

# Multi-channel convolutional neural network for the identification of eyewitness tweets of disaster

Abhinav Kumar · Jyoti Prakash Singh · Nripendra  
P. Rana · Yogesh K. Dwivedi

Received: date / Accepted: date

**Abstract** During a disaster, a large number of disaster-related social media posts are widely disseminated. Only a small percentage of disaster-related information is posted by eyewitnesses. The post of a disaster eyewitness offers an accurate depiction of the disaster. Therefore, the information posted by the eyewitness is preferred over the other source of information as it is more effective at helping organize rescue and relief operations and potentially saving lives. In this work, we propose a multi-channel convolutional neural network (MCNN) that uses three different word-embedding vectors together to classify disaster-related tweets into eyewitness, non-eyewitness, and don't know classes. We compared the performance of the proposed multi-channel convolutional neural network with sev-

---

Abhinav Kumar<sup>1,2</sup>

<sup>1</sup>Department of Computer Science and Engineering, Siksha 'O' Anusanshan (Deemed to be University), Bhubaneswar, India,

<sup>2</sup>National Institute of Technology Patna, India

E-mail: abhinavanand05@gmail.com

Jyoti Prakash Singh

Department of Computer Science and Engineering, National Institute of Technology Patna, India

E-mail: jps@nitp.ac.in

Nripendra P. Rana

College of Business and Economics, Qatar University, Doha, Qatar

E-mail: nrananp@gmail.com

Yogesh K. Dwivedi<sup>3,4</sup>

<sup>3</sup>Emerging Markets Research Centre (EMaRC), School of Management, Room #323, Swansea University, Bay Campus, Fabian Bay, Swansea, SA1 8EN, Wales, UK

<sup>4</sup>Department of Management, Symbiosis Institute of Business Management, Pune & Symbiosis International (Deemed University), Pune, Maharashtra, India

E-mail: y.k.dwivedi@swansea.ac.uk

eral attention-based deep-learning models and conventional machine learning-models such as recurrent neural network, gated recurrent unit, Long-Short-Term-Memory, convolutional neural network, logistic regression, support vector machine, and gradient boosting. The proposed multi-channel convolutional neural network achieved an  $F_1$ -score of 0.84, 0.88, 0.84, and 0.86 with four disaster-related datasets of floods, earthquakes, hurricanes, and wildfires, respectively. The experimental results show that the training MCNN model with different word embedding together performs better than the conventional machine-learning models and several other deep-learning models.

**Keywords** Disaster · Eyewitness tweets · Informative contents · Multi-channel convolutional neural network · Recurrent neural network

## 1 Introduction

Social media has been found to play a vital role in disaster relief operations. It has shown its usefulness by potentially saving lives in several instances (Imran et al., 2015; Kumar & Singh, 2019; Mendon et al., 2021; Bandyopadhyay et al., 2018). For example, a woman was rescued from a hurricane when she asked for help on Twitter because the 911 emergency number was not reachable.<sup>1</sup> She posted, “*I have 2 children with me and tge [sic],water is swallowing us up. Please send help. 11115 Sageview, Houston,Tx, 911 is not responding!!!!!!*”. Due to its potential usages during a disaster, more and more people are posting and forwarding disaster-related information on social media such as Twitter and Facebook (Imran et al., 2015; Kumar et al., 2020; Palshikar et al., 2018; Beydoun et al., 2018; Mirbabaie et al., 2021; Mendon et al., 2021). These disaster-related social media posts contain information about affected individuals, injured and missing people, and damage related to infrastructure (Kumar et al., 2019). It has been found that several disaster-related events have first appeared on social media instead of traditional news channels (Zahra et al., 2020). For example, the news of the Westgate Mall attack in Kenya in 2013 was first reported on Twitter, before it was reported by any TV channels (Zahra et al., 2020). Similarly, an airplane crash over the Hudson River was first reported on Twitter (Sakaki et al., 2012).

Although social media posts can be posted rapidly and contain useful information related to disasters, they are equally filled with redundant and false information which limits their authenticity and usages (Kumar et al., 2020; Tanev et al., 2017; Doggett & Cantarero, 2016; Stefan et al., 2019; Fang et al., 2016; Morstatter et al., 2014). The posts of local citizens and eyewitnesses of disasters convey more authentic information related to the disaster (Zahra et al., 2020). Information provided by the eyewitness can give a better perspective on the disaster (Zahra et al., 2020). Therefore, the content posted by the eyewitness is preferred

<sup>1</sup> <https://twitter.com/RitzWillis/status/901687498732175360>

---

over all the other sources of information (Zahra et al., 2020; Stefan et al., 2019). Hence, extracting the contents shared by the eyewitness of a disaster is very important, but it is equally challenging due to its limited volume in the massive social media posts (Zahra et al., 2020; Morstatter et al., 2014; Tanev et al., 2017). In an analysis of 26 different events, Olteanu et al. (2015) found that only 9% of messages were posted by the eyewitness. One straightforward approach for finding posts of people from the disaster-hit area is through geotagged tweets Kumar et al. (2017). But tweets with geotagged information are rare, as only 1-3% of the total tweets are geo-tagged (Imran et al., 2020; Zahra et al., 2020; Zola et al., 2020). The second option is to use the user's home location to decide the people belonging from the disaster-hit area, but this home location field on the Twitter user's account is very noisy and inaccurate (Kumar et al., 2017). As the geotagged tweets are limited in numbers and users' home location information is noisy, the textual content of the tweet can be used to identify eyewitness tweets using natural language processing techniques (Zahra et al., 2020; Tanev et al., 2017).

Recently, researchers (Zahra et al., 2020; Stefan et al., 2019; Tanev et al., 2017; Fang et al., 2016; Morstatter et al., 2014) have proposed Natural Language Processing (NLP) based approaches using textual features of the messages to identify eyewitness messages from social media. Zahra et al. (2020) applied the Random Forest classifier with uni-gram and bi-gram TF-IDF (Term Frequency-Inverse Document Frequency) features along with several domain-specific features to identify eyewitness tweets. Tanev et al. (2017) used Naive Bayes, Support Vector Machine, and Random Forest classifiers on textual features and meta-data information of tweets to find eyewitness tweets. Morstatter et al. (2014) extracted several features such as uni-gram, bi-gram, crisis-sensitive features, and preposition phrases to train a Naive Bayes classifier. Fang et al. (2016) extracted linguistic features, meta-data features, and word-embedding vectors to train decision tree, random forest, and support vector machine (SVM) classifiers. Most of these works used handcrafted features with conventional machine-learning-based classifiers to identify eyewitness messages, but their models achieved very limited performance. Recently, Stefan et al. (2019) proposed dense neural network and Long-Short-Term-Memory (LSTM) network-based models for the identification of eyewitness messages using bag-of-words and pre-trained GloVe word-embedding vectors. But they tested their models on a very limited number of data samples.

Most of the deep neural network-based models operate over word-embedding vectors. Word-embedding represents each word as a real-valued vector that captures syntactic and semantic information, and thus they are very useful in many natural language processing (NLP) tasks (Kumar & Singh, 2019; Yu et al., 2019). Nowadays, many pre-trained word embeddings are available over the web Qiu et al. (2020), but different word embeddings may encode different aspects of the language (Levy & Goldberg, 2014; Qiu et al., 2020). Zhang et al. (2016) and Pham & Le (2018) explored the idea of integrating different word-

embedding vectors and proposed a convolutional neural network-based model that achieved state-of-the-art performance for sentence classification tasks. Motivated by their works, in this study, we are proposing a multi-channel convolutional neural network (MCNN) that uses three different word embeddings' vectors together for eyewitness tweet identification. The proposed MCNN model uses GloVe<sup>2</sup>, Crisis<sup>3</sup>, and Word2vec embedding vectors together. GloVe and Crisis are the pre-trained models with GloVe being trained on the 27 billion tokens of tweets whereas Crisis is trained on the crisis-related corpus. For word2vec embedding, we trained it on the disaster-specific dataset using Gensim<sup>4</sup>. Irrespective of this, we are also exploring the suitability of deep neural network-based models such as Recurrent Neural Network with attention, Gated Recurrent Unit with attention, Long-short-Term-Memory with attention, and Convolutional Neural Network (CNN) where we used GloVe-embedding vector for the identification of disaster-related eyewitness tweets. The proposed system is validated on four different disasters related to floods, earthquakes, hurricanes, and wildfires. The overall contributions of the paper can be summarized below:

- Proposing a multi-channel convolutional neural network (MCNN) for the identification of disaster-related eyewitness tweets. The proposed MCNN model performed better than the existing methods by achieving an  $F_1$ -score of 0.84, 0.88, 0.84, and 0.86 for flood, earthquake, hurricane, and wildfire events respectively.
- The proposed MCNN uses three different word embeddings to capture the better contextual meaning of the tweets.
- The performance of the proposed MCNN model is compared with several popular machine-learning and deep-learning models.

The rest of the sections are organized as follows: Section 2 discusses the related work, Section 3 discusses the detailed methodology used to perform the experiments. Section 4 lists the experimental results, Section 5 discusses the findings of the proposed model with some future directions and, finally, Section 6 concludes the paper.

## 2 Related works

Disaster-related social media content is effectively utilized by several researchers (Kumar et al., 2020, 2019; Kumar & Singh, 2019; Stefan et al., 2019; Zahra et al., 2020; Nguyen et al., 2017b; Liu & Xu, 2018) to propose different methods to provide better situational awareness in citizens, to perform better rescue and relief operations, assess the damage automatically, and find the location information of events and posts. Identification of disaster-

<sup>2</sup> <http://nlp.stanford.edu/data/glove.twitter.27B.zip>

<sup>3</sup> [https://crisisnlp.qcri.org/data/lrec2016/crisisNLP\\_word2vec\\_model\\_v1.2.zip](https://crisisnlp.qcri.org/data/lrec2016/crisisNLP_word2vec_model_v1.2.zip)

<sup>4</sup> <https://radimrehurek.com/gensim/>

---

related informative content from social media has been a very active area of research around the globe (Kumar et al., 2020, 2019; Singh et al., 2019; Nguyen et al., 2017a; Caragea et al., 2016; Imran et al., 2014). This informative content contains posts of people from disaster-hit areas as well as from outside the disaster regions. As the posts of local citizens and eyewitnesses of disaster convey more authentic information and can give a better perspective on the disaster, several researchers (Stefan et al., 2019; Zahra et al., 2018, 2020; Tanev et al., 2017; Morstatter et al., 2014) have shown a strong interest in finding eyewitnesses' social media content, i.e., content posted from the disaster-hit area. Therefore, for the better organization of related literature, this section is divided into two subsections: (i) Informative disaster content identification from social media, and (ii) Eyewitness disaster content identification from social media.

## 2.1 Informative disaster content identification from social media

This section briefly discusses some of the potential works for the identification of informative disaster-related content from social media. The convolutional neural network-based model was used by (Caragea et al., 2016; Nguyen et al., 2017a; Kersten et al., 2019; Madichetty & Sridevi, 2019) to classify informative and non-informative disaster-related tweets. Yu et al. (2019) used convolutional neural network-based model to do a fine grain classification of informative disaster-related tweets in *caution and advice*, *casualties and damage*, and *donation and aid* classes. Singh et al. (2019) extracted several features from the textual content of tweets to experiment with support vector machine, random forest, and gradient-boosting classifiers to classify high-priority and low-priority tweets. Nguyen et al. (b) implemented a deep neural network to first classify disaster-related informative and non-informative tweets. They further used informative tweets to identify tweets related to *affected individuals*, *infrastructure and utility damage*, and *sympathy and support*. Kumar et al. (2019) performed a comparative analysis of different machine-learning and deep-learning techniques to classify informative tweets into further classes such as *affected individuals*, *infrastructure and utility damage*, *injured or dead people*, *missing or found people*, *rescue volunteering or donation effort*, and *vehicle damage*. They found that deep-learning-based models perform better than the several popular machine-learning classifiers. Roy et al. (2020) used SVM to classify *cyclone fani* tweets into informative and non-informative classes. They further summarized informative tweets to get better information regarding the disaster. Kumar et al. (2020) built a multi-modal system to classify informative and non-informative tweets of disaster using LSTM and VGG-16 for tweet text and images respectively. Madichetty & Sridevi (2021) proposed a model that uses low-level lexical, syntactic, and top-most frequency features to find damage assessment tweets during a disaster. Mohanty et al. (2021) proposed four differ-

ent models: a geospatial model, an image model, a user model, and a text model to provide situational awareness from social media data during disaster events.

One group of researchers (Alam et al., 2017; Nguyen et al., a,a; Daly & Thom, 2016; Lagerstrom et al., 2016) developed models to filter informative social media images related to disaster to help humanitarian organizations. Alam et al. (2017) proposed a framework to identify relevant images of Queensland Australian Cyclone, 2017, from social media. Nguyen et al. (a) used transfer learning for irrelevant image filtering and the perceptual hashing technique for redundant image filtering from disaster-related social media content. Chaudhuri & Bose (2019) used a convolutional neural network for the identification of human body parts from the earthquake-related images whereas Daly & Thom (2016) and Lagerstrom et al. (2016) developed models to identify fire-related disaster images.

## 2.2 Eyewitness disaster content identification from social media

Kumar et al. (2013) proposed a method to identify the subset of users from social media who can provide quick and relevant information related to an event. They reported that the selected users generate more quality information compared to other social media users. Morstatter et al. (2014) analyzed the Boston Marathon bombing and Hurricane Sandy events and found that several linguistic differences exist between the tweets posted from disaster-affected areas and outside the affected areas. They used features such as uni-gram, bi-gram, crisis-sensitive features, preposition phrases to train a Naive Bayes classifier. They found the best  $F_1$ -score of 0.83 and 0.88 for the Boston bombing and Hurricane Sandy event respectively to classify tweets posted from the disaster-hit area or outside area. Truelove et al. (2015) observed that the eyewitness accounts had fewer location name references but more personal location information such as “my home” by analyzing bushfire events. Different filtering rules for events such as shooting, protests and police activity were used by Doggett & Cantarero (2016) to filter eyewitness tweets. The presence of first-person pronouns, immediate temporal words like just and now, locative markers such as school, here, and exclamative or emotive punctuation were used while defining their filtering rules. Features such as linguistic, crisis-sensitive, conversational, meta-data were utilized by Fang et al. (2016) to classify eyewitness social media accounts. They used machine-learning classifiers such as decision tree, random forest, and support vector machine (SVM). They found that SVM performed best with an  $F_1$ -score of 0.90 when it was trained on linguistic features, meta-data features, and word-embedding vectors. Tanev et al. (2017) used uni-gram, bi-gram, semantic features, and metadata information of the earthquake, wildfire, landslide, and flood tweets to classify them into eyewitness and non-eyewitness classes. They used English and Italian tweets with the Naive Bayes, Support Vector Machine, and Random Forest classifiers. The

**Table 1** Summary of some of the potential related works for the identification of eyewitness social media content

Authors	Dataset	Features	Model	Performance
Morstatter et al. (2014)	Boston Marathon bombing, Hurricane Sandy	Uni-gram, bi-gram, crisis-sensitive features, preposition phrases	Naive Bayes	$F_1$ -score = 0.83-0.88
Fang et al. (2016)	Cyclone, Earthquake, Shooting, Bombing, etc	Linguistic features, metadata features, word embedding vectors, etc.	SVM	$F_1$ -score = 0.90
Tanev et al. (2017)	Earthquake, wildfire, landslide, and flood	Uni-gram, bi-gram, metadata, semantic features, etc.	Random Forest, Naive Bayes, SVM	$F_1$ -score = 0.69-0.79
Zahra et al. (2020)	Flood, earthquake, hurricane, and wildfire	Uni-gram and bi-gram TF-IDF, + domain-specific features	Random Forest	$F_1$ -score = 0.40-0.92
Stefan et al. (2019)	California wildfire, Iran-Iraq earthquake, and Hurricane Harvey	Word-embedding, pronouns, Time-specific words, Location-specific words, feeling-related words	Naive Bayes, Logistic regression, LSTM, etc.	Precision = 0.79-1.00
Pekar et al. (2020)	Colorado wildfires, Philippines floods, Australia bushfire, etc.	Parts-of-speech	Support vector machine, K-nearest neighbor, Naive Bayes, and ensembles methods	$F_1$ -score < 0.5
Truelove et al. (2017)	Australian Football League	Bag-of-visual-words features	Linear support vector machine	Accuracy = 90.17%

best  $F_1$ -score of 0.79 for English language tweets and 0.69 for Italian language tweets were achieved with Random Forest and Naive Bayes classifiers respectively.

Zahra et al. (2018) analyzed the different disaster-related tweets and labeled them into three different classes: (i) eyewitness, (ii) non-eyewitness, and (iii) don't know. They further labeled eyewitness tweets into three different classes: direct eyewitness, indirect eyewitness, and vulnerable direct eyewitness. They observed that tweets expressing terms like feeling, seeing, and hearing appeared often in direct eyewitness tweets, whereas prayers, emotions, and thoughts appeared frequently in indirect eyewitness tweets. Zahra et al. (2020) classified flood, earthquake, hurricane, and wildfire tweets into eyewitness, non-eyewitness, and don't know classes. They used uni-gram, bi-gram TF-IDF, and several domain-specific features. They reported that the Random Forest classifier outperformed the other machine-learning classifiers in identifying eyewitness tweets by achieving an  $F_1$ -score of 0.57, 0.92, 0.60, and 0.40 for flood, earthquake, hurricane, and wildfire events respectively. Stefan et al. (2019) applied several classifiers such as Naive Bayes, logistic regression, and neural network-based approach to classifying tweets into eyewitness and non-eyewitness classes. They manually labeled datasets related to the California wildfire, Iran-Iraq earthquake, and Hurricane Harvey. But, their data size was limited having only one thousand tweets. Their best-performed models achieved an average precision of 0.79, 1.0, and 0.89 for the California wildfire, Iran-Iraq earthquake, and Hurricane Harvey respectively. Pekar et al. (2020)

applied several machine-learning classifiers such as SVM, K-nearest neighbor, Naive Bayes, and ensembles methods on four different tasks: identification of eyewitness reports, detection of relevant messages, informative messages, and topical classification of messages. They performed experiments in several different settings to combine the disaster-specific dataset into one to test the performance of models in case of heterogeneous disaster events. They stated that the identification of eyewitness tweets is very hard and found that none of the models achieved  $F_1$ -score of more than 50%. Truelove et al. (2017) extracted bag-of-visual-words features from the images and applied several machine-learning classifiers such as decision tree, Naive Bayes, random forest, linear support vector machine to classify images posted from a witness account or not. Their implemented linear support vector machine classifiers performed best with an accuracy of 90.17%. The summary of some of the potential works for the identification of eyewitness social media content can be seen in Table 1.

Most of the earlier works used hand-crafted features and trained conventional machine-learning classifiers to identify eyewitness messages. In this work, we are exploring different recurrent neural network-based models with attention mechanism, convolutional neural network, and multi-channel convolutional neural network with three different word-embedding techniques such as GloVe, Crisis, and Word2vec to classify tweets into eyewitness, non-eyewitness, and don't know classes. The detailed description of the proposed models and word-embedding techniques can be seen in section 3.

### 3 Methodology

The extensive experiments were performed with seven conventional machine-learning models and five deep-learning models. The conventional machine-learning classifiers are: (i) Support Vector Machine (SVM), (ii) Naive Bayes (NB), (iii) K-Nearest Neighbors (KNN), (iv) Logistic Regression (LR), (v) Decision Tree (DT), (vi) Random Forest (FT), and (vii) Gradient Boosting (GB). The deep-learning models are (i) Recurrent Neural Network with Attention (RNN-A), (ii) Gated Recurrent Unit with Attention (GRU-A), (iii) Long-Short-Term-Memory with Attention (LSTM-A), (iv) Convolutional Neural Network (CNN), and (v) Multichannel Convolutional Neural Network (MCNN).

#### 3.1 Data description

The proposed work uses the dataset<sup>5</sup> published by (Zahra et al., 2020). The dataset includes tweets related to four different disasters, such as earthquakes, floods, fires, and hurricanes.

---

<sup>5</sup> [https://crisisnlp.qcri.org/data/eyewitness\\_tweets\\_annotations\\_14k\\_public.zip](https://crisisnlp.qcri.org/data/eyewitness_tweets_annotations_14k_public.zip)

**Table 2** The statistics of the datasets used in this study

	Earthquake	Flood	Hurricane	Wildfire
Eyewitness	1,967	775	763	189
Non-eyewitness	521	664	1,299	1,379
Don't know	1,512	2,561	1,938	432
Total	4,000	4,000	4,000	2,000

Three different levels were assigned to tweets, such as *eyewitness*, *non-eyewitness*, and *don't know*. Two sets of datasets were provided. In set-1, the datasets were labeled by authors and their groups. In set 2, the labeling was done using crowdsourcing. Out of a total of four different disaster-related events such as flood, earthquake, hurricane, and wildfire, the wildfire event dataset was labeled by crowdsourcing only, whereas for other events both crowdsourcing and authors manual labeling was done. We merged the event-specific dataset of manual analysis and crowdsourcing into one for earthquake, flood, and hurricane. The overall statistic of the dataset used in this study can be seen in Table 2. To perform the pre-processing of tweets, special characters such as “!, #, %, &, ?, @” and extra space, as well as tab between the words, were removed. Finally, tweets were transformed to lowercase, which is then employed by several conventional machine-learning and deep-learning models.

### 3.2 Representation of tweets

In the case of conventional machine-learning models, tweets are represented using TF-IDF vectors. For the calculation of TF-IDF vectors, uni-gram, bi-gram, and tri-gram words of the tweets are used. In the case of deep-learning models, tweets are represented using the word-embedding technique. The word-embedding technique represents each word of the tweet corpus in a real-value fixed-dimensional vector. The word-embedding generates a similar word vector for words having a similar meaning. This work uses three different word-embedding vectors: (i) GloVe<sup>6</sup>, (ii) Crisis<sup>7</sup>, and (iii) Word2vec. GloVe and Crisis are the pre-trained models with GloVe being trained on the 27 billion tokens of tweets whereas Crisis is trained on the crisis-related corpus. For Word2vec embedding, we trained it on the disaster-specific dataset using Gensim CBOW strategy<sup>8</sup>. We have used 100-dimensional, 300-dimensional, and 100-dimensional word vectors for GloVe, Crisis, and Word2vec embedding respectively. The complete tweet matrix after embedding each word of the tweet into a pre-defined fixed dimension of the real-valued vector can be represented by  $T_i$ .

<sup>6</sup> <http://nlp.stanford.edu/data/glove.twitter.27B.zip>

<sup>7</sup> [https://crisisnlp.qcri.org/data/lrec2016/crisisNLP\\_word2vec\\_model\\_v1.2.zip](https://crisisnlp.qcri.org/data/lrec2016/crisisNLP_word2vec_model_v1.2.zip)

<sup>8</sup> [https://radimrehurek.com/gensim\\_3.8.3/models/word2vec.html](https://radimrehurek.com/gensim_3.8.3/models/word2vec.html)

$$T_i = \begin{matrix} & W_1 & W_2 & W_3 & \dots & W_m \\ \begin{bmatrix} r_{11} & r_{21} & r_{31} & \dots & r_{m1} \\ r_{12} & r_{22} & r_{32} & \dots & r_{m2} \\ r_{13} & r_{23} & r_{33} & \dots & r_{m3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{1k} & r_{2k} & r_{3k} & \dots & r_{mk} \end{bmatrix} \end{matrix}$$

where  $T_i$  represents the embedding matrix of the tweets having  $m$  words. The tweets that have fewer than  $m$  words are padded to make it of the dimension of  $m$  and the tweets with greater than  $m$  words are curtailed to make it of the dimension of  $m$ . For word,  $W_m$ , embedding is represented by the vector  $[r_{m1} \ r_{m2} \ \dots \ r_{mk}]$ , where  $k$  represents the embedding dimension of the word. The length of the tweet ( $m$ ) is fixed at 30 for the experiments because most of the tweets in the dataset had at most 30 words.

### 3.3 Convolutional Neural Network

The convolutional neural network uses different  $n$ -gram filters to extract features from the tweet matrix followed by the pooling operation. The overall architecture of the convolutional neural network can be seen in Figure 1. The convolution process applies a filter with the dimension of  $(w \times k)$ , where  $w$  is the length of the words and  $k$  is the embedding dimension of the word. The filter applied on the tweet matrix first performs element-wise multiplication with the first  $w$  words of the tweets and then the summation of all the multiplied values is passed through an activation function to produce a feature. This filter is again applied to the  $w$  words of the tweets by moving one step down, which means the filter is applied on the  $w$  words of the tweet matrix by leaving the first column of the tweet matrix. A simple convolution operation on the tweet matrix with the filter size of  $(w = 3)$  can be represented as:

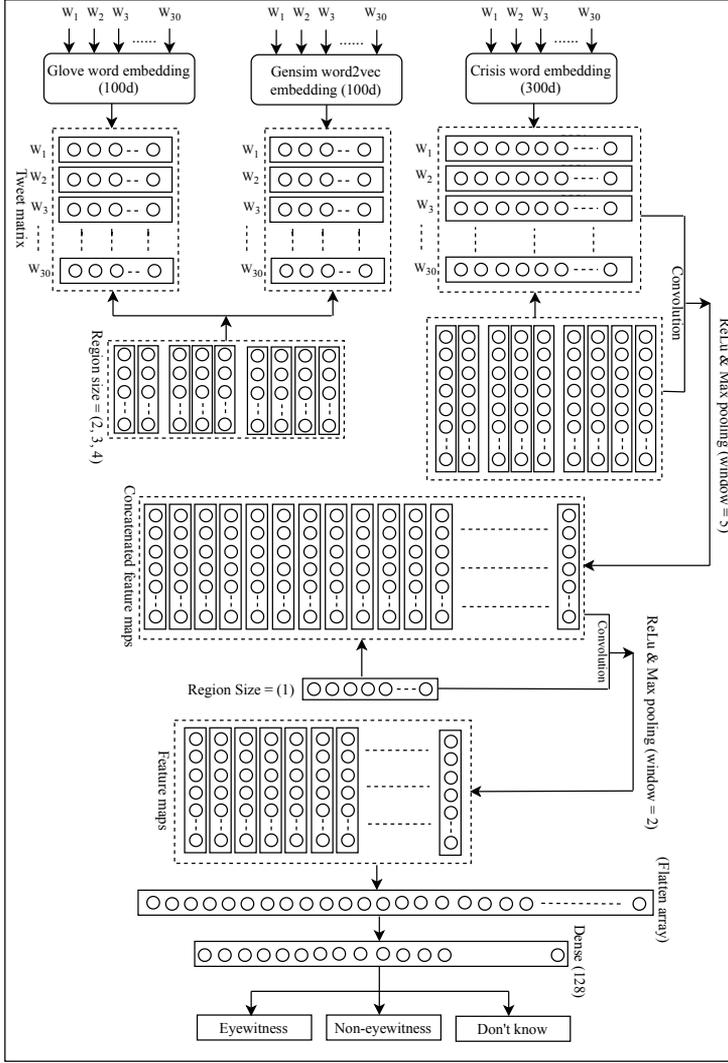
$$T_i = \begin{matrix} & W_1 & W_2 & W_3 & \dots & W_m \\ \begin{bmatrix} r_{11} & r_{21} & r_{31} & \dots & r_{m1} \\ r_{12} & r_{22} & r_{32} & \dots & r_{m2} \\ r_{13} & r_{23} & r_{33} & \dots & r_{m3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{1k} & r_{2k} & r_{3k} & \dots & r_{mk} \end{bmatrix} \bullet \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \\ \vdots & \vdots & \vdots \\ a_{1k} & a_{2k} & a_{3k} \end{bmatrix} \end{matrix}$$

$$\begin{bmatrix} cf_1 \\ cf_2 \\ cf_3 \\ \vdots \\ cf_{(m-w+1)} \end{bmatrix} = \begin{bmatrix} F\{r_{11}a_{11} + r_{21}a_{21} + r_{31}a_{31} + r_{12}a_{12} + r_{22}a_{22} + r_{32}a_{32} + \dots + \\ r_{1k}a_{1k} + r_{2k}a_{2k} + r_{3k}a_{3k}\} \\ F\{r_{21}a_{11} + r_{31}a_{21} + r_{41}a_{31} + r_{22}a_{12} + r_{32}a_{22} + r_{42}a_{32} + \dots + \\ r_{2k}a_{1k} + r_{3k}a_{2k} + r_{4k}a_{3k}\} \\ F\{r_{31}a_{11} + r_{41}a_{21} + r_{51}a_{31} + r_{32}a_{12} + r_{42}a_{22} + r_{52}a_{32} + \dots + \\ r_{3k}a_{1k} + r_{4k}a_{2k} + r_{5k}a_{3k}\} \\ \vdots \\ F\{r_{(m-2)1}a_{11} + r_{(m-1)1}a_{21} + r_{(m)1}a_{31} + r_{(m-2)2}a_{12} + r_{(m-1)2}a_{22} + \\ r_{(m)2}a_{32} + \dots + r_{(m-2)k}a_{1k} + r_{(m-1)k}a_{2k} + r_{(m)k}a_{3k}\} \end{bmatrix}$$

where,  $F(x) = \max(0, x)$  represents the ReLU activation function. The ReLU activation function converts negative values to zero and for the positive value it returns  $x$  itself. The feature map after the convolution operation is represented by  $[cf_1, cf_2, cf_3, \dots, cf_{(m-w+1)}]$ . After getting a feature map from the convolution process, max-pooling operation is performed, which pools the maximum value from a window size of  $p$ , as given in equation 1.

$$\hat{cf}_1 = \max(cf_i, cf_{i+1}, cf_{i+2}, \dots, cf_p), i \geq 1 \quad (1)$$

In this work, two different convolutional neural network-based models were implemented. In the case of a single-channel convolutional neural network (CNN), only GloVe embedding was used whereas in the case of the multi-channel convolutional neural network (MCNN), three different channels were used with three different word-embedding vectors. The overall architecture of the multi-channel CNN can be seen from Figure 1. Three different word-embedding vectors GloVe, Crisis, and Word2vec are used at the three-channel of multi-channel CNN, as shown in Figure 1. Three different tweet matrixes were obtained with the size of  $(30 \times 100)$ ,  $(30 \times 100)$ , and  $(30 \times 300)$  for GloVe, Word2vec, and Crisis embedding respectively. We applied 128 filters of 2-gram, 3-gram, and 4-gram on the tweet matrix to extract relevant features from them. After the convolution operation, a max-pooling operation was performed with a window size of 5. The vector after the max-pooling from separate channels is then concatenated with each other as shown in Figure 1. The second convolution operation is then applied to the concatenated feature maps with 128 filters of 1-gram followed by max-pooling operation with the windows size of 5. The feature maps obtained from the second convolutional operation are flattened and passed through a dense layer having 128 neurons. Finally, the dense layers having 128 neurons were passed through a softmax layer to calculate the probability of the tweets belonging to eyewitness,



**Fig. 1** Proposed diagram of convolutional neural network for eyewitness classification

non-eyewitness, and don't know classes. We used categorical cross-entropy as a loss function to calculate loss between true and predicted values and Adam (Kingma & Ba, 2014) as the optimizer to back-propagate the loss in the network. The categorical cross-entropy is defined by equation 2.

$$\text{Categorical cross-entropy} = - \sum_{i=1}^N V_c \times \log(\hat{y}_i) \quad (2)$$

Where,  $V_c$  represents the classes into the one-hot vector form, and  $\hat{y}_i$  is the predicted class probability of the model for the  $i^{\text{th}}$  training sample.

**Table 3** Description of the hyper-parameters used in the case of deep-learning models

Hyper-parameters	MCNN	CNN	RNN-A	GRU-A	LSTM-A
Number of CNN layers	2	2	-	-	-
Number of dense layers	2	2	1	1	1
Filter size	(2, 3, 4), 1	(2, 3, 4), 1	-	-	-
Number of filters	128	128	-	-	-
Max-pooling window	5	5	-	-	-
Number of RNN layers/units	-	-	2, (256 & 128)	-	-
Number of GRU layers/units	-	-	-	2, (256 & 128)	-
Number of LSTM layers/units	-	-	-	-	2, (256 & 128)
Number of neurons at dense layer	128, 3	128, 3	3	3	3
Activation functions	ReLU, Softmax	ReLU, Softmax	Softmax	Softmax	Softmax
Learning rate	0.001	0.001	0.001	0.001	0.001
Batch size	100	100	100	100	100
Epochs	100	100	100	100	100

### 3.4 Attention-based recurrent neural network models

The attention-based techniques performed well in several natural language-processing tasks (Yang et al., 2016; Wang et al., 2016) that motivated us to use this technique with RNN, GRU, and LSTM networks. Attention-based techniques can identify eyewitness tweets efficiently as it is sensitive to distinctive textual features. The attention layer averages the weights of the input sequence and propagates these weights through the softmax layers to obtain the importance of the words. A detailed explanation of the attention-based mechanism can be seen in (Vaswani et al., 2017). The detailed explanation of the working of RNN, GRU, and LSTM models can be seen in (Hochreiter & Schmidhuber, 1997; Chung et al., 2014; Mikolov et al., 2010).

During experiments with conventional machine-learning classifiers, the default parameters provided in the Sklearn<sup>9</sup> library are utilized, whereas deep-learning models require a comprehensive sensitivity analysis since they are extremely sensitive to the chosen hyper-parameters. Therefore, extensive testing was conducted to find the proper hyper-parameters for deep-learning models. Table 3 lists the best-suited hyper-parameters for deep-learning models. Section 3.5 provides a complete description of selecting the best-suited hyper-parameters.

### 3.5 Hyper-parameter selection for deep neural network models

The proposed multi-channel CNN model consists of many hyper-parameters such as the number of filters, filter size, max-pooling window size, learning rate, batch size, activation functions, number of epochs, number of CNN layers, number of dense layers, number of

<sup>9</sup> <https://scikit-learn.org/stable/>

neurons in dense layer and others. As the performance of the deep-learning models is very sensitive to the selected parameters, therefore the right choice of hyper-parameter selection is very important. To get the best set of hyper-parameters, first, we fixed the batch size of 80, max-pooling size of 5, and the learning rate to its default value i.e., 0.001, then started experiments with 2 layers of CNN and 2 layers of dense (128 and 3 neurons respectively at dense layers) where different combinations of 1-gram, 2-gram, 3-gram, and 4-gram filters were applied on each of the CNN layers by varying the number of filters. The extensive experiments were performed for 100 epochs, we found 128 filters of 2-gram, 3-gram, and 4-gram filters at the first CNN layer, and 128 filters of 1-gram at the second CNN layer performed best in the case of the multi-channel CNN model. Next, experiments were performed with different combinations of batch size and learning rate where the batch sizes of 80, 100, and 120 and learning rates of 0.01, 0.001, and 0.0001 were used. The multi-channel CNN performed best with a batch size of 100 and a learning rate of 0.001. Next, with the best-obtained batch size, learning rate, number of filters, and filter size, we experimented with one layer of multi-channel CNN, two layers of multi-channel CNN, and three layers of multi-channel CNN. The best performance was achieved in the case of two layers of multi-channel CNN. The detailed hyper-parameters for the multi-channel CNN model can be seen in Table 3.

As the best result was found with the multi-channel CNN with the batch size of 100, the learning rate of 0.001, we fixed this value for other experiments. In the case of single-channel CNN (CNN), the same set of experiments were performed with the number of filters and filter size. The best-performed hyper-parameters can be seen in Table 3. In the case of RNN with attention (RNN-A), GRU with attention (GRU-A), and LSTM with attention (LSTM-A) experiments were performed by varying the number of layers and number of hidden output units. The best performance was achieved with 2 layers having 256 and 128 hidden outputs units for each of the models. The detailed hyper-parameters for the RNN-A, GRU-A, and LSTM-A models can be seen in Table 3.

## 4 Results

All of the experiments were carried out on the Google Collaboratory<sup>10</sup> platform, which includes an Nvidia K80/T4 GPU and 16 GB of RAM. Keras<sup>11</sup> Python deep-learning library with Tensorflow<sup>12</sup> as a backend is used to create deep-learning models, whereas Sklearn<sup>13</sup> Python library is used to implement conventional machine-learning classifiers. The exten-

---

<sup>10</sup> <https://colab.research.google.com/>

<sup>11</sup> <https://keras.io/>

<sup>12</sup> <https://www.tensorflow.org/>

<sup>13</sup> <https://scikit-learn.org/stable/>

**Table 4** The weighted average of the precision (P), recall (R), and  $F_1$ -score ( $F_1$ ) for the conventional machine-learning classifiers (10-fold)

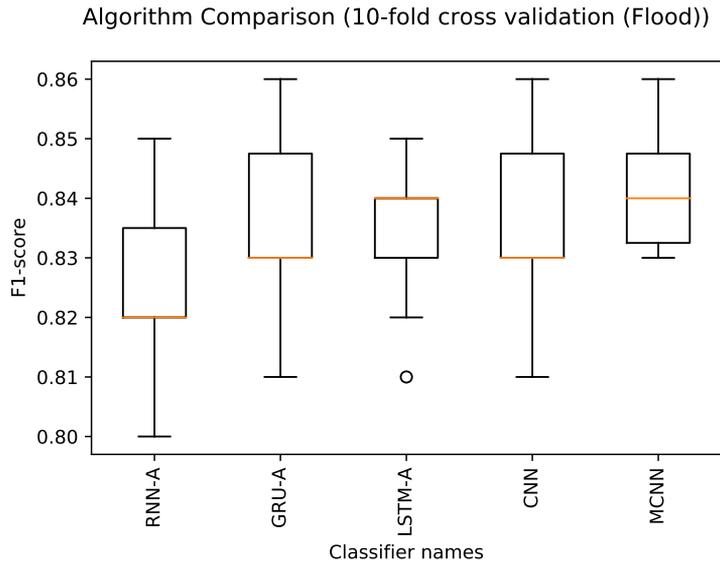
Classifier	Flood			Earthquake			Hurricane			Wildfire		
	P	R	$F_1$									
SVM	0.41	0.64	0.50	0.24	0.49	0.33	0.24	0.48	0.32	0.48	0.69	0.56
RF	0.76	0.76	0.72	0.83	0.82	0.81	0.77	0.78	0.76	0.81	0.82	0.79
LR	<b>0.78</b>	<b>0.78</b>	<b>0.75</b>	<b>0.87</b>	<b>0.87</b>	<b>0.86</b>	<b>0.79</b>	<b>0.79</b>	<b>0.77</b>	0.76	0.73	0.65
KNN	0.75	0.76	0.75	0.80	0.77	0.76	0.77	0.77	0.76	0.75	0.76	0.73
NB	0.73	0.75	0.73	0.81	0.81	0.81	0.73	0.73	0.73	0.74	0.76	0.74
GB	0.72	0.74	0.73	0.78	0.77	0.76	0.76	0.77	0.76	<b>0.81</b>	<b>0.81</b>	<b>0.80</b>
DT	0.68	0.67	0.67	0.75	0.73	0.72	0.71	0.72	0.71	0.78	0.79	0.78

sive experiments were performed with 10-fold cross-validation (Kohavi et al., 1995) for both conventional machine-learning and deep-learning models. In 10-fold cross-validation, the entire data sample is randomly divided into 10 subsamples, in which one subsample is used to validate the system while the system is trained with nine subsamples. This process is repeated 10 times, with each of the ten subsamples being used only once as the validation data. The results from each of the folds are averaged to estimate the overall system performance. The performance of the models is evaluated using precision, recall,  $F_1$ -score, accuracy, and Area Under Curve (AUC). The detailed description of each of the evaluation metrics can be seen in (Powers, 2011).

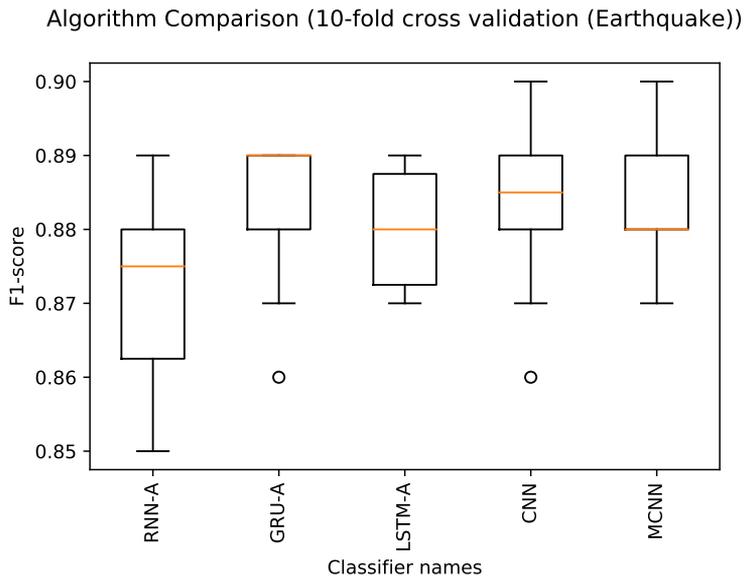
First, the experimentation was performed with seven different conventional machine-learning classifiers. The uni-gram, bi-gram, and tri-gram TF-IDF features of the tweets were used to classify tweets. The weighted averages of the precision, recall, and  $F_1$ -score for each of the classifiers are listed in Table 4. In the case of flood, earthquake, and hurricane, the logistic regression (LR) classifier performed best with the weighted  $F_1$ -score of 0.75, 0.86, 0.77, respectively. In the case of wildfire, the gradient boosting (GB) classifier performed better with a weighted  $F_1$ -score of 0.80.

Next, the experimentation was performed with different deep neural network models. The range of the weighted average of  $F_1$ -score for 10-fold cross-validation for different events is plotted using the box-whisker plot. The box-whisker plot of the RNN-A, GRU-A, LSTM-A, CNN, and MCNN models for flood, earthquake, hurricane, and wildfire events can be seen from Figure 2, Figure 3, Figure 4, and Figure 5 respectively.

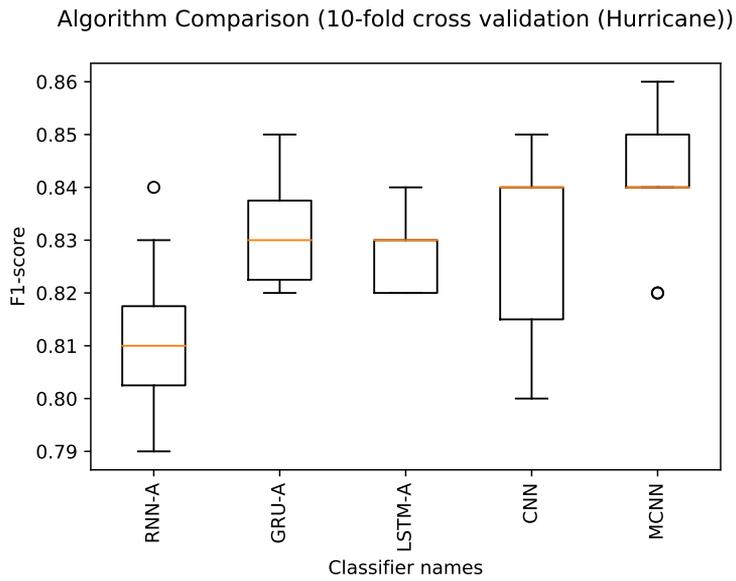
For the flood event, the results of the different deep-learning models are listed in Table 5. The multi-channel CNN (MCNN) and signal channel CNN (CNN) both achieved a weighted  $F_1$ -score of 0.84 and AUC of 0.93, which is better than other deep-learning models and conventional machine-learning models. Among all the implemented deep-learning



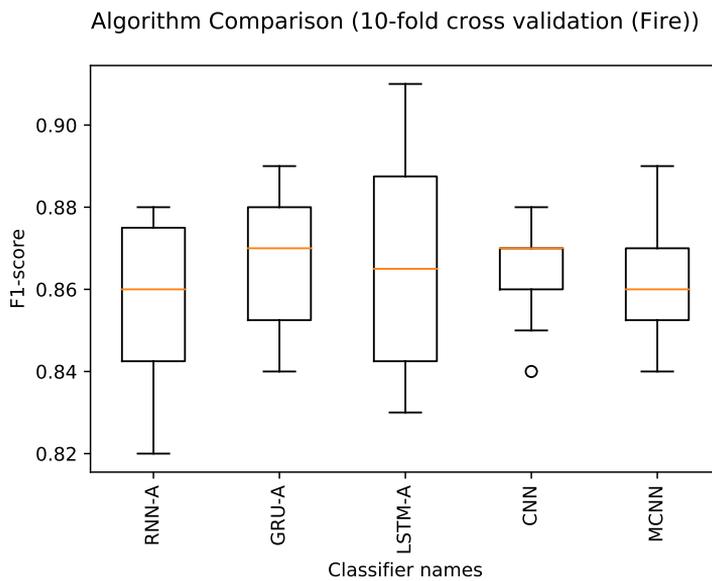
**Fig. 2** Classification comparison with 10-fold cross validation in case of flood event



**Fig. 3** Classification comparison with 10-fold cross validation in case of earthquake event



**Fig. 4** Classification comparison with 10-fold cross validation in case of hurricane event



**Fig. 5** Classification comparison with 10-fold cross validation in case of wildfire event

**Table 5** Results of the different deep-learning models for the flood event

	Class	Precision	Recall	$F_1$ -score	Accuracy	AUC
RNN-A	Eyewitness	0.74	0.61	0.66	0.83	0.91
	Non-eyewitness	0.74	0.72	0.72		
	Don't know	0.88	0.93	0.90		
	Weighted avg.	0.83	0.83	0.83		
GRU-A	Eyewitness	0.75	0.65	0.69	0.84	0.92
	Non-eyewitness	0.75	0.72	0.73		
	Don't know	0.89	0.93	0.91		
	Weighted avg.	0.84	0.84	0.84		
LSTM-A	Eyewitness	0.75	0.62	0.68	0.84	0.92
	Non-eyewitness	0.75	0.73	0.74		
	Don't know	0.88	0.93	0.91		
	Weighted avg.	0.84	0.84	0.83		
CNN	Eyewitness	0.76	0.64	0.69	0.84	0.93
	Non-eyewitness	0.75	0.72	0.73		
	Don't know	0.88	0.93	0.91		
	Weighted avg.	0.84	0.84	0.84		
Multi-channel CNN	Eyewitness	0.76	0.65	0.70	<b>0.84</b>	<b>0.93</b>
	Non-eyewitness	0.74	0.74	0.74		
	Don't know	0.90	0.93	0.91		
	Weighted avg.	<b>0.84</b>	<b>0.84</b>	<b>0.84</b>		

models, the multi-channel CNN (MCNN) performed best for the eyewitness class with an  $F_1$ -score of 0.70. For the earthquake event, the results of the different deep-learning models are listed in Table 6. Among all the implemented deep-learning models, GRU-A, LSTM-A, and MCNN performed equally well and achieved a weighted  $F_1$ -score of 0.88 and AUC of 0.95. The CNN model performed comparatively well with a weighted  $F_1$ -score of 0.88 and AUC of 0.96. For the eyewitness class, all models GRU-A, LSTM-A, CNN, and MCNN achieved an  $F_1$ -score of 0.90. For hurricane event, the results of the different deep-learning models are listed in Table 7. The multi-channel CNN outperformed all the deep-learning and conventional machine-learning classifiers with the weighted  $F_1$ -score of 0.84 and AUC of 0.94.

For the wildfire event, the results of the different deep-learning models are listed in Table 8. Among all the implemented deep-learning models, GRU-A, LSTM-A performed well with a weighted  $F_1$ -score of 0.87 and AUC of 0.92, whereas the single-channel CNN and multi-channel CNN achieved an  $F_1$ -score of 0.86 and AUC of 0.93. For the eyewitness class, LSTM-A performed best with an  $F_1$ -score of 0.67 whereas multi-channel CNN achieved an  $F_1$ -score of 0.60. The performance of the models for the eyewitness class is comparatively

**Table 6** Results of the different deep-learning models for the earthquake event

	Class	Precision	Recall	$F_1$ -score	Accuracy	AUC
RNN-A	Eyewitness	0.89	0.90	0.89	0.87	0.94
	Non-eyewitness	0.94	0.83	0.88		
	Don't know	0.84	0.86	0.85		
	Weighted avg.	0.88	0.87	0.87		
GRU-A	Eyewitness	0.90	0.91	0.90	0.88	0.95
	Non-eyewitness	0.92	0.84	0.88		
	Don't know	0.85	0.87	0.86		
	Weighted avg.	0.88	0.88	0.88		
LSTM-A	Eyewitness	0.90	0.90	0.90	0.88	0.95
	Non-eyewitness	0.91	0.84	0.87		
	Don't know	0.85	0.86	0.86		
	Weighted avg.	0.88	0.88	0.88		
CNN	Eyewitness	0.89	0.91	0.90	<b>0.88</b>	<b>0.96</b>
	Non-eyewitness	0.94	0.82	0.88		
	Don't know	0.85	0.86	0.86		
	Weighted avg.	<b>0.88</b>	<b>0.88</b>	<b>0.88</b>		
Multi-channel CNN	Eyewitness	0.89	0.91	0.90	<b>0.88</b>	<b>0.95</b>
	Non-eyewitness	0.95	0.85	0.89		
	Don't know	0.86	0.86	0.86		
	Weighted avg.	<b>0.89</b>	<b>0.89</b>	<b>0.88</b>		

low. The possible reason for this can be a lower number of eyewitness samples (189 samples) were available (as can be seen from Table 2) for the wildfire event.

## 5 Discussion

The major finding of this research is that the multi-channel convolutional neural network (MCNN) is a better model to segregate eyewitness and non-eyewitness posts from Twitter compared to conventional machine-learning models and several other deep-learning models. The use of three different word-embedding vectors, GloVe, Crisis, and Word2vec in the multi-channel CNN, performed well as the features extracted from three different channels of CNN using 2-gram, 3-gram, and 4-gram filters provide robust features for the identification of eyewitness tweets. As the proposed system automatically identifies the most suitable features in their hidden layers for the identification of eyewitnesses, it does not need any feature engineering. In the case of the flood event, the  $F_1$ -score of multi-channel CNN was improved by 9% in comparison to the logistic regression model of conventional machine-learning models whereas it performed comparatively well in the comparison of other deep-learning models. In the case of the earthquake event, the  $F_1$ -score of multi-

**Table 7** Results of the different deep-learning models for the hurricane event

	Class	Precision	Recall	$F_1$ -score	Accuracy	AUC
RNN-A	Eyewitness	0.69	0.61	0.64	0.82	0.91
	Non-eyewitness	0.84	0.83	0.83		
	Don't know	0.85	0.89	0.87		
	Weighted avg.	0.81	0.82	0.81		
GRU-A	Eyewitness	0.70	0.68	0.68	0.83	0.93
	Non-eyewitness	0.85	0.83	0.84		
	Don't know	0.87	0.89	0.88		
	Weighted avg.	0.83	0.83	0.83		
LSTM-A	Eyewitness	0.70	0.70	0.70	0.83	0.93
	Non-eyewitness	0.84	0.83	0.83		
	Don't know	0.88	0.89	0.88		
	Weighted avg.	0.83	0.83	0.83		
CNN	Eyewitness	0.70	0.67	0.69	0.83	0.93
	Non-eyewitness	0.84	0.85	0.84		
	Don't know	0.88	0.88	0.88		
	Weighted avg.	0.83	0.83	0.83		
Multi-channel CNN	Eyewitness	0.74	0.68	0.70	<b>0.84</b>	<b>0.94</b>
	Non-eyewitness	0.85	0.85	0.85		
	Don't know	0.88	0.90	0.89		
	Weighted avg.	<b>0.84</b>	<b>0.84</b>	<b>0.84</b>		

channel CNN was improved by 2% in comparison to the best-performed logistic regression model of conventional machine-learning models. In the case of hurricane event, the  $F_1$ -score of multi-channel CNN improved by 7% in comparison to the best-performed logistic regression model of conventional machine-learning models. In the case of wildfire event, the  $F_1$ -score of multi-channel CNN improved by 6% in comparison to the best-performed gradient-boosting model of conventional machine-learning models.

Our proposed multi-channel CNN outperformed the earlier work by (Zahra et al., 2020) by a significant margin. For a fair comparison, we implemented our proposed MCNN model with the same set of datasets (without oversampling) used by (Zahra et al., 2020). A comparative  $F_1$ -scores for the classes eyewitness, non-eyewitness, and don't know of the proposed MCNN models with (Zahra et al., 2020) is listed in Table 9. In the case of the flood event, the proposed MCNN model achieved an  $F_1$ -score of 0.72, 0.77, and 0.84 for the eyewitness, non-eyewitness, and don't know classes, which is better than the earlier reported result by (Zahra et al., 2020) on the same dataset. Similarly, for the earthquake event, MCNN achieved an  $F_1$ -score of 0.94, 0.79, 0.39 for eyewitness, non-eyewitness, and don't know classes, respectively. In the case of the hurricane, MCNN achieved an  $F_1$ -score of 0.67, 0.87, and 0.66 for eyewitness, non-eyewitness, and don't know classes, respectively. In the case of a wildfire

**Table 8** Results of the different deep-learning models for the wildfire event

	Class	Precision	Recall	$F_1$ -score	Accuracy	AUC
RNN-A	Eyewitness	0.74	0.54	0.61	0.86	0.92
	Non-eyewitness	0.89	0.95	0.92		
	Don't know	0.81	0.73	0.77		
	Weighted avg.	0.86	0.86	0.86		
GRU-A	Eyewitness	0.76	0.60	0.66	<b>0.87</b>	0.92
	Non-eyewitness	0.90	0.95	0.92		
	Don't know	0.82	0.74	0.78		
	Weighted avg.	<b>0.87</b>	<b>0.87</b>	<b>0.87</b>		
LSTM-A	Eyewitness	0.73	0.63	0.67	<b>0.87</b>	0.92
	Non-eyewitness	0.90	0.95	0.92		
	Don't know	0.82	0.73	0.78		
	Weighted avg.	<b>0.87</b>	<b>0.87</b>	<b>0.87</b>		
CNN	Eyewitness	0.76	0.49	0.59	<b>0.87</b>	<b>0.93</b>
	Non-eyewitness	0.89	0.96	0.92		
	Don't know	0.83	0.76	0.80		
	Weighted avg.	<b>0.87</b>	<b>0.87</b>	<b>0.86</b>		
Multi-channel CNN	Eyewitness	0.70	0.55	0.60	<b>0.87</b>	<b>0.93</b>
	Non-eyewitness	0.89	0.95	0.92		
	Don't know	0.85	0.74	0.79		
	Weighted avg.	<b>0.87</b>	<b>0.87</b>	<b>0.86</b>		

**Table 9** The comparison of the  $F_1$ -scores of the proposed model with the existing model

Events	Class	Zahra et al. (2020)	Proposed MCNN model
		(Domain-expert + text features)	(GloVe, Crisis, & Word2vec embedding)
Flood	Eyewitness	0.57	<b>0.72</b>
	Non-eyewitness	0.70	<b>0.77</b>
	Don't know	0.75	<b>0.81</b>
Earthquake	Eyewitness	0.92	<b>0.94</b>
	Non-eyewitness	0.74	<b>0.79</b>
	Don't know	0.21	<b>0.39</b>
Hurricane	Eyewitness	0.60	<b>0.67</b>
	Non-eyewitness	0.84	<b>0.87</b>
	Don't know	0.64	<b>0.66</b>
Wildfire	Eyewitness	0.40	<b>0.60</b>
	Non-eyewitness	0.91	<b>0.92</b>
	Don't know	0.73	<b>0.79</b>

event, MCNN achieved an  $F_1$ -score of 0.60, 0.92, 0.79 for eyewitness, non-eyewitness, and don't know classes, respectively. In all the events, the proposed MCNN model outperformed the earlier results reported by (Zahra et al., 2020) by a significant margin.

### 5.1 Theoretical Contributions

During a disaster, a lot of irrelevant and redundant information is posted by people that is not useful for humanitarian aid (Kumar et al., 2020; Nguyen et al., 2017a). Therefore, the proposed system requires a filtering system (Kumar et al., 2020; Singh et al., 2019; Nguyen et al., 2017a; Caragea et al., 2016) at the top of it to first filter informative disaster-related tweets. Informative disaster-related tweets are those which are useful for humanitarian aid (Kumar et al., 2020). Then the proposed system can be used to classify these informative tweets into eyewitness, non-eyewitness, and don't know classes to get a better perspective of the disaster. The development of a multi-channel convolutional neural network that employs three different word-embedding vectors to identify disaster-related eyewitness tweets is one of the proposed system's significant theoretical contributions. The usage of three separate word-embedding vectors performs exceptionally well in terms of extracting robust features. Another theoretical contribution is that the suggested multi-channel convolutional neural network extract features automatically in their hidden layers, the proposed system does not require any human effort to extract significant features.

### 5.2 Implications for Practice

The suggested approach identifies eyewitness tweets only based on their textual content, making it simple to incorporate with any social media site. An android application based on the proposed MCNN might be developed to recognise eyewitness disaster tweets from live Twitter streams, allowing individuals to become more situationally aware during a crisis. The proposed system can be easily used by humanitarian organisations to locate disaster eyewitness information for better decision-making and to provide immediate assistance to victims because the posts of local citizens and disaster eyewitnesses convey more authentic information and provide a better perspective on the disaster. The predicted eyewitness social media posts can also be used by the location reference extractor model (Kumar & Singh, 2019; Dutt et al., 2018) to extract the location references mentioned in the social media posts, which can help to locate the victims.

Our proposed multi-channel convolutional neural network identified *“Earthquake in Shinjuku, could feel the entire hotel moving. #earthquake #earthquakeintokyo #massiveearthquake #japan”* and *“So a small earthquake woke me up.... I'm fine btw! A couple things fell in the streets. Never experienced one before so it was something”* as the eyewitness tweet of earthquake event whereas *“Hosman Church needs your help! our ceiling collapsed due to a water flood in the apartment upstairs. Thank you! <https://t.co/wbAYbsn5rb>”* and *“Our friends at Mechanic suffered some horrible flood damage last night. They are great guys. Consider helping them <https://t.co/GLOhCNY4IY>”* as the eyewitness tweets of flood event. The first

---

eyewitness tweet of the earthquake incident includes a crucial situational awareness statement as well as event location references such as *Shinjuku* and *Japan*. Similarly, the third eyewitness tweet about the flood event has details about individuals pleading for aid by giving their location. A similar scenario can be observed in the fourth eyewitness tweet, in which a person requests assistance for a friend during a flood disaster by mentioning their location details. In the case of the second earthquake eyewitness tweet, it can be observed that the eyewitness of the earthquake was simply reporting a minor earthquake by noting his own status, implying that the devastation caused by the earthquake will be low at that location. Therefore, humanitarian and government organisations may use the outcomes of the current work to make better decisions during disasters. The suggested approach may also be used efficiently in a variety of applications such as better event detection, finding credible information, and obtaining useful first-hand information in journalism.

### 5.3 Limitations and Future Directions

The limitation of this work is that the proposed system is trained with English language tweets only, but during the disaster, people also post a considerable amount of tweets in their regional languages. Therefore, the proposed system may not perform equally in those cases. In the proposed work, we did not check the authenticity of the posted tweets which is another limitation of the proposed work. Therefore, in the future, a system can be made to check the authenticity of the posted tweets and a multilingual system can be made to solve the issue of multilinguality. In this work, we have only used textual content of the tweets by discarding images that can contain important visual clues in identifying eyewitness posts. Therefore, a multi-modal system can also be made in the future to get more insight into identifying eyewitness tweets.

## 6 Conclusion

The identification of eyewitness tweets during the disaster provides the reality on the ground. But the identification of eyewitness tweets is very challenging, as a very small fraction of geotagged tweets is available. Identifying eyewitness tweets from their textual content is difficult as tweets are of short length and do not provide enough context about the event. In this work, we have proposed a multi-channel convolutional neural network (MCNN) model to classify tweets into eyewitness, non-eyewitness, and don't know classes. The proposed model overcomes the need for handcrafted features. The proposed MCNN model performs better than the conventional machine-learning models and several other deep-learning models by achieving an  $F_1$ -score of 0.84, 0.88, 0.84, and 0.86 for flood, earthquake, hurricane,

and wildfire respectively. The proposed system can be integrated with any social media platform as it uses only textual information for training the model. As the availability of eyewitness messages is limited, a generative adversarial neural network-based model can be made to deal with the situation of the data imbalance problem.

### Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### References

- Alam, F., Imran, M., & Ofli, F. (2017). 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* (pp. 601–604). ACM.
- Bandyopadhyay, A., Ganguly, D., Mitra, M., Saha, S. K., & Jones, G. J. (2018). An embedding based ir model for disaster situations. *Information Systems Frontiers*, 20, 925–932.
- Beydoun, G., Dascalu, S., Dominey-Howes, D., & Sheehan, A. (2018). Disaster management and information systems: Insights to emerging challenges. *Information Systems Frontiers*, 20, 649–652.
- Caragea, C., Silvescu, A., & Tapia, A. H. (2016). Identifying informative messages in disaster events using convolutional neural networks. In *International conference on information systems for crisis response and management* (pp. 137–147).
- Chaudhuri, N., & Bose, I. (2019). Application of image analytics for disaster response in smart cities. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*.
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- Daly, S., & Thom, J. A. (2016). Mining and classifying image posts on social media to analyse fires. In *ISCRAM* (pp. 1–14). Citeseer.
- Doggett, E., & Cantarero, A. (2016). Identifying eyewitness news-worthy events on Twitter. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media* (pp. 7–13).
- Dutt, R., Hiware, K., Ghosh, A., & Bhaskaran, R. (2018). Savitr: A system for real-time location extraction from microblogs during emergencies. In *Companion Proceedings of the The Web Conference 2018* (pp. 1643–1649).

- 
- Fang, R., Nourbakhsh, A., Liu, X., Shah, S., & Li, Q. (2016). Witness identification in Twitter. In *Proceedings of the Fourth International Workshop on Natural Language Processing for Social Media* (pp. 65–73).
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9, 1735–1780.
- Imran, M., Castillo, C., Diaz, F., & Vieweg, S. (2015). Processing social media messages in mass emergency: A survey. *ACM Computing Surveys (CSUR)*, 47, 1–38.
- Imran, M., Castillo, C., Lucas, J., Meier, P., & Vieweg, S. (2014). Aidr: Artificial intelligence for disaster response. In *Proceedings of the 23rd International Conference on World Wide Web* (pp. 159–162).
- Imran, M., Ofli, F., Caragea, D., & Torralba, A. (2020). Using AI and social media multimodal content for disaster response and management: Opportunities, challenges, and future directions. *Information Processing & Management*, 57. doi:<https://doi.org/10.1016/j.ipm.2020.102261>.
- Kersten, J., Kruspe, A., Wiegmann, M., & Klan, F. (2019). Robust filtering of crisis-related tweets. In Z. Franco, & J. H. Canós (Eds.), *ISCRAM 2019*. URL: <https://elib.dlr.de/127586/>.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (pp. 1137–1145). Montreal, Canada volume 14.
- Kumar, A., & Singh, J. P. (2019). Location reference identification from tweets during emergencies: A deep learning approach. *International journal of disaster risk reduction*, 33, 365–375.
- Kumar, A., Singh, J. P., Dwivedi, Y. K., & Rana, N. P. (2020). A deep multi-modal neural network for informative Twitter content classification during emergencies. *Annals of Operations Research*, (pp. 1–32). doi:<https://doi.org/10.1007/s10479-020-03514-x>.
- Kumar, A., Singh, J. P., & Rana, N. P. (2017). Authenticity of geo-location and place name in tweets. In *Proceedings of the 23rd Americas Conference on Information Systems* (pp. 1–10).
- Kumar, A., Singh, J. P., & Saumya, S. (2019). A comparative analysis of machine learning techniques for disaster-related tweet classification. In *IEEE Region 10 Humanitarian Technology Conference* (pp. 222–227).
- Kumar, S., Morstatter, F., Zafarani, R., & Liu, H. (2013). Whom should I follow? Identifying relevant users during crises. In *Proceedings of the 24th ACM conference on hypertext and social media* (pp. 139–147).
- Lagerstrom, R., Arzhaeva, Y., Szul, P., Obst, O., Power, R., Robinson, B., & Bednarz, T. (2016). Image classification to support emergency situation awareness. *Frontiers in*

- Robotics and AI*, 3, 54.
- Levy, O., & Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 302–308).
- Liu, F., & Xu, D. (2018). Social roles and consequences in using social media in disasters: a structural perspective. *Information Systems Frontiers*, 20, 693–711.
- Madichetty, S., & Sridevi, M. (2019). Detecting informative tweets during disaster using deep neural networks. In *2019 11th International Conference on Communication Systems & Networks (COMSNETS)* (pp. 709–713). IEEE.
- Madichetty, S., & Sridevi, M. (2021). A novel method for identifying the damage assessment tweets during disaster. *Future Generation Computer Systems*, 116, 440–454.
- Mendon, S., Dutta, P., Behl, A., & Lessmann, S. (2021). A hybrid approach of machine learning and lexicons to sentiment analysis: enhanced insights from Twitter data of natural disasters. *Information Systems Frontiers*, (pp. 1–24).
- Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent neural network-based language model. In *Interspeech* (pp. 1045–1048). Makuhari volume 2.
- Mirbabaie, M., Ehnis, C., Stieglitz, S., Bunker, D., & Rose, T. (2021). Digital nudging in social media disaster communication. *Information Systems Frontiers*, 23, 1097–1113.
- Mohanty, S. D., Biggers, B., Sayedahmed, S., Pourebrahim, N., Goldstein, E. B., Bunch, R., Chi, G., Sadri, F., McCoy, T. P., & Cosby, A. (2021). A multi-modal approach towards mining social media data during natural disasters - a case study of hurricane irma. *International journal of disaster risk reduction*, 54, 102032.
- Morstatter, F., Lubold, N., Pon-Barry, H., Pfeffer, J., & Liu, H. (2014). Finding eyewitness tweets during crises. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. Baltimore, MD, USA: Association for Computational Linguistics.
- Nguyen, D., Al Mannai, K. A., Joty, S., Sajjad, H., Imran, M., & Mitra, P. (2017a). Robust classification of crisis-related data on social networks using convolutional neural networks. In *Proceedings of the International AAAI Conference on Web and Social Media* (pp. 632–635). volume 11.
- Nguyen, D. T., Alam, F., Ofli, F., & Imran, M. (a). Automatic image filtering on social networks using deep learning and perceptual hashing during crises. *arXiv preprint arXiv:1704.02602*, . doi:10.48550/ARXIV.1704.02602.
- Nguyen, D. T., Joty, S., Imran, M., Sajjad, H., & Mitra, P. (b). Applications of online deep learning for crisis response using social media information. *arXiv preprint arXiv:1610.01030*, . doi:10.48550/ARXIV.1610.01030.
- Nguyen, D. T., Ofli, F., Imran, M., & Mitra, P. (2017b). 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* (pp. 569–576).
- Olteanu, A., Vieweg, S., & Castillo, C. (2015). What to expect when the unexpected happens: Social media communications across crises. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing* (pp. 994–1009).
- Palshikar, G. K., Apte, M., & Pandita, D. (2018). Weakly supervised and online learning of word models for classification to detect disaster reporting tweets. *Information Systems Frontiers, 20*, 949–959.
- Pekar, V., Binner, J., Najafi, H., Hale, C., & Schmidt, V. (2020). Early detection of heterogeneous disaster events using social media. *Journal of the Association for Information Science and Technology, 71*, 43–54.
- Pham, D.-H., & Le, A.-C. (2018). Exploiting multiple word embeddings and one-hot character vectors for aspect-based sentiment analysis. *International Journal of Approximate Reasoning, 103*, 1–10.
- Powers, D. M. W. (2011). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. doi:10.48550/ARXIV.2010.16061.
- Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences, 63*, 1872–1897.
- Roy, S., Mishra, S., & Matam, R. (2020). Classification and summarization for informative tweets. In *2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)* (pp. 1–4). IEEE.
- Sakaki, T., Okazaki, M., & Matsuo, Y. (2012). Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Transactions on Knowledge and Data Engineering, 25*, 919–931.
- Singh, J. P., Dwivedi, Y. K., Rana, N. P., Kumar, A., & Kapoor, K. K. (2019). Event classification and location prediction from tweets during disasters. *Annals of Operations Research, 283*, 737–757. doi:10.1007/s10479-017-2522-3.
- Stefan, I., Rebedea, T., & Caragea, D. (2019). Classification of eyewitness tweets in emergency situations. In *RoCHI* (pp. 46–52).
- Tanev, H., Zavarella, V., & Steinberger, J. (2017). Monitoring disaster impact: detecting micro-events and eyewitness reports in mainstream and social media. In *ISCRAM*.
- Truelove, M., Khoshelham, K., McLean, S., Winter, S., & Vasardani, M. (2017). Identifying witness accounts from social media using imagery. *ISPRS International Journal of Geo-Information, 6*, 120.
- Truelove, M., Vasardani, M., & Winter, S. (2015). Towards credibility of micro-blogs: characterising witness accounts. *GeoJournal, 80*, 339–359.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information*

- processing systems* (pp. 5998–6008).
- Wang, Y., Huang, M., Zhu, X., & Zhao, L. (2016). Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 606–615).
- Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the Association for Computational Linguistics: human language technologies* (pp. 1480–1489).
- Yu, M., Huang, Q., Qin, H., Scheele, C., & Yang, C. (2019). 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*, 0, 1–18. doi:10.1080/17538947.2019.1574316.
- Zahra, K., Imran, M., & Ostermann, F. O. (2020). Automatic identification of eyewitness messages on Twitter during disasters. *Information processing & management*, 57, 102107.
- Zahra, K., Imran, M., Ostermann, F. O., Boersma, K., & Tomaszewski, B. (2018). Understanding eyewitness reports on Twitter during disasters. In *Proceedings of the of the ISCRAM (2018)* (pp. 687–695).
- Zhang, Y., Roller, S., & Wallace, B. C. (2016). MGNC-CNN: A simple approach to exploiting multiple word embeddings for sentence classification. In *Proceedings of NAACL-HLT* (pp. 1522–1527).
- Zola, P., Ragno, C., & Cortez, P. (2020). A Google Trends spatial clustering approach for a worldwide Twitter user geolocation. *Information Processing & Management*, 57, 102312.