Disaster related social media content processing for sustainable cities

Pradeep Kumar Roy[a], Abhinav Kumar[b], Jyoti Prakash Singh[c,*], Yogesh Kumar Dwivedi[d,f], Nripendra Pratap Rana[e], Ramakrishnan Raman[f]

[a]*Department of Computer Science and Engineering, Indian Institute of Information Technology, Surat*
[b]*Department of Computer Science & Engineering, Sikhsa 'O' Anushandhan (Deemed to be University), Bhubaneswar, India*
[c]*Department of Computer Science and Engineering, National Institute of Technology Patna, India*
[d]*Emerging Markets Research Centre, School of Management, Swansea University, Bay Campus, Fabian Way, Swansea, UK*
[e]*College of Business and Economics, Qatar University, Doha, Qatar*
[f]*Symbiosis Institute of Business Management, Pune & Symbiosis International (Deemed to be University), Pune, India*

Abstract

The current study offers a hybrid convolution neural networks (CNN) model that filters relevant posts and categorises them into several humanitarian classifications using both character and word embedding of textual content. The distinct embeddings for words and characters are used as input to the CNN model's various channels. A hurricane, flood, and wildfire dataset are used to validate the proposed model. The model performed similarly across all datasets, with the F1-score ranging from 0.66 to 0.71. Because it uses existing social media posts and may be used as a layer with any social media, the model provides a sustainable solution for disaster analysis. With domain-specific training, the suggested approach can be used to locate useful information in other domains such as traffic accidents and civil unrest also.

Keywords: Disaster, Twitter, Deep Learning, CNN, Word Embedding, Character Embedding

1. Introduction

Disasters are adverse natural events that cause loss of life and property damage, such as floods, hurricanes, earthquakes, tsunamis, and storms, and are a significant impediment to the development of cities and societies. Every city's sustainable development should include strategies for avoidance and effective disaster mitigation. The use of existing infrastructures and technologies for disaster mitigation may provide a more cost-effective and sustainable solution because the cost of deployment and maintenance will be minimal, with little impact on city ecosystems. Social networking, which began as a chit-chat app, has made inroads into modern cultures and cities. They are being used by city dwellers to share their status, highlight their accomplishments, show consent and dissent to government policies and regulations [1, 2], and report any unwanted events and disasters [3] with text, images, and video messages. Businesses use social media to sell

*Corresponding Author: Jyoti Prakash Singh
Email addresses: pkroynitp@gmail.com (Pradeep Kumar Roy), abhinavkumar@soa.ac.in (Abhinav Kumar), jps@nitp.ac.in (Jyoti Prakash Singh), y.k.dwivedi@swansea.ac.uk (Yogesh Kumar Dwivedi), nrananp@gmail.com (Nripendra Pratap Rana), director@sibmpune.edu.in (Ramakrishnan Raman)

their products and services, while government organisations use it to raise public awareness about various societal concerns and policies. Many news stories are first reported on social media platforms before being published on news channels in recent years, as citizens rapidly share what they see and feel on social media. Because smartphones and high-speed Internet are readily available, immediate sharing is now possible. In an emergency, such as a disaster, where people require quick assistance, this immediate communication is extremely beneficial. A natural disaster causes enormous damage and necessitates significant government and individual effort to recover [4, 5, 6]. During an emergency, rescue groups require essential damage information as well as a precise location in order to provide timely assistance to those who are affected.

However, immediate sharing has a disadvantage in that it overburdens social media by circulating the same content. Other individuals begin to express compassion for the victims, commend rescue organisations, and so forth. Although all of these communications are related to a disaster, they may not be very effective in terms of disaster mitigation. During an emergency, such as an earthquake, flood, tsunami, or other disaster scenarios, Twitter and Facebook get a large number of content [7, 8, 9, 10]. Twitter receives around 500 million tweets every day from its users on a variety of topics [1]. As a result, extracting useful data from social media is a difficult operation that is nearly impossible to complete manually due to its enormous volume and velocity. However, these posts are significant because many of them are made by eyewitnesses who share real-time images of the calamity. On February 07, 2021, a flood occurred in the Chamoli district of Uttarakhand (India)[2], which was recorded by many eyewitnesses via video and audio messages on Twitter[3].

Relevant or useful tweets include those that call for aid, provide damage information, photos of injured or deceased individuals, or inquire about family. It's nearly hard to manually process and extract useful tweets from millions of tweets. Furthermore, manual screening of disaster-related tweets required a significant amount of human labour, even if it was impossible to obtain all of them. As a result, there's a good risk that useful tweets may get lost in the shuffle. To close these gaps and collect nearly all disaster-related tweets, a strong system is required that can scan incoming tweets and automatically filter the informative ones from Twitter. Tweets on Twitter are restricted in the character count (currently 280), so users are compelled to utilise abbreviations, an irregular short form of the text with numerous typos [11, 6, 12], making it a difficult task to create a robust system that can collect informative tweets.

Some studies have been conducted utilising machine learning (ML) techniques to solve catastrophe-related challenges such as landslide [13] and forecasting population elimination during disaster [14, 15]. The primary drawback of the earlier works is that features must be extracted manually, which is a time-consuming procedure in which many essential textual elements are missed. For example, if the model fails to capture the meaning of the phrase, or if a word is misspelt, it is likely that it will not appear in the forecast.

[1] https://www.dsayce.com/social-media/
[2] https://en.wikipedia.org/wiki/2021_Uttarakhand_flood
[3] https://twitter.com/search?q=Chamoli&src=typed_query

To address these difficulties and capture the semantics of tweets, this study presents a deep learning-based framework capable of dealing with the issues that arise in a machine learning-based method. The convolutional neural network (CNN) model has been used successfully to solve Natural Language Processing (NLP) based problems such as spam detection [16], question answering [17, 18], sentiment analysis [19], disaster-related tweet classification [5], and others [20, 21]. To filter the informative tweets, this study used a hybrid CNN model. To better capture the semantics of the message, hybridization is done at the level of word and character embedding. To extract the features from the input, multiple size kernels that convolve across the embedded tweet matrix are utilised. The following are the research's key contributions:

- Proposed a hybrid CNN model to categorise tweets into disaster-related categories.
- Combining the character and word embedding methods to capture message semantics.
- To confirm the system's resilience, the suggested hybrid CNN model is trained and evaluated on a cross-disaster dataset.

The rest of the article is organized as follows: Section 2 is the literature review. Section 3 discusses the proposed hybrid CNN model. In Section 4, the experimental outcomes of the proposed model is presented, followed by the discussion on the obtained results in Section 5. Finally, Section 6 concludes the article.

2. Literature Review

This section discusses the works that use information gathered from a social media platform to get disaster-related information. Social media, particularly Twitter, has been utilised in customer satisfaction [22], transit rider debate analysis [2], and fake news detection [23]. Other researchers have utilised social media for urban analytical and geo-visual systems [24], as well as hazard response [3]. Many scholars have lately utilised social media to raise awareness about the issue on time [25, 26, 27]. To give immediate assistance to the victims, relief agencies required information about the tragedy, which could be obtained through a social media platform. As a result, using social media as a method for obtaining informative material is critical for humanitarian groups. Several researchers have proposed models for obtaining informative content from social media platforms [28, 11, 6, 12, 29, 30, 31, 32, 33]

Caragea et al. [34] developed a technique to categorise tweets on the Haiti earthquake. They utilised a dataset of 3,598 tweets that had been manually labelled. The dataset included 10 categories, including *medical emergency*, *people trapped*, *food shortage*, and *shelter required*. To categorise the tagged text, two techniques were proposed: (i) keyword classification and (ii) ML algorithm classification. Topical words and a bag of words techniques are utilised to extract the characteristics. The testing results indicated that the keywords-based classifier produced a $F1$-score of 0.47 for the best instance, whereas the ML classifier (SVM) achieved a $F1$-score of 0.59.

Verma et al. [35] created an automated situational awareness system. The dataset for the study was created by gathering four crises situations. Naive Bayes and Maximum Entropy were employed as classifiers. The Maximum Entropy classifier obtained an accuracy of 80% using retrieved linguistic characteristics as well as hand-annotated features. Cameron et al. [36], like [35], proposed an automated methodology for identifying situation awareness posts on Twitter. Imran et al. [37] surveyed the processing of textual material gathered from social networking platforms. Many datasets linked to the crisis [4] were provided to assist future researchers.

Imran et al. [29, 38, 39] presented various models for disaster message classification. Imran and his group created a system called Artificial Intelligence for Disaster Response (AIDR) [29] that is capable of categorising tweets into user-defined classifications in real-time. The model was evaluated using the Pakistan earthquake dataset and yielded an Area Under Curve (AUC) of 0.80. A Naive Bayes classifier was utilised in [38] to categorise tweets into several classes with textual features. For identifying the informative tweets, the model had the highest $F1$-score of 0.809. A model was built in [39] to extract disaster-related information from tweets. They utilised two datasets: Jolphin 2011 and Sandy 2012. The Jolphin 2011 dataset has 206,764 tweets, whilst the Sandy 2012 dataset contains 140,000 tweets. Tweets are classified into three categories: (i) personal, (ii) informative, and (iii) others. Furthermore, the informative tweets are divided into several categories. In the best-case scenario, their model had a detection rate of 91%.

Ashktorab et al. [31] suggested a three-stage method for filtering informative tweets. Their model consists of three phases: (i) classification, (ii) clustering, and (iii) extraction. To begin, tweets relating to harm and causalities are filtered using traditional machine learning classifiers such as Support Vector Machine (SVM), sparse Latent Dirichlet Allocation (sLDA), and Logistic Regression (LR). Second, tweets with similar contexts are put together using clustering techniques. Finally, the tokens and phrases are extracted to obtain information on the various sorts of crisis-related harms. To validate the model, they used a total of twelve crises datasets. The LR classifier produced a $F1$-score of 0.65 in the best scenario. Olteanu et al. [40] developed a lexicon of crisis-related terms that appear often in messages uploaded on the social network during various crisis situations. These lexicons were utilised to extract the new terms that characterised the situation automatically. Li et al. [41] proposed a domain adaptation technique in which a model was trained on a labelled dataset of one event and tested on an unlabeled dataset of another event. They applied Naive Bayes classifiers to two datasets (i) hurricane Sandy and (ii) the Boston Marathon attack to report the best result with an AUC value of 0.73.

Rudra et al. [42] built a strategy to categorise the Nepal earthquake dataset into several catastrophe categories. An abstract summary was also created from the classified messages. Huang and Xiao [43] developed the model using the hurricane sandy disaster dataset. For the best situation, they utilised logistic

[4] `https://crisisnlp.qcri.org/`

regression to get an $F1$-score of 0.66. Nguyen et al. [44] created a model that uses a deep learning framework to categorise tweets as (i) informative or (ii) non-informative. The informative messages are further divided into groups such as sympathy, damages, impacted persons, and others. Nguyen et al. [12] created another model that uses CNN to categorise messages into informative and non-informative categories. Their experiment demonstrated that out-of-event data may also be used to train the algorithm at the start of a disaster. Caragea et al. [44] built a similar model with a CNN network, in which tweets are categorised into informative and non-informative categories.

Aipe et al. [45] presented a multi-label classification model with a deep learning architecture. In addition to the tweet content, the hashtag, user-mentions, and keywords derived from the URL are all taken into account for model construction. The results of the experiments indicated that the extra characteristics have a beneficial effect on total predictions. The model's $F1$-score ranged from 0.75 to 0.98. Graf et al. [46] created a cross-domain categorization model . They retrieved a variety of characteristics, including emotional, sentimental, and linguistic ones. The model was trained and evaluated using twenty-six datasets, twenty-five of which were utilised for training and the remaining one for testing. Their model achieved an average accuracy of 80%. Yu et al. [30] utilised three catastrophe datasets for model development: (i) hurricane Harvey, (ii) hurricane Sandy, and (iii) hurricane Irma. The created model was evaluated with two settings, (i) event-specific dataset and (ii) out-of-event dataset, and achieved the best case $F1$-score of 0.80 utilising a CNN deep learning model.

The model proposed by Singh et al. [47] classified tweets as high-priority or low-priority. They validated their model using a flood dataset. They also predicted the user location using a hidden Markov model and reported a best-case location prediction accuracy of 87%. Kumar and Singh [5] created a CNN-based deep learning model to extract location references from emergency tweets. For the best situation, their model had a hamming loss of 0.002 and a $F1$-score of 0.96. Kumar et al. [48] presented an additional deep multimodal technique for informative content categorization. They employed transfer learning and Long-Short Term Memory to get a $F1$-score of 0.92 with the textual dataset. When the image and text were combined, the best-reported result was an $F1$-score of 0.93.

Madichetty et al. [49] developed a multi-modal method for detecting disaster-related relevant tweets that combine fine-tuned BERT and DenseNet. They tested their method on seven different catastrophe event datasets, obtaining $F1$-scores ranging from 0.66 to 0.88. Malla et al. [50] suggested an ensemble-based strategy for identifying COVID-19 related relevant tweets by combining the RoBERTa, BERTweet, and CT-BERT models. With an $F1$-score of 0.91, their method beat traditional machine and deep learning models. By merging the RoBERTa and feature-based techniques, Madichetty et al. [51] developed a neural-based strategy for detecting disaster-related situational tweets. They tested their model with different disasters including Typhoon Hagupit, the Hyderabad bomb blast, the Sandy Hook shooting, the Nepal earthquake, and the HarDerail derailment. For several catastrophic occurrences, their model achieved $F1$-

Table 1: Some of the potential works for disaster-related informative tweet identification

Authors	Disaster Event	Model	Performance
Madichetty et al. [49]	hurricane Irma, Mexico Earthquake, California Wildfire, etc.	BERT + DenseNet	$F1$-score = 0.66 - 0.88
Malla et al. [50]	COVID-19	Ensemble model	$F1$-score = 0.91
Madichetty et al. [51]	Typhoon Hagupit, Hyderabad bomb blast, Sandy Hook shooting, etc.	RoBERTa + Feature-based method	$F1$-score = 91% - 100 %
Caragea et al. [34]	Haiti disaster	Keywords-based & ML algorithm	$F1$-score = 0.47 - 0.59
Imran et al. [29]	Pakistan earthquake	Artificial Intelligence for Disaster Response (AIDR)	AUC = 0.80
Li et al. [41]	hurricane Sandy & Boston Marathon bombing	Domain adaption approach	AUC = 0.73
Huang and Xiao [43]	hurricane sandy	Logistic regression	$F1$-score = 0.66
Kumar et al. [48]	hurricane Harvey, hurricane Irma, California Wildfire, Sri Lanka Flood	Transfer learning + LSTM	$F1$-score = 0.74 - 0.93
Aipe et al. [45]	California earthquake, Nepal earthquake, India Flood	CNN	$F1$-score = 0.75 - 0.98
Caragea et al. [28]	Philippines foods (2012), Colo- rado foods (2013), etc.	CNN	Accuracy = 75.90–82.52

scores ranging from 91% to 100%. Chy et al. [52] presented a neural network-based method and tested it on the TREC-2018 dataset, yielding an $F1$-score of 0.63. Table 1 lists some of the potential works highlighting the disaster dataset used, the proposed model and their performance in disaster-related informative content identification.

Researchers utilise a variety of deep learning models to categorise informative and uninformative tweets as well as location predictions. The models, however, were trained and evaluated on the same event dataset, which did not demonstrate their resilience. This study bridges the gap by presenting a hybrid CNN model that is trained and evaluated on different event datasets. The models' results validated its resilience by attaining excellent forecast accuracies across a variety of calamities.

3. Methodology

This research aims to build a robust system to classify informative tweets automatically. Three disaster datasets (i) hurricane Harvey, (ii) California Wildfire, and (iii) Kerala Flood are used in this research. The hurricane Harvey dataset consists of a total of 6,378 training, 929 validation, and 1,805 test samples. In the California Wildfire dataset, the number of training, validation and testing samples are 5,163, 752, and 1,461, respectively. The Kerala flood dataset has 5,588 training, 814 validation and 1,582 test samples. The hurricane Harvey and Kerala flood datasets have ten different humanitarian classes. The classes are (i) Caution and Advice (CA), (ii) Displaced people and Evacuations (DPE), (iii) Infrastructure and Utility Damage (IUD), (iv) Injured or Dead People (IDP), (v) Not Humanitarian (NH), (vi) Other Relevant Information (ORI), (vii) Requests or Urgent Needs (RUN), (viii) Rescue Volunteering or Donation Effort (RVDE), (ix) Sympathy and Support (SS), (x) Missing or Found People (MFP). The California wildfire dataset has nine different classes mentioned above except the last class, i.e., Missing or Found People. Detailed data statistics in each sub-category of all three datasets are shown in Table 2.

The proposed model is based on a convolutional neural network that uses both word and character embedding of the disaster-related tweets to classify them into different humanitarian classes. The step-wise working of the proposed model is shown in Figure 1. The performance of the proposed model is

Table 2: Data statistics used to validate the proposed model

Class	hurricane Harvey			California Wildfire			Kerala Flood		
	Training	Dev	Testing	Training	Dev	Testing	Training	Dev	Testing
Caution and Advice (CA)	379	55	107	97	14	28	97	14	28
Displaced people and Evacuations (DPE)	482	70	136	258	38	72	39	6	11
Infrastructure and Utility Damage (IUD)	852	124	241	295	43	84	207	30	59
Injured or Dead People (IDP)	488	71	139	1,362	199	385	254	37	72
Not Humanitarian (NH)	287	42	81	923	134	261	319	47	90
Other Relevant Information (ORI)	1,237	180	350	727	106	205	669	97	189
Requests or Urgent Needs (RUN)	233	34	66	55	8	16	413	60	117
Rescue Volunteering or Donation Effort (RVDE)	1,976	288	559	991	144	280	3,005	438	851
Sympathy and Support (SS)	444	65	126	330	48	94	585	85	165
Missing or Found People (MFP)	-	-	-	125	18	36	-	-	-
Total	6,378	929	1,805	5,163	752	1,461	5,588	814	1,582

compared with other existing deep learning and machine learning models. A total of seven models are implemented: (i) CNN (Word-Character), (ii) CNN (Word), (iii) Random Forest (RF), (iv) K-Nearest Neighbour (KNN), (v) Naive Bayes (NB), (vi) Decision Tree (DT), and (vii) Gradient Boosting (GB). For conventional machine learning models, uni-gram, bi-gram, and tri-gram TF-IDF (Term-Frequency and Inverse-Document-Frequency) features are used. This section discusses the proposed convolutional neural network-based model in detail, along with embedding layers and different convolution processes.

3.1. Convolutional Neural Network

To process the disaster data and extract the contextual information from it automatically, a deep learning-based convolutional neural network (CNN) is used in this research. This section highlights the working of the CNN model with the textual dataset. The CNN mainly consists of three layers: (i) Convolution (ii) Pooling, and (iii) Fully-Connected Dense layer with some other pre-requisites such as padding and embedding. One of the main requirements of the CNN model is the equal length of the input samples. The model does not process the variable size of inputs. Hence, padding is used. There are two types of padding supported by Keras, (i) pre-padding and (ii) post-padding. In pre-padding, the zeros are added at the beginning of the sentence, whereas in post-padding, the zeros are added at the end to equalize the lengths of the input samples.

3.1.1. Embedding Layer

The embedding layer helps to create the embedded matrix for the given input word sequences. For example, if a sentence consists of t words $(W_1, W_2, W_3, ..., W_t)$, then from a pre-trained embedding such as GloVe [53] and FastText, the corresponding word-vector is extracted by one-to-one mapping. For each word W_i, a word-vector having dimension d is extracted. The extracted vectors can be represented as $S(W)_{1:t} = e(W_1), e(W_2), e(W_3), ..., e(W_t)$, where, $(S(W)_{1:t})$ represents the complete sentence and $e(W_1), e(W_2), e(W_3), ..., e(W_t)$ is the individual word's embedding extracted from the pre-trained embed-

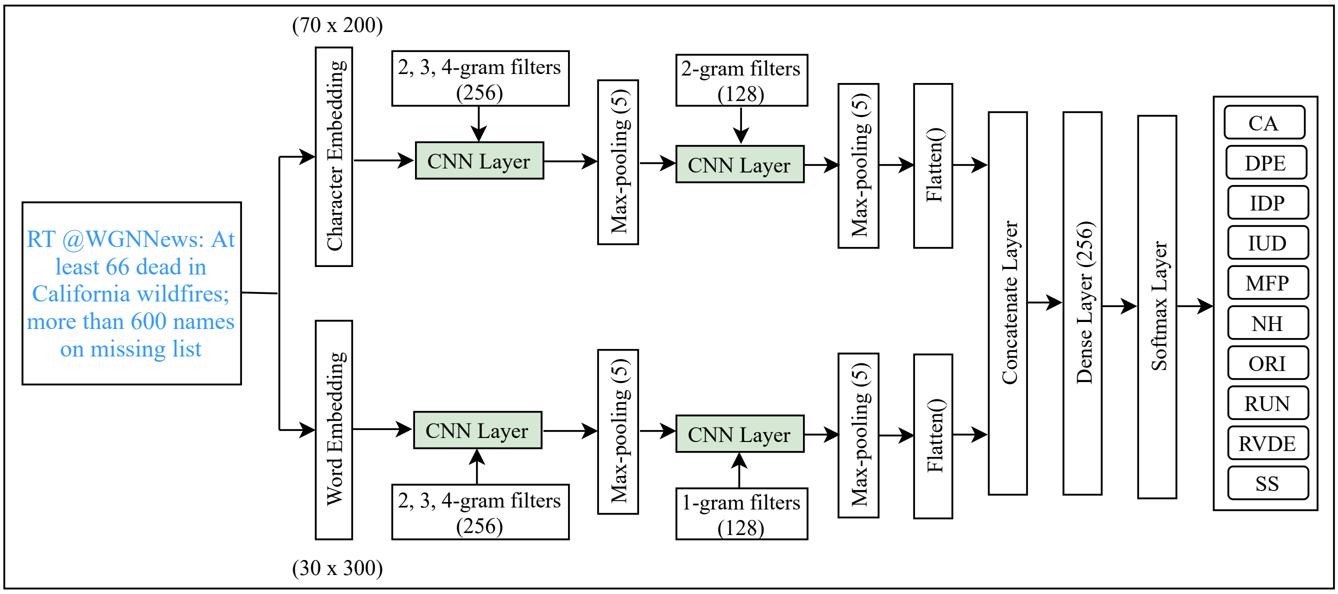

Figure 1: Proposed multi-channel convolutional neural network-based model

ding vector. To form the matrix of the input words, the extracted word-embedding $e(W_1)$, $e(W_2)$, $e(W_3)$,...., $e(W_t)$ are concatenated together.

$$S(W)_{1:t} = e(W_1) \oplus e(W_2) \oplus e(W_3) \oplus ..... \oplus e(W_t) \tag{1}$$

here $\oplus$ is a concatenation operator.

In this way, for every sentence s, a sentence matrix $\mathbf{S} \in \mathbb{R}^{d \times |t|}$ is formed, where, $|t|$ is the total number of words in the sentence and d is the dimension of the word embedding vector. Each word of the sentence is represented with d dimensional word-vector. The formed matrix is shown in equation (2).

$$\mathbf{S} = \begin{bmatrix} W_{11} & W_{21} & W_{31} & \dots & W_{t1} \\ W_{12} & W_{22} & W_{32} & \dots & W_{t2} \\ W_{13} & W_{23} & W_{33} & \dots & W_{t3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ W_{1d} & W_{2d} & W_{3d} & \dots & W_{td} \end{bmatrix} \tag{2}$$

The proposed CNN (Word-Character) model, uses both word embedding and character embedding. For the word embedding, we fixed a maximum word size to 30, and we mapped each word into a 300-dimensional embedding vector using pre-trained FastText[5] embedding vector. Therefore, for each of the tweets, a (30 $\times$ 300) matrix is obtained (see Figure 1). For the character embedding vector, we fixed the

[5]https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.en.300.bin.gz

maximum size of 200 for the characters in each of the tweets. In our dataset, we found 70 different characters, including numbers and special symbols. Therefore, a (70×200) matrix is obtained for each of the tweets. Then these two words and character matrix are used by the convolution operation to extract robust features from the tweets.

3.1.2. Convolution Operation

The convolution layer uses different size of n-grams kernels to extract the hidden contextual features from the input sentence. The n-gram size can vary from uni-gram to quad- or five-gram or even more. For easy understanding, a detailed mathematical explanation of the feature extraction process using a n-gram kernel $\mathbf{F} \in \mathbb{R}^{d\times|n|}$ on sentence matrix $\mathbf{S} \in \mathbb{R}^{d\times|t|}$ can be described as follows:

$$\mathbf{S} = \begin{bmatrix} W_{11} & W_{21} & W_{31} & \dots & W_{t1} \\ W_{12} & W_{22} & W_{32} & \dots & W_{t2} \\ W_{13} & W_{23} & W_{33} & \dots & W_{t3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ W_{1d} & W_{2d} & W_{3d} & \dots & W_{td} \end{bmatrix} \odot \ \mathbf{F} = \begin{bmatrix} F_{11} & F_{21} \\ F_{12} & F_{22} \\ F_{13} & F_{23} \\ \vdots & \vdots \\ F_{1d} & F_{2d} \end{bmatrix}$$

where $\odot$ is the convolution operator.

The sentence matrix S consisting of t words with d dimensional vector for each word, whereas the kernel matrix F consists of $n = 2$ words with dimension d. When the kernel matrix F convolve with the first two words of the sentence matrix S, i.e., W_1 and W_2, it yields a feature value f_1. Next, the kernel F convolve with the next two words, i.e., W_2 and W_3 and produce another feature f_2. Similarly, the kernel convolve with last word pair, i.e., w_{t-1} and w_t to produce f_k feature. The convolution operation between the sentence matrix $\mathbf{S}$ and kernel $\mathbf{F}$ produce the feature matrix having the dimension of *((length of sentence - size of filter) +1)* $\times$ *1*, i.e., $((|t| - F) + 1) \times 1$.

For example, If the total number of words in the sentence is (t=) 20, and the size of the kernel (F) is 2, then a total of 19 ($f_k = ((20 - 2) + 1) = 19$) features are obtained after the convolution operation. In general, the convolution operation between sentence $\mathbf{S} \in \mathbb{R}^{d\times|t|}$ and kernel $\mathbf{F} \in \mathbb{R}^{d\times|n|}$ produces $f_k = ((|t| - \ size \ of \ (F)) + 1)$ features. The extracted features $f_1, f_2, f_3, ..., f_k$ are stored in matrix $\mathbf{C}$.

$$\mathbf{C} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_k \end{bmatrix}$$

where $f_1 = W_{11}F_{11} + W_{12}F_{12} + \dots + W_{1d}F_{1d} + W_{21}F_{21} + W_{22}F_{22} + \dots + W_{2d}F_{2d}$,

$f_2 = W_{21}F_{11} + W_{22}F_{12} + \cdots + W_{2d}F_{1d} + W_{31}F_{21} + W_{32}F_{22} + \cdots + W_{3d}F_{2d}$, and

$f_k = W_{(t-1)1}F_{11} + W_{(t-1)2}F_{12} + \cdots + W_{(t-1)d}F_{1d} + W_{t1}F_{21} + W_{t2}F_{22} + \cdots + W_{td}F_{2d}$

The features obtained using the convolution operation are passed through a non-linear activation function called ReLU. The ReLU activation function is defined by Eq. (3).

$$\sigma(u) = max(0, u) \tag{3}$$

ReLU activation function returns zero for the negative value, whereas, for the positive value, it returns that positive value only. The outcomes are stored in matrix C'. The C' matrix is also having the same dimension as of matrix **C**.

3.1.3. Pooling

The purpose of the pooling layer is to reduce the feature dimensions obtained from the convolution operation. The convolution operations between sentence matrix and kernel yield a large number of features; however, all of them are not important. Hence, the relevant features are pooled out with the help of the pooling layer. The *Keras* library support three types of pooling operations for 1-dimensional convolutional neural network: (i) Max-pooling, (ii) Average Pooling, and (iii) Global Average Pooling[6].

From the features stored in matrix (C'), the max-pooling operation is performed. To pool the features $\hat{p}_i$ from C', a fixed window size k is selected. Max pooling operation pools maximum value from a window size of k. Mathematically, it can be represented as:

$$\hat{p}_i = max(f_1, f_2, ...f_k) \tag{4}$$

Max-pooling provides the features $\hat{p}$=[$\hat{p}_1, \hat{p}_2, p_2, ...\hat{p}_L$], where the length of L is defined by the Eq. (5).

$$L = \lfloor \frac{|C'|}{k} \rfloor \tag{5}$$

where, $|C'|$ is the dimension of the feature vector and k is the size of the max-pooling window. For example, if the total number of features in $|C'| = 22$ and the size of the max-pooling window is $k = 5$, then total four features ($L = \lfloor \frac{22}{5} \rfloor = 4$) will be pooled-out using max-pooling operation.

3.1.4. Fully-Connected Dense Layer

The last layer of the CNN model is the fully connected dense layer. Here, each neuron is connected with the neuron present at the next level. For example, if the number of neurons present at the first dense layer is fifty, and the number of neurons at the second dense layer is twenty. In that case, every neuron

[6]https://keras.io/api/layers/pooling_layers/

Table 3: Best suited hyper-parameters for the proposed model

Model hyper-parameters	CNN (Word-Character)	CNN (Word)
Number of CNN layer	4	4
Number of Dense layer	2	2
Dense layer neurons	256, 9/10	256, 9/10
Number of filters	256, 256, 256, 128	256, 256, 256, 128
Filter size	2-gram, 3-gram, 4-gram, 2-gram	2-gram, 3-gram, 4-gram, 2-gram
Max-pooling window	5	5
Activation function	ReLU, Softmax	ReLU, Softmax
Learning rate	0.001	0.001
Optimizer	Adam	Adam
Loss function	Categorical Crossentropy	Categorical Crossentropy
Batch size	32	32
Epochs	100	100

of the first layer is connected with all twenty neurons of the second layer. Hence, in total, the number of connections between the first and second dense layer will be 50 × 20, i.e., 1,000 + 20 (bias)= 1,020. The features obtained after max-pooling operations, i.e., $\hat{p}$=[$\hat{p}_1, \hat{p}_2, ...\hat{p}_L$] was flattened and passes to the dense layer present at the end of CNN model for further processing.

Activation Function: On output layer, mainly two activation functions (i) sigmoid and (ii) softmax can be used. For binary classification, both activation functions can be used, whereas for multi-class classification problems the softmax activation is preferred. The softmax activation function is defined in Eq. (6).

$$\sigma(w_j) = \frac{e^{w_j}}{\sum_{j=1}^{N} e^{w_j}} \tag{6}$$

where, w_j is the numerical value at the output neuron j, the number of class can vary from 1 to N. The summation of all the probability values for all the classes is equal to 1 $(\sigma(w_1) + \sigma(w_2) + ... + \sigma(w_N) = 1)$. The class that receives the highest probability value will be considered as a predicted class by the model. In general, we can say the predicted class of the input sentence defined by Eq. (7).

$$class\ (q_i) = max\ (\sigma(w)_j) \tag{7}$$

Optimizer: The Keras library supports many optimizers[7] among them; the widely used optimizers are RMSprop, Adam, and SGD. The purpose of the optimizer is to achieve better parameter values in less time. This helps to converge the model quickly. Thi model proposed in the current research uses an Adam optimizer.

[7] https://keras.io/api/optimizers/

Loss function: Another parameter of the CNN model is the loss function. Two loss functions, (i) Binary cross-entropy and (ii) Categorical cross-entropy, are the most preferred choice for the classification tasks. The binary cross-entropy loss function is used for binary classification whereas, categorical cross-entropy is used for multi-class classification problems. The categorical cross-entropy is defined by Eq. (8).

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log p_{ij} \tag{8}$$

Where:

N: is the number of instances.

M: is the number of classes.

y_{ij} is the indicator whether the label j is correct classification or not for instance i.

p_{ij} is the model probability to assigning label j to instance i.

The process of extracting the contextual features from the tweets are explained in this section. The performance of the models with different settings and extracted features are discussed in Section 4.

4. Experimental Results

The proposed convolutional neural network-based model (CNN (Word-Character)) uses both word embedding and character embedding vector of the disaster-related tweets to classify them into different humanitarian classes. The proposed model is validated with the three disaster event categories: hurricane Harvey, the Kerala flood, and the California wildfire. To evaluate the performance of the proposed CNN (Word-Character) model, Equations 9, 10, and 11 are used. The experiment is carried out on the Google Colab platform[8] with their default settings.

- Precision (P): The number of truly predicted informative instances among all retrieved informative instances. Mathematically, it is defined as:

$$Precision\ (P) = \frac{True\ Positive}{True\ Positive\ +\ False\ Positive} \tag{9}$$

- Recall (R): The number of truly predicted informative instance among the total number of true informative instance. Mathematically, it is defined as:

$$Recall\ (R) = \frac{True\ Positive}{True\ Positive\ +\ False\ Negative} \tag{10}$$

[8]https://colab.research.google.com/

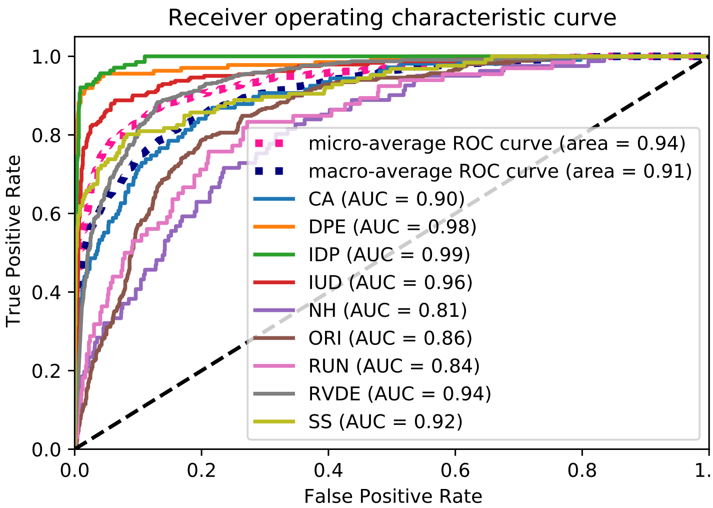

Figure 2: ROC curve for the proposed CNN (Word-Character) model for hurricane Harvey event

- $F1$-score (F1): The harmonic mean of the Precision and Recall is the $F1$-score of the model.

$$F1 - score\ (F1) = 2 * \frac{Precision\ +\ Recall}{Precision\ *\ Recall} \tag{11}$$

- AUC-ROC: It is a curve plotted between the true positive rate (TPR) to false-positive rate (FPR). The value of the area under the curve is closer to 1 represents the best performance of the model.

We extensively performed the experiments by varying the number of CNN layers, learning rate, batch size, epochs, and other parameters. The best-suited hyper-parameters, number of CNN layers, number of neurons in the dense layer, pooling window, etc. are listed in Table 3. Along with the proposed CNN (Word-Character) model, CNN with only word embedding vector (CNN-(Word) and five different machine learning classifiers such as RF, KNN, NB, DT, and GB classifiers are also implemented to compare the performance of the proposed model with them.

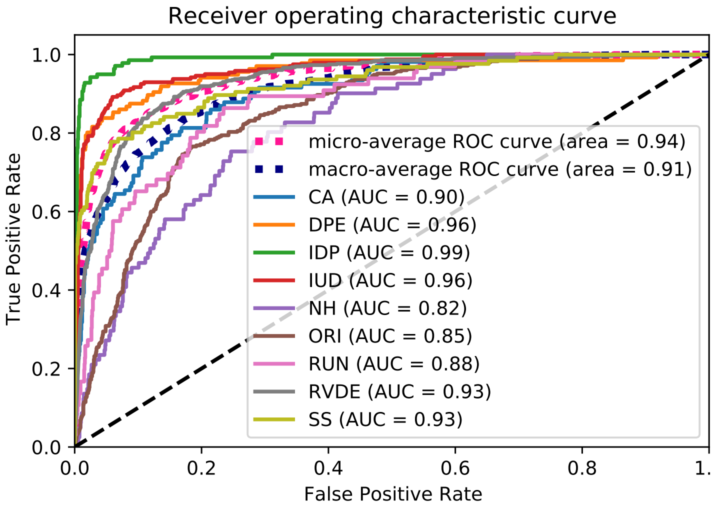

Figure 3: ROC curve for the proposed CNN-word model for hurricane Harvey event

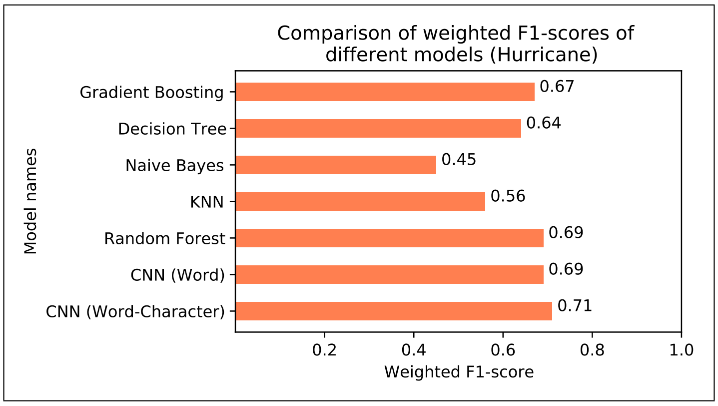

Figure 4: Comparison of weighted $F1$-scores of different models (hurricane Harvey)

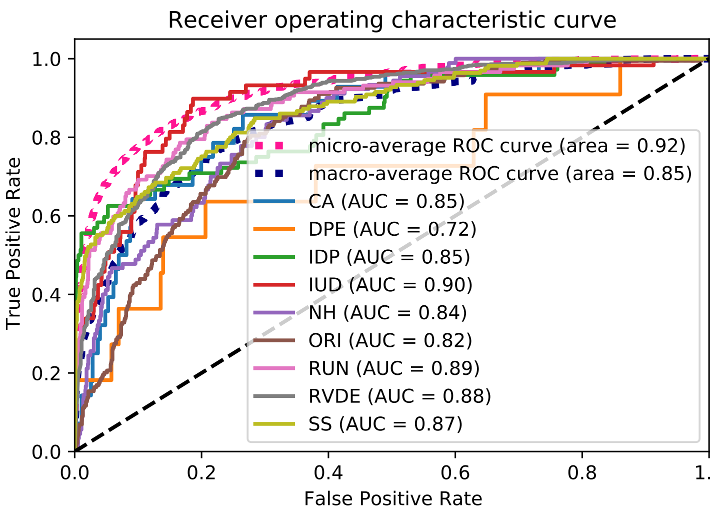

Figure 5: ROC curve for the proposed CNN (Word-Character) model for Kerala flood event

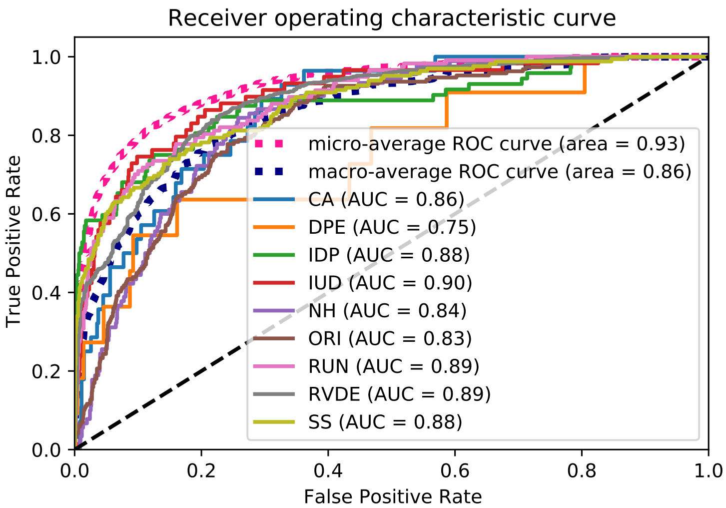

Figure 6: ROC curve for the proposed CNN-word model for Kerala flood event

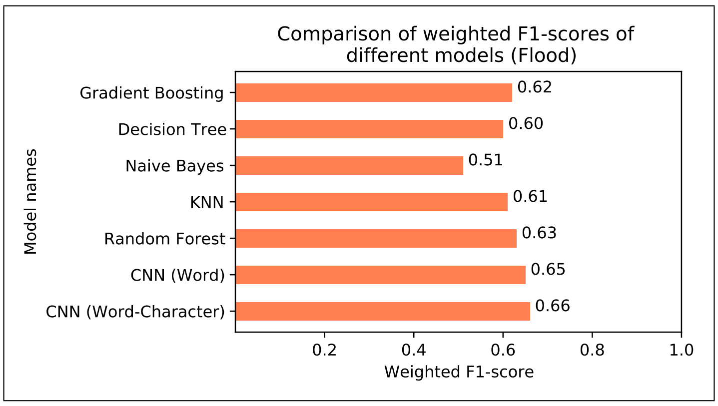

Figure 7: Comparison of weighted $F1$-scores of different models (Kerala flood)

Table 4: Results of the different deep learning and machine learning models for hurricane Harvey event

Class	CNN (Word-Character)			CNN (Word)			Random Forest			KNN			Naive Bayes			Decision Tree			Gradient Boosting		
	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$
CA	0.64	0.44	0.52	0.59	0.50	0.55	0.81	0.36	0.49	0.45	0.47	0.46	0.40	0.36	0.38	0.48	0.44	0.46	0.58	0.42	0.49
DPE	0.91	0.88	0.90	0.81	0.76	0.78	0.90	0.85	0.88	0.53	0.68	0.60	0.48	0.44	0.46	0.85	0.85	0.85	0.85	0.82	0.84
IDP	0.88	0.92	0.90	0.84	0.91	0.87	0.84	0.86	0.85	0.68	0.71	0.70	0.55	0.58	0.56	0.85	0.87	0.86	0.83	0.82	0.83
IUD	0.79	0.83	0.81	0.81	0.81	0.81	0.90	0.81	0.85	0.60	0.63	0.62	0.47	0.56	0.51	0.77	0.72	0.74	0.82	0.78	0.80
NH	0.33	0.20	0.25	0.29	0.12	0.17	0.50	0.01	0.02	0.27	0.15	0.19	0.15	0.17	0.16	0.20	0.17	0.19	0.26	0.10	0.14
ORI	0.55	0.62	0.58	0.54	0.60	0.57	0.53	0.74	0.62	0.41	0.42	0.41	0.23	0.19	0.21	0.49	0.52	0.51	0.52	0.62	0.57
RUN	0.33	0.17	0.22	0.33	0.21	0.26	0.43	0.15	0.22	0.26	0.18	0.21	0.13	0.14	0.13	0.35	0.26	0.30	0.36	0.21	0.27
RVDE	0.76	0.85	0.80	0.77	0.85	0.80	0.71	0.87	0.78	0.67	0.71	0.69	0.64	0.64	0.64	0.72	0.77	0.74	0.73	0.83	0.78
SS	0.85	0.60	0.71	0.72	0.62	0.66	0.92	0.52	0.66	0.78	0.52	0.62	0.46	0.52	0.49	0.61	0.57	0.59	0.72	0.56	0.63
Weighted Avg.	**0.71**	**0.72**	**0.71**	0.69	0.70	0.69	0.73	0.72	0.69	0.56	0.57	0.56	0.45	0.46	0.45	0.64	0.65	0.64	0.67	0.68	0.67

Table 5: Results of various models when it is trained with Kerala flood dataset

Class	CNN (Word-Character)			CNN (Word)			Random Forest			KNN			Naive Bayes			Decision Tree			Gradient Boosting		
	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$
CA	0.17	0.14	0.15	0.00	0.00	0.00	1.00	0.07	0.13	0.27	0.21	0.24	0.08	0.04	0.05	0.18	0.14	0.16	0.22	0.14	0.17
DPE	0.25	0.18	0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.09	0.12	0.00	0.00	0.00	0.11	0.09	0.10	0.29	0.18	0.22
IDP	0.71	0.54	0.61	0.69	0.51	0.59	0.82	0.62	0.71	0.54	0.60	0.57	0.30	0.29	0.30	0.60	0.58	0.59	0.78	0.68	0.73
IUD	0.37	0.37	0.37	0.39	0.41	0.40	0.68	0.22	0.33	0.39	0.34	0.36	0.27	0.19	0.22	0.49	0.32	0.39	0.43	0.27	0.33
NH	0.39	0.26	0.31	0.24	0.22	0.23	0.00	0.00	0.00	0.30	0.17	0.21	0.22	0.19	0.20	0.17	0.16	0.16	0.18	0.10	0.13
ORI	0.39	0.41	0.40	0.44	0.36	0.40	0.50	0.42	0.46	0.36	0.31	0.34	0.32	0.29	0.30	0.31	0.32	0.32	0.40	0.37	0.38
RUN	0.58	0.52	0.55	0.53	0.54	0.53	0.68	0.40	0.51	0.46	0.44	0.45	0.36	0.42	0.39	0.47	0.41	0.44	0.55	0.32	0.40
RVDE	0.80	0.86	0.83	0.80	0.87	0.83	0.69	0.97	0.81	0.73	0.85	0.79	0.68	0.70	0.69	0.76	0.79	0.78	0.72	0.89	0.80
SS	0.59	0.55	0.57	0.57	0.61	0.59	0.88	0.44	0.58	0.66	0.46	0.54	0.37	0.44	0.40	0.54	0.55	0.54	0.66	0.50	0.57
Weighted Avg.	**0.65**	**0.66**	**0.66**	0.64	0.66	0.65	0.65	0.68	0.63	0.60	0.63	0.61	0.51	0.52	0.51	0.59	0.60	0.60	0.62	0.65	0.62

Table 6: Results of various models when it is trained with California wildfire dataset

Class	CNN (Word-Character)			CNN (Word)			Random Forest			KNN			Naive Bayes			Decision Tree			Gradient Boosting		
	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$
CA	0.44	0.39	0.42	0.40	0.14	0.21	0.50	0.04	0.07	0.28	0.25	0.26	0.21	0.14	0.17	0.26	0.29	0.27	0.24	0.14	0.18
DPE	0.68	0.57	0.62	0.56	0.49	0.52	0.81	0.54	0.65	0.57	0.56	0.56	0.28	0.25	0.26	0.61	0.60	0.61	0.69	0.51	0.59
IDP	0.88	0.86	0.87	0.81	0.86	0.84	0.83	0.94	0.88	0.73	0.85	0.78	0.71	0.75	0.73	0.86	0.85	0.85	0.86	0.89	0.87
IUD	0.56	0.27	0.37	0.42	0.50	0.46	0.56	0.52	0.54	0.46	0.40	0.43	0.30	0.26	0.28	0.44	0.51	0.48	0.57	0.51	0.54
MFP	0.83	0.42	0.56	0.52	0.31	0.39	0.88	0.42	0.57	0.56	0.39	0.46	0.19	0.22	0.21	0.61	0.47	0.53	0.74	0.47	0.58
NH	0.55	0.64	0.59	0.56	0.63	0.59	0.53	0.72	0.61	0.49	0.57	0.53	0.53	0.64	0.58	0.47	0.49	0.48	0.47	0.50	0.48
ORI	0.39	0.52	0.45	0.40	0.41	0.40	0.41	0.29	0.34	0.34	0.27	0.30	0.33	0.23	0.27	0.33	0.35	0.34	0.32	0.37	0.34
RUN	0.12	0.06	0.08	0.25	0.06	0.10	0.50	0.12	0.20	0.21	0.19	0.20	0.17	0.06	0.09	0.19	0.19	0.19	0.38	0.31	0.34
RVDE	0.76	0.80	0.78	0.82	0.77	0.80	0.69	0.83	0.75	0.72	0.73	0.73	0.68	0.69	0.68	0.69	0.69	0.69	0.69	0.77	0.73
SS	0.68	0.44	0.53	0.64	0.61	0.62	0.89	0.35	0.50	0.65	0.33	0.44	0.33	0.37	0.35	0.39	0.33	0.36	0.75	0.43	0.54
Weighted Avg.	**0.67**	**0.66**	**0.66**	0.64	0.65	0.64	0.67	0.67	0.65	0.58	0.59	0.58	0.52	0.54	0.53	0.59	0.59	0.59	0.63	0.62	0.62

The results of different deep learning and machine learning models for the hurricane Harvey event are listed in Table 4. The proposed CNN (Word-Character) model performed best among all the implemented models and achieved a weighted precision of 0.71, recall of 0.72, and $F1$-score of 0.71. The CNN (Word) model achieved a weighted precision of 0.69, recall of 0.70, and $F1$-score of 0.69, whereas, among the machine learning models, random forest performed best with the weighted precision of 0.73, recall of 0.72, and $F1$-score of 0.69. It means for the hurricane event combination of character embedding with the word embedding performed best as can be seen in Table 4. The ROC curve for the proposed CNN (Word-Character) and CNN (word) models can be seen in Figure 2, and Figure 3, respectively. The performance of all the implemented models in terms of weighted $F1$-score are plotted in Figure 4.

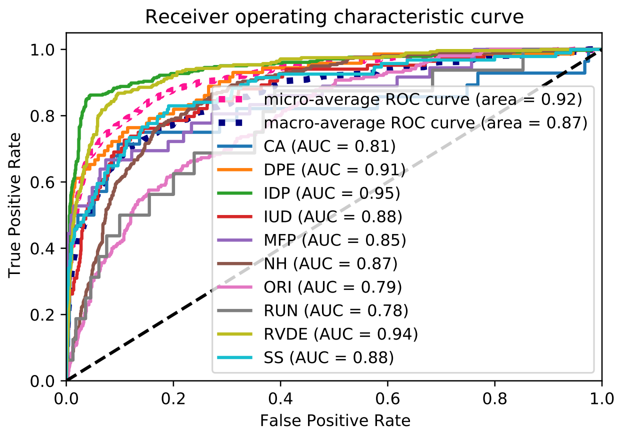

Figure 8: ROC curve for the proposed CNN (Word-Character) model for California wildfire event

The result of all the implemented models for the Kerala flood event is listed in Table 5. The proposed CNN (Word-Character) model again performed best among all the implemented deep learning and machine learning models. The proposed CNN (Word-Character) model achieved a weighted precision of 0.65, recall of 0.66, and $F1$-score of 0.66 whereas the CNN (Word) model achieved a weighted precision of 0.64, recall of 0.66, and $F1$-score of 0.65. Among all the machine learning models, the random forest classifier performed better with the weighted precision of 0.65, recall of 0.68, and $F1$-score of 0.63. The ROC curve for the CNN (Word-Character) and CNN (Word) models can be seen in Figure 5, and Figure 6, respectively. The performance comparison of weighted $F1$-score of all the implemented models can be seen in Figure 7.

The result of different models for the California wildfire event is listed in Table 6. The proposed CNN (Word-Character) achieved a weighted precision of 0.67, recall of 0.66, and $F1$-score of 0.66 whereas CNN (Word) achieved a weighted precision of 0.64, recall of 0.65, and $F1$-score of 0.64. Among all the other implemented machine learning classifiers, random forest is again performed best with the weighted precision of 0.67, recall of 0.67, and $F1$-score of 0.65. The ROC curve for CNN (Word-Character) and CNN (Word)

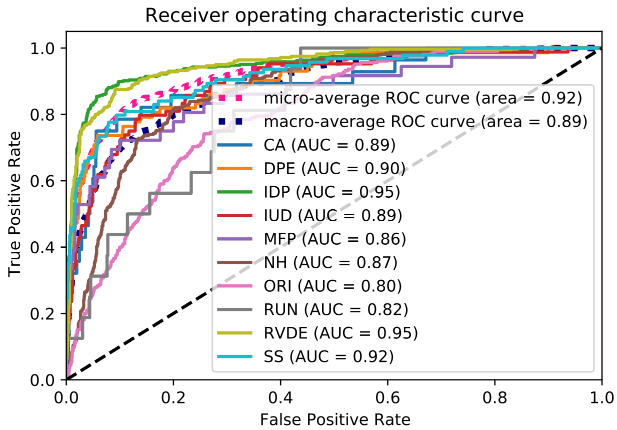

Figure 9: ROC curve for the proposed CNN-Word model for California wildfire event

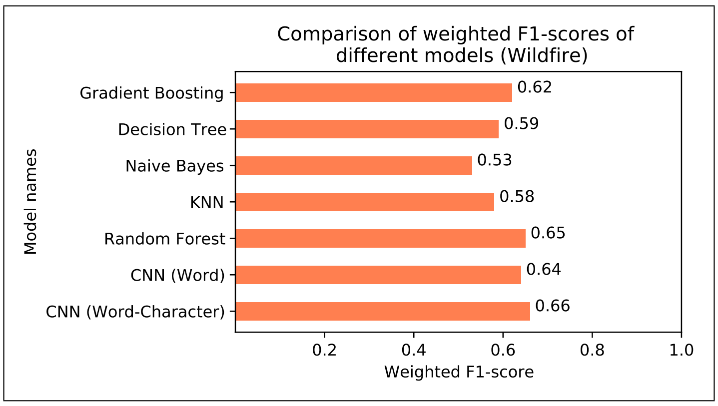

Figure 10: Comparison of weighted $F1$-scores of different models (California Wildfire)

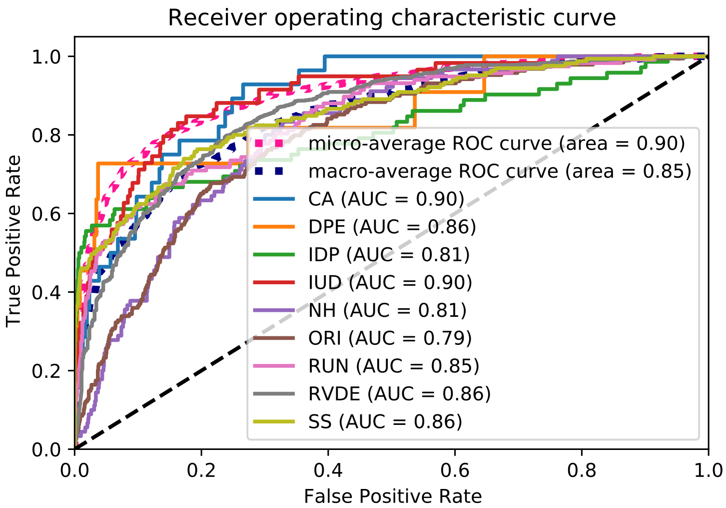

Figure 11: ROC curve obtained by the model which was trained with hurricane Harvey dataset and tested with Kerala flood dataset

Table 7: Results obtained by the model which was trained with hurricane Harvey dataset and tested with Kerala flood dataset

Class	**P**	**R**	$F1$
CA	0.43	0.21	0.29
DPE	0.21	0.45	0.29
IDP	0.62	0.51	0.56
IUD	0.35	0.51	0.41
NH	0.20	0.19	0.20
ORI	0.32	0.47	0.38
RUN	0.51	0.50	0.51
RVDE	0.80	0.78	0.79
SS	0.85	0.46	0.60
Weighted Avg	**0.66**	**0.62**	**0.63**

model can be seen in Figure 8 and Figure 9, respectively. The performance of all the implemented models in terms of weighted $F1$-score can be seen in Figure 10.

To check the robustness of the model, we have trained and tested it with cross event dataset. The hurricane dataset was used to train the model, whereas the trained model is tested with the Kerala Flood dataset. The results of this experiment are presented in Table 7. The weighted average precision, recall and $F1$-score are 0.66, 0.62 and 0.63, respectively, which is very close to the performance values of the model when it was trained and tested with the same dataset. These results indicate the robustness of the proposed hybrid model, as it can predict other events with similar accuracies. The AUC-ROC plot obtained with different trained-test datasets is shown in Figure 11. The micro average ROC value of the model trained and tested with different datasets is 0.85, which is similar to the performance of the model trained and tested with the same dataset (Figure 5).

5. Discussion

The main conclusion of this study is that CNN with a fusion of character and word embeddings outperforms CNN with simply a word embedding model in detecting informative tweets. The suggested CNN (word-char) model was trained and evaluated on three datasets, and it performed better on all three (Tables 4, 5, and 6. Traditional machine learning classifiers built with features derived using tf-idf approaches have poorer prediction accuracy than deep neural models with features generated automatically using various convolution procedures. As a result, automated feature extraction approaches are more suited for predicting disaster-related tweets, according to another conclusion of this study. On hurricane Harvey, Kerala flood, and California wildfire datasets, the CNN model with the fusion of word and char embedding obtained 2%, 1%, and 2% higher $F1$-score value than the CNN model with only word embedding technique, as shown in Tables 4, 5, and 6, respectively. Another finding of the research is that the developed model is robust as predicts the different events with similar performance values. The performance of the model in cross-domain setting and same domain setting is shown in Figure 5 and 11. The reason for the robustness of the model is better feature extraction due to word and character embeddings.

The present study intends to use social media posts to gather information about needy individuals during catastrophes. When a tragedy strikes, many people seek assistance; however, owing to a lack of communication, the request is not received by the appropriate authorities, and the victims confront several difficulties. The suggested model is useful in this circumstance since it can extract important information. The proposed model's performance on multiple datasets revealed that it could filter informative disaster-related tweets with high accuracy. As a result, during an emergency, it may be used as an initial filter to collect disaster-related tweets. To offer immediate assistance to victims while minimising harm. To evaluate the model's resilience, we trained it on the hurricane Harvey dataset and tested it on the Kerala flood dataset. The results produced by this model show that the model predicts the event with equal accuracy when the micro average ROC value of both models is the same (Figure 5 and 11).

This study employs both traditional machine learning models and a sophisticated deep learning framework. The traditional ML model got input from a term frequency-inverse document frequency (tf-idf) vectorizer, whereas deep learning received embedded input from a pre-trained embedding. The results of the various models with varied settings revealed that the proposed multi-channel CNN model outperforms existing ML classifiers. The findings of several models also suggest that the tf-idf vectorizer is ineffective for this task. The tf-idf vectorizer fails to capture the semantics of such messages, but the pre-trained embedding captures the sentence meanings successfully. Every second, thousands of tweets are sent out on Twitter, with just a small percentage of them falling into one of these categories. As a result, the suggested automated method may aid in the automatic extraction of such useful tweets from the tweets.

Entities responsible for delivering aid during the catastrophe must receive timely information. However,

Table 8: Snapshot of the actual and predicted classes of disaster-related tweet test sample using proposed model

Event	Tweet	Correct label	Predicted label
hurricane Harvey	After #Harvey, we need to re-think the messaging/communications for #flood victims caught in #tornado warnings. #TXflood #LAflood	Caution and Advice	Injured or Dead People
	RT @HuffPost: Harvey spawns tornadoes that devastate homes outside Houston	Infrastructure and Utility Damage	Caution and Advice
	Houston continues to deal with record flooding in the aftermath of hurricane Harvey.	Not Humanitarian	Other Relevant Information
California Wildfire	#BREAKING: Eight additional bodies were found and at least 1,000 people are missing.	Injured or Dead People	Missing or Found People
	California wildfire death toll rises, Arizona firefighters to assist efforts	Injured or Dead People	Other Relevant Information
	Several killed in California wildfires @CNN	Injured or Dead People	Other Relevant Information
Kerala Flood	Some looted the distillery during floods. Imagine what happens afterwards.	Other Relevant Information	Infrastructure and Utility Damage
	Jaipur girl uses insta stories to help Kerala flood victims	Rescue Volunteering or Donation Effort	Other Relevant Information
	Please help them. #SBI #KeralaFloodRelief #CMDRF #Donate4Kerala	Requests or Urgent Needs	Rescue Volunteering or Donation Effort

due to the huge volume of tweets sent every second, human screening of disaster-related tweets is nearly impossible. The suggested multi-channel system has both character and word characteristics capable of accurately collecting meaningful tweets. Once the informative tweets have been isolated from the rest of the tweets, they may be readily classified into several categories of catastrophe information. During a crisis, for example, individuals may seek medical assistance, look for someone who can give water and food, locate missing persons, and do other such activities. Once these categories are determined, the appropriate team will be notified, and the victim will receive assistance. One of the reasons behind the miss-classification of the tweets is the short form of the text. People have developed a practice of typing messages in abbreviated forms on social media. For example, the word 'help' can be written as hlp', Helpp', hlpme'; the word 'before' can be typed as b4', be4', and so on; recognising the context of such phrases by the model is challenging. Although the combination of character-level and word-level characteristics worked better in this situation (as shown in Tables 4, 5, and 6). However, if a user tweets for assistance, he or she must use a valid grammatical term so that the system can recognise it automatically. Otherwise, such communications may be classed as non-informative.

Table 8 contains several examples of where the proposed model fails to identify the right classes. The tweet *After #Harvey, we need to rethink the messaging/communications for #flood victims trapped in #tornado warnings. #TXflood #LAflood"* belongs to the *Caution and Advice class*, however, the suggested model predicted it in the *Injured or Dead People* class. The term "*victims caught*" in the posted tweets might be the cause of the misclassification, given the word victims is commonly used in the *Injured or Dead People* class. Similarly, the *#BREAKING: Eight further bodies have been discovered, and at least 1,000 people are missing.*" belongs to the *Injured or Dead People* class, yet the suggested model predicted it as *Missing or Found People*. The usage of the terms *found* and *missing* in the post might be one of the causes. As a result, classifying these postings into the right groups becomes even more challenging if these types of similar terms appear throughout the different classes.

The proposed approach is sustainable as it does not require a huge computation facility as it utilizes the dataset of existing social media. The recommended approach may be incorporated with any social media platform to discover disaster-related relevant posts. An Android app may be created by combining the recommended approach for analysing the live stream of social media posts in order to help individuals become more situationally aware of the crisis. If domain-specific training is conducted, the suggested approach can also be used in comparable occurrences such as traffic accidents and civil unrest. The suggested approach has a restriction in that it only examines tweets in English; however, during an emergency, people may tweet in regional languages as well. As a result, in the future, a deep neural network-based model for dealing with multilingual issues may be developed.

6. Conclusion

In this study, we used a multi-channel convolutional neural network to create a robust deep learning framework. To detect informative tweets, traditional machine learning classifiers such as RF, KNN, NB, DT, and GB classifier are employed. However, the experimental results revealed that the traditional ML-based classifier misclassified a large number of tweets. The proposed deep hybrid model, on the other hand, has a reasonable prediction accuracy. The suggested hybrid model received a $F1$-score of 0.71 for hurricane Harvey and 0.66 for both the Kerala flood and California wildfire datasets. To test the model's resilience, it is trained on the hurricane Harvey dataset and tested on the Kerala flood dataset, yielding a $F1$-score value of 0.63, which is close to the value obtained by the model trained and tested on the same dataset.

Future studies can increase the identification rate by incorporating more information accessible with tweets, such as the number of times retweeted, the number of short words, the URL, and others. This study solely utilised tweets in English. Including different languages may aid in obtaining more accurate forecasts. In the future, a cross-domain framework capable of effectively capturing any disaster-related tweets can be built.

References

[1] J. Osorio-Arjona, J. Horak, R. Svoboda, Y. García-Ruíz, Social media semantic perceptions on Madrid metro system: Using Twitter data to link complaints to space, Sustainable Cities and Society 64 (2021) 102530.

[2] M. Hosseini, T. El-Diraby, A. Shalaby, Supporting sustainable system adoption: Socio-semantic analysis of transit rider debates on social media, Sustainable cities and society 38 (2018) 123–136.

[3] M. F. DiCarlo, E. Z. Berglund, Connected communities improve hazard response: An agent-based model of social media behaviors during hurricanes, Sustainable Cities and Society 69 (2021) 102836.

[4] A. Kumar, J. P. Singh, N. P. Rana, Authenticity of geo-location and place name in tweets, in: Proceedings of the 23rd americas conference on information systems, 2017, pp. 1–9.

[5] A. Kumar, J. P. Singh, Location reference identification from tweets during emergencies: A deep learning approach, International Journal of Disaster Risk Reduction 33 (2019) 365–375.

[6] D. T. Nguyen, F. Ofli, M. Imran, P. Mitra, Damage assessment from social media imagery data during disasters, in: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, ACM, 2017, pp. 569–576.

[7] K. K. Kapoor, K. Tamilmani, N. P. Rana, P. Patil, Y. K. Dwivedi, S. Nerur, Advances in social media research: Past, present and future, Information Systems Frontiers 20 (3) (2018) 531–558.

[8] J. Kim, M. Hastak, Social network analysis: Characteristics of online social networks after a disaster, International Journal of Information Management 38 (1) (2018) 86–96.

[9] J. R. Ragini, P. R. Anand, V. Bhaskar, Big data analytics for disaster response and recovery through sentiment analysis, International Journal of Information Management 42 (2018) 13–24.

[10] J. Son, H. K. Lee, S. Jin, J. Lee, Content features of tweets for effective communication during disasters: A media synchronicity theory perspective, International Journal of Information Management 45 (2019) 56–68.

[11] D. T. Nguyen, F. Alam, F. Ofli, M. Imran, Automatic image filtering on social networks using deep learning and perceptual hashing during crises, arXiv preprint arXiv:1704.02602 (2017).

[12] D. Nguyen, K. A. A. Mannai, S. Joty, H. Sajjad, M. Imran, P. Mitra, Robust classification of crisis-related data on social networks using convolutional neural networks, in: International AAAI Conference on Web and Social Media, 2017, pp. 632–635.

[13] B. Gordan, D. J. Armaghani, M. Hajihassani, M. Monjezi, Prediction of seismic slope stability through combination of particle swarm optimization and neural network, Engineering with Computers 32 (1) (2016) 85–97.

[14] X. Song, Q. Zhang, Y. Sekimoto, T. Horanont, S. Ueyama, R. Shibasaki, Intelligent system for human behavior analysis and reasoning following large-scale disasters, IEEE Intelligent Systems 28 (4) (2013) 35–42.

[15] X. Song, Q. Zhang, Y. Sekimoto, R. Shibasaki, N. J. Yuan, X. Xie, A simulator of human emergency mobility following disasters: Knowledge transfer from big disaster data, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 29(1), 2015, pp. 730–736.

[16] P. K. Roy, J. P. Singh, S. Banerjee, Deep learning to filter SMS spam, Future Generation Computer Systems 102 (2020) 524–533.

[17] P. K. Roy, J. P. Singh, Predicting closed questions on community question answering sites using convolutional neural network, Neural Computing and Applications 32 (2020) 10555–10572.

[18] P. K. Roy, Multilayer convolutional neural network to filter low quality content from quora, Neural Processing Letters 52 (1) (2020) 805–821.

[19] J. Wang, L.-C. Yu, K. R. Lai, X. Zhang, Dimensional sentiment analysis using a regional CNN-LSTM model, in: Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers), 2016, pp. 225–230.

[20] S. Saumya, J. P. Singh, Y. K. Dwivedi, Predicting the helpfulness score of online reviews using convolutional neural network, Soft Computing (2019) 1–17.

[21] P. K. Roy, A. K. Tripathy, T. K. Das, X.-Z. Gao, A framework for hate speech detection using deep convolutional neural

network, IEEE Access 8 (2020) 204951–204962.

[22] T. El-Diraby, A. Shalaby, M. Hosseini, Linking social, semantic and sentiment analyses to support modeling transit customers' satisfaction: Towards formal study of opinion dynamics, Sustainable Cities and Society 49 (2019) 101578.

[23] J. Zeng, Y. Zhang, X. Ma, Fake news detection for epidemic emergencies via deep correlations between text and images, Sustainable Cities and Society 66 (2021) 102652.

[24] F. Yao, Y. Wang, Towards resilient and smart cities: A real-time urban analytical and geo-visual system for social media streaming data, Sustainable Cities and Society 63 (2020) 102448.

[25] M. Imran, C. Castillo, Towards a data-driven approach to identify crisis-related topics in social media streams, in: Proceedings of the 24th International Conference on World Wide Web, ACM, 2015, pp. 1205–1210.

[26] F. Atefeh, W. Khreich, A survey of techniques for event detection in Twitter, Computational Intelligence 31 (1) (2015) 132–164.

[27] X. Zheng, J. Han, A. Sun, A survey of location prediction on Twitter, IEEE Transactions on Knowledge and Data Engineering 30 (9) (2018) 1652–1671.

[28] C. Caragea, A. Silvescu, A. H. Tapia, Identifying informative messages in disaster events using convolutional neural networks, in: International Conference on Information Systems for Crisis Response and Management, 2016, pp. 137–147.

[29] M. Imran, C. Castillo, J. Lucas, P. Meier, S. Vieweg, Aidr: Artificial intelligence for disaster response, in: Proceedings of the 23rd International Conference on World Wide Web, ACM, 2014, pp. 159–162.

[30] M. Yu, Q. Huang, H. Qin, C. Scheele, C. Yang, Deep learning for real-time social media text classification for situation awareness–using hurricanes Sandy, Harvey, and Irma as case studies, International Journal of Digital Earth 12 (11) (2019) 1230–1247.

[31] Z. Ashktorab, C. Brown, M. Nandi, A. Culotta, Tweedr: Mining Twitter to inform disaster response., in: Proceedings of the 11th ISCRAM Conference, 2014, pp. 354–358.

[32] F. Alam, M. Imran, F. Ofli, Image4act: Online social media image processing for disaster response, in: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, ACM, 2017, pp. 601–604.

[33] S. Daly, J. A. Thom, Mining and classifying image posts on social media to analyse fires, in: ISCRAM 2016 Conference, 2016, pp. 1–14.

[34] C. Caragea, N. Mcneese, A. Jaiswal, G. Traylor, H. woo Kim, P. Mitra, D. Wu, A. H. Tapia, L. Giles, B. J. Jansen, J. Yen, Classifying text messages for the Haiti earthquake, in: Proceedings of the 8th International Conference on Information Systems for Crisis Response and Management (ISCRAM2011), 2011, pp. 1–10.

[35] S. Verma, S. Vieweg, W. Corvey, L. Palen, J. Martin, M. Palmer, A. Schram, K. Anderson, Natural language processing to the rescue? extracting" situational awareness" tweets during mass emergency, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 5(1), 2011, pp. 385–392.

[36] M. A. Cameron, R. Power, B. Robinson, J. Yin, Emergency situation awareness from Twitter for crisis management, in: Proceedings of the 21st International Conference on World Wide Web, ACM, 2012, pp. 695–698.

[37] M. Imran, C. Castillo, F. Diaz, S. Vieweg, Processing social media messages in mass emergency: A survey, ACM Computing Surveys (CSUR) 47 (4) (2015) 1–38.

[38] M. Imran, S. Elbassuoni, C. Castillo, F. Diaz, P. Meier, Extracting information nuggets from disaster-related messages in social media, in: Proceedings of the 10th ISCRAM Conference, 2013, pp. 1–10.

[39] M. Imran, S. Elbassuoni, C. Castillo, F. Diaz, P. Meier, Practical extraction of disaster-relevant information from social media, in: Proceedings of the 22nd International Conference on World Wide Web, ACM, 2013, pp. 1021–1024.

[40] A. Olteanu, C. Castillo, F. Diaz, S. Vieweg, Crisislex: A lexicon for collecting and filtering microblogged communications in crises, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 8(1), 2014, pp. 376–385.

[41] H. Li, N. Guevara, N. Herndon, D. Caragea, K. Neppalli, C. Caragea, A. C. Squicciarini, A. H. Tapia, Twitter mining for disaster response: A domain adaptation approach, in: 12th ISCRAM Conference, 2015, pp. 1–7.

[42] K. Rudra, S. Banerjee, N. Ganguly, P. Goyal, M. Imran, P. Mitra, Summarizing situational tweets in crisis scenario, in: Proceedings of the 27th ACM Conference on Hypertext and Social Media, ACM, 2016, pp. 137–147.

[43] Q. Huang, Y. Xiao, Geographic situational awareness: mining tweets for disaster preparedness, emergency response, impact, and recovery, ISPRS International Journal of Geo-Information 4 (3) (2015) 1549–1568.

[44] D. Tien Nguyen, S. Joty, M. Imran, H. Sajjad, P. Mitra, Applications of online deep learning for crisis response using social media information, arXiv e-prints (2016) arXiv–1610.

[45] A. Aipe, A. Ekbal, N. Mukuntha, , S. Kurohashi, Linguistic feature assisted deep learning approach towards multi-label classification of crisis related tweets, in: Proceedings of the 15th ISCRAM Conference, 2018, pp. 705–717.

[46] D. Graf, W. Retschitzegger, W. Schwinger, B. Pröll, E. Kapsammer, Cross-domain informativeness classification for disaster situations, in: Proceedings of the 10th International Conference on Management of Digital EcoSystems, ACM, 2018, pp. 183–190.

[47] J. P. Singh, Y. K. Dwivedi, N. P. Rana, A. Kumar, K. K. Kapoor, Event classification and location prediction from tweets during disasters, Annals of Operations Research 283 (1) (2019) 737–757.

[48] A. Kumar, J. P. Singh, Y. K. Dwivedi, N. P. Rana, A deep multi-modal neural network for informative Twitter content classification during emergencies, Annals of Operations Research (2020) 1–32.

[49] S. Madichetty, S. Muthukumarasamy, P. Jayadev, Multi-modal classification of twitter data during disasters for humanitarian response, Journal of Ambient Intelligence and Humanized Computing (2021) 1–15.

[50] S. Malla, P. Alphonse, Covid-19 outbreak: An ensemble pre-trained deep learning model for detecting informative tweets, Applied Soft Computing 107 (2021) 107495.

[51] S. Madichetty, S. M, A neural-based approach for detecting the situational information from twitter during disaster, IEEE Transactions on Computational Social Systems 8 (4) (2021) 870–880. doi:10.1109/TCSS.2021.3064299.

[52] A. N. Chy, U. A. Siddiqua, M. Aono, Exploiting transfer learning and hand-crafted features in a unified neural model for identifying actionable informative tweets, Journal of Information Processing 29 (2021) 16–29.

[53] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.