Culture, Research, and Technology. Main Contract Number: 056/E5/PG.02.00.PL/2024, Derivative Contract Number: 1026/UN8.2/PG/2024.

Acknowledgment

The authors would like to express their sincere gratitude to all individuals and institutions that contributed to this research. Special appreciation is extended to the researchers from Bac Lieu University (Vietnam) and Nnamdi Azikiwe University (Nigeria) for their valuable collaboration and contributions to this study.

Declarations

The authors declare that there is no conflict of interest regarding the publication of this paper.

Declaration of Generative AI and AI-assisted technologies

Generative AI tools (e.g., ChatGPT) were used solely for language refinement (grammar, spelling, and clarity). The scientific content, analysis, interpretation, and conclusions were developed entirely by the authors. The authors reviewed and approved all final text.

REFERENCES

[1] K. Zahra, M. Imran, and F. O. Ostermann, "AUTOMATIC IDENTIFICATION OF EYEWITNESS MESSAGES ON TWITTER DURING DISASTERS," *Inf Process Manag*, vol. 57, no. 1, pp. 102–107, 2020, doi: https://doi.org/10.1016/j.ipm.2019.102107

[2] M. R. Faisal, R. A. Nugroho, R. Ramadhani, F. Abadi, R. Herteno, and T. H. Saragih, "NATURAL DISASTER ON TWITTER: ROLE OF FEATURE EXTRACTION METHOD OF WORD2VEC AND LEXICON BASED FOR DETERMINING DIRECT EYEWITNESS," *Trends in Sciences*, vol. 18, no. 23, p. 680, 2021, doi: https://doi.org/10.48048/tis.2021.680

[3] I. Budiman, M. R. Faisal, F. Abadi, M. Haekal, and D. T. Nugrahadi, "A COMPARISON OF WORD EMBEDDING-BASED EXTRACTION FEATURE TECHNIQUES AND DEEP LEARNING MODELS OF NATURAL DISASTER MESSAGES CLASSIFICATION," *Journal of Computer Sciences Institute*, vol. 27, pp. 145–153, 2023, doi: https://doi.org/10.35784/jcsi.3322

[4] K. Y. Firlia, M. R. Faisal, D. Kartini, R. A. Nugroho, and F. Abadi, "ANALYSIS OF NEW FEATURES ON THE PERFORMANCE OF THE SUPPORT VECTOR MACHINE ALGORITHM IN CLASSIFICATION OF NATURAL DISASTER MESSAGES," in *2021 4th International Conference of Computer and Informatics Engineering (IC2IE)*, pp. 317–322, 2021, doi: https://doi.org/10.1109/IC2IE53219.2021.9649107

[5] M. K. Delimayanti, R. Sari, M. Laya, M. R. Faisal, Pahrul, and R. F. Naryanto, "THE EFFECT OF PRE-PROCESSING ON THE CLASSIFICATION OF TWITTER'S FLOOD DISASTER MESSAGES USING SUPPORT VECTOR MACHINE ALGORITHM," *Proceedings of ICAE 2020 - 3rd International Conference on Applied Engineering*, no. February 2021, 2020, doi: https://doi.org/10.1109/ICAE50557.2020.9350387

[6] M. R. Faisal, I. Budiman, F. Abadi, D. T. Nugrahadi, M. Haekal, and I. Sutedja, "APPLYING FEATURES BASED ON WORD EMBEDDING TECHNIQUES TO 1D CNN FOR NATURAL DISASTER MESSAGES CLASSIFICATION," *2022 5th International Conference on Computer and Informatics Engineering, IC2IE 2022*, no. December, pp. 192–197, 2022, doi: https://doi.org/10.1109/IC2IE56416.2022.9970188

[7] M. R. Faisal *et al.*, "USING SOCIAL MEDIA DATA TO MONITOR NATURAL DISASTER: A MULTI DIMENSION CONVOLUTIONAL NEURAL NETWORK APPROACH WITH WORD EMBEDDING," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 6, pp. 1037–1046, 2022, doi: https://doi.org/10.29207/resti.v6i6.4525

[8] M. R. Faisal *et al.*, "3D WORD EMBEDDING VECTOR FEATURE EXTRACTION AND HYBRID CNN-LSTM FOR NATURAL DISASTER REPORTS IDENTIFICATION," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 22, no. 5, pp. 1196–1208, 2024, doi: https://doi.org/10.12928/telkomnika.v22i5.26091

[9] H. Ren *et al.*, "GRAPH CONVOLUTIONAL NETWORKS IN LANGUAGE AND VISION: A SURVEY," *Knowl Based Syst*, vol. 251, 2022, doi: https://doi.org/10.1016/j.knosys.2022.109250

[10] A. G. Vrahatis, K. Lazaros, and S. Kotsiantis, "GRAPH ATTENTION NETWORKS: A COMPREHENSIVE REVIEW OF METHODS AND APPLICATIONS," *Future Internet*, vol. 16, no. 9, 2024, doi: https://doi.org/10.3390/fi16090318

[11] S. A. Zikrina and Fitriyani, "IMPROVING CYBERBULLYING DETECTION WITH GRAPH NEURAL NETWORK APPROACHES," in *ICADEIS 2025 - 2025 International Conference on Advancement in Data Science, E-learning and Information System: Integrating Data Science and Information System, Proceeding*, 2025, doi: https://doi.org/10.1109/ICADEIS65852.2025.10933470

[12] K. Zhao, L. Huang, R. Song, Q. Shen, and H. Xu, "A SEQUENTIAL GRAPH NEURAL NETWORK FOR SHORT TEXT CLASSIFICATION," *Algorithms*, vol. 14, no. 12, 2021, doi: https://doi.org/10.3390/a14120352

[13] Z. Chen, K. Huang, L. Wu, Z. Zhong, and Z. Jiao, "RELATIONAL GRAPH CONVOLUTIONAL NETWORK FOR TEXT-MINING-BASED ACCIDENT CAUSAL CLASSIFICATION," *Applied Sciences*, vol. 12, no. 5, 2022, doi: https://doi.org/10.3390/app12052482

[14] Y. Xi, J. Wang, Y. Tang, S. Qiao, R. Cao, and X. Liu, "RESEARCH ON IDENTIFICATION OF NETWORK PUBLIC OPINION INFORMATION BASED ON GRAPH CONVOLUTIONAL NETWORKS," in *Proceedings - 2020 International Conference on Intelligent Computing and Human-Computer Interaction, ICHCI 2020*, pp. 409 – 418, 2020, doi: https://doi.org/10.1109/ICHCI51889.2020.00092

[15] B. Zhang, Q. He, and D. Zhang, "HETEROGENEOUS GRAPH NEURAL NETWORK FOR SHORT TEXT CLASSIFICATION," *Applied Sciences*, vol. 12, no. 17, 2022. https://doi.org/10.3390/app12178711

[16] C. P. Chai, "COMPARISON OF TEXT PREPROCESSING METHODS," *Nat Lang Eng*, vol. 29, no. 3, pp. 509–553, 2023, doi: https://doi.org/10.1017/S1351324922000213

[17] M. Alirridlo, A. F. Septiyanto, R. Sarno, and D. Sunaryono, "A COMPARATIVE ANALYSIS OF TEXT NORMALIZATION TECHNIQUES FOR ENHANCED SENTIMENT ANALYSIS PERFORMANCE," in *2024 Beyond Technology Summit on Informatics International Conference (BTS-I2C)*, pp. 474–479, 2024, doi: https://doi.org/10.1109/BTS-I2C63534.2024.10942038

[18] M. George, "IMPROVING SENTIMENT ANALYSIS OF FINANCIAL NEWS HEADLINES USING HYBRID WORD2VEC-TFIDF FEATURE EXTRACTION TECHNIQUE," *Procedia Comput Sci*, vol. 244, pp. 1–8, 2024, doi: https://doi.org/10.1016/j.procs.2024.10.172

[19] G. Lachaud, P. Conde-Cespedes, and M. Trocan, "MATHEMATICAL EXPRESSIVENESS OF GRAPH NEURAL NETWORKS," *Mathematics*, vol. 10, no. 24, 2022, doi: https://doi.org/10.3390/math10244770

[20] C. Sun *et al.*, "ATTENTION-BASED GRAPH NEURAL NETWORKS: A SURVEY," *Artif Intell Rev*, vol. 56, no. Suppl 2, pp. 2263–2310, 2023, doi: https://doi.org/10.1007/s10462-023-10577-2

[21] S. Wang, H. Lu, A. Khan, F. Hajati, M. Khushi, and S. Uddin, "A MACHINE LEARNING SOFTWARE TOOL FOR MULTICLASS CLASSIFICATION," *Software Impacts*, vol. 13, p. 100383, 2022, doi: https://doi.org/10.1016/j.simpa.2022.100383

[22] J.-P. van Zyl and A. P. Engelbrecht, "ANALYSIS OF CLASSIFICATION METRIC BEHAVIOUR UNDER CLASS IMBALANCE," *Egyptian Informatics Journal*, vol. 31, p. 100711, 2025. https://doi.org/10.1016/j.eij.2025.100711

[23] M. R. Faisal *et al.*, "LSTM AND BI-LSTM MODELS FOR IDENTIFYING NATURAL DISASTERS REPORTS FROM SOCIAL MEDIA," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 5, no. 4, 2023, doi: https://doi.org/10.35882/jeeemi.v5i4.319

[24] S. Li, Y. Wang, H. Huang, and Y. Zhou, "STUDY ON THE RUMOR DETECTION OF SOCIAL MEDIA IN DISASTER BASED ON MULTI-FEATURE FUSION METHOD," *Natural Hazards*, vol. 120, no. 4, pp. 4011–4030, 2024, doi: https://doi.org/10.1007/s11069-023-06284-4

[25] N. R. Paul and R. C. Balabantaray, "DISASTER RELATED TWEET CLASSIFICATION METHOD BASED ON BERT AND GAT," *International Journal of Information Technology*, pp. 1–13, 2025, doi: https://doi.org/10.1007/s41870-024-02389-6

[26] S. M. Jeba, T. T. Aurpa, and M. R. S. Adib, "FROM FACEBOOK POSTS TO NEWS HEADLINES: USING TRANSFORMER MODELS TO PREDICT POST-DISASTER IMPACT ON MASS MEDIA CONTENT," *Soc Netw Anal Min*, vol. 14, no. 1, p. 200, 2024, doi: https://doi.org/10.1007/s13278-024-01363-1

[27] F. Vitiugin and H. Purohit, "MuLTILINGUAL SERVICEABILITY MODEL FOR DETECTING AND RANKING HELP REQUESTS ON SOCIAL MEDIA DURING DISASTERS," in *Proceedings of the International AAAI Conference on Web and Social Media*, pp. 1571–1584, 2024, doi:https://doi.org/10.1609/icwsm.v18i1.31410

[28] A. Katyshev, A. Anikin, and A. Zubankov, "BIDIRECTIONAL TRANSFORMERS AS A MEANS OF EFFICIENT BUILDING OF KNOWLEDGE BASES: A CASE STUDY WITH XLM-ROBERTA," in *Novel \& Intelligent Digital Systems Conferences*, pp. 292–297, 2023, doi:https://doi.org/10.1007/978-3-031-44146-2_30

[29] N. Badri, F. Kboubi, and A. H. Chaibi, "COMBINING FASTTEXT AND GLOVE WORD EMBEDDING FOR OFFENSIVE AND HATE SPEECH TEXT DETECTION," *Procedia Comput Sci*, vol. 207, pp. 769–778, 2022, doi: https://doi.org/10.1016/j.procs.2022.09.132