J Grid Computing (2022) 20: 20
https://doi.org/10.1007/s10723-022-09609-y

Randomized Convolutional Neural Network Architecture for Eyewitness Tweet Identification During Disaster

Abhinav Kumar · Jyoti Prakash Singh · Amit Kumar Singh

Received: 19 April 2021 / Accepted: 25 May 2022 / Published online: 16 June 2022

Abstract During a disaster, Twitter is flooded with disaster-related information. Among huge disaster-related Twitter posts, a fraction of them is posted by the eyewitness of disaster. The post of an eyewitness of the disaster contains an authentic description of the disaster. Therefore, eyewitness disaster-related posts are preferred over all other sources of information to know the floor reality of the disaster. In this work, we have used a convolutional neural network (CNN) with randomly initialized weights to extract features from the textual contents of the tweets and proposed three different random neural network-based models. The feature extracted from the untrained random convolutional neural network (RCNN) is passed through a trainable dense neural network (DNN), echo state network (ESN), and extreme learning machine (ELM) to identify eyewitness tweets. The proposed system is validated with hurricane, earthquake, flood, and wildfire datasets. In the extensive experiments with three different random neural network-based models such as RCNN-DNN, RCNN-ESN, RCNN-ELM, and other machine learning and deep learning models such as KNN, Naive Bayes, Decision Tree, Convolutional neural network, and Dense Neural Network, the RCNN-DNN model outperformed all the other models. The RCNN-DNN model achieved impressive performance with a weighted F_1-scores of 0.79, 0.86, 0.79, and 0.85 for hurricane, earthquake, flood, and wildfire, respectively.

Keywords Eyewitness tweets · Disaster · Random convolutional neural network · Echo state network · Extreme learning machine

A. Kumar (✉)
Department of Computer Science and Engineering, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India
e-mail: abhinavanand05@gmail.com

J. P. Singh · A. K. Singh
Department of Computer Science and Engineering, National Institute of Technology Patna, Patna, India

J. P. Singh
e-mail: jps@nitp.ac.in

A. K. Singh
e-mail: amit.singh@nitp.ac.in

1 Introduction

Social media have been found to play a significant role in disaster relief works by potentially saving lives in various instances [1, 11, 14, 30, 37, 46, 49]. Social media posts are rapid and contain various useful information linked to disasters, but they are equally filled with irrelevant and misleading information that limits their authenticity and use [7, 8, 22, 29, 36, 38]. The post of local citizens and eyewitnesses of a disaster relay more credible disaster-related information. Thus, eyewitness knowledge may offer a clearer view of the disaster. Therefore, content reported by the eyewitness is preferred over other sources of information. However, it is challenging to find eyewitness posts as they are very limited and buried under huge social media posts. One straightforward way to get posts from the disaster-hit area is through geotagged tweets. But geotagged tweets are rare, as only 1-3% of the overall tweets are geotagged [15, 49, 53].

A good number of works [8, 29, 36, 38, 49] reported in the past few years used textual contents of the tweets to train machine learning models for identifying eyewitness tweets. Zahra et al. [49] trained Random Forest classifier using domain-specific and TF-IDF (Term-Frequency-Inverse-Document-Frequency) features of the tweets to identify disaster-related eyewitness tweets. Tanev et al. [38] extracted several textual and meta-data information of the tweets to train Naive Bayes, Support Vector Machine, and Random Forest classifiers for eyewitness tweets identification. Similarly, Morstatter et al. [29] trained a Naive Bayes classifier whereas Fang et al. [8] trained Decision Tree, Random Forest, and Support Vector Machine by utilizing the textual features of the tweets to identify eyewitness tweets. Recently, Stefan et al. [36] utilized deep learning-based models such as dense neural networks and Long-Short-Term-Memory (LSTM) network to identify eyewitness messages.

Deep learning and machine learning-based models achieved great success in various natural language processing tasks [26], malware detection [5], and emotion recognition [10, 25]. Researchers have recently explored random neural network models because they are faster and can potentially be used with less computation than current approaches [4, 43]. Echo State Network (ESN) [16, 27] and Extreme Learning Machine (ELM) [12, 13] are two popular random neural network-based models which are efficiently utilized in number of classification and prediction tasks [17, 28, 40]. The ESN extracts the features of a given input by performing a dot product between the input and the randomly generated untrained weight matrix. Interestingly, the features extracted in this way are expressive enough to recognize the input signal, and thus various challenging tasks can be solved by using them as an input to a linear model [24, 42]. Tong & Tanaka [43] proposed a slightly different approach where they used untrained CNN to extract features from the images and used the extracted features to train an ESN to classify images. They achieved state-of-the-art performance with the MNIST image dataset. Similarly, Chang & Futagami [4] also utilized an untrained CNN model to extract visual features from the images and achieved significant performance.

The success of randomly fixed-weight models such as the ESN and ELM motivated us to explore the usability of various random neural network-based models for the identification of disaster-related tweets. In this work, we have utilized the CNN model with randomly assigned weights to extract features from disaster-related tweets. Then the extracted features are used by ESN, ELM, and trainable dense neural network models to identify eyewitness tweets. The proposed models are validated with four different disaster-related events such as earthquake, hurricane, flood, and wildfire. The overall contribution of the paper can be summarized below:

- Proposing three different random neural network-based models such as (i) RCNN-ESN (Random Convolutional Neural Network-Echo State Machine), (ii) RCNN-ELM (Random Convolutional Neural Network-Extreme Learning Machine), and (iii) RCNN-DNN (Random Convolutional Neural Network-Dense Neural Network) to identify disaster-related eyewitness tweets.
- The performance of the proposed system is validated with four different disasters such as hurricane, earthquake, flood, and wildfire.
- The performance of the proposed random neural network-based models is compared with popular machine learning and deep learning models such as K-Nearest Neighbor (KNN), Naive Bayes, Decision Tree, Dense Neural Network, and Convolutional Neural Network.

The rest of the paper is organized as follows: Section 2 lists the related works, Section 3 discusses the proposed methodology in detail. Section 4 lists the findings of the proposed models, Section 5 discusses the findings and Section 6 concludes the paper.

2 Related Works

The disaster-related social media contents have been effectively utilized by several researchers [21–23, 34–36, 49] to disseminate situation awareness in the people and to better organize rescue and relief operations. Among various disaster-related systems, identification of eyewitness contents is one of the important natural language processing tasks [7, 36, 38, 45, 49, 50]. As the current work is focused on exploring the random neural network-based models for the identification of eyewitness tweets, this section is divided into two subsections: (i) Eyewitness disaster-related social media contents identification, and (ii) Random neural network-based models.

2.1 Eyewitness Disaster-Related Social Media Contents Identification

Morstatter et al. [29] studied the Boston Marathon bombing and Hurricane Sandy incidents and observed various linguistic differences between the tweets posted from disaster-affected regions and outside the affected regions. They extracted several features such as crisis-sensitive features, preposition phrases, and n-gram. They trained a Naive Bayes classifier to classify tweets posted from disaster-affected regions and outside the affected regions to achieve an F_1-score of 0.83 and 0.88 for the Boston bombing and Hurricane Sandy events, respectively. Doggett & Cantarero [7] developed a filtering mechanism by utilizing several terms such as the presence of first-person pronouns, temporal words like *now* and *just*, *here*, and exclamative or emotive punctuation marks to identify eyewitness tweets. Fang et al. [8] trained decision tree, random forest, and support vector machine classifiers on several features such as linguistic, crisis sensitive, conversational, and meta-data to identify eyewitness social media accounts. They got the best F_1-score of 0.90 in the case of the support vector machine classifier. Tanev et al. [38] extracted semantic features, metadata feature, and n-gram features together. They trained Naive Bayes, Support Vector Machine, and Random Forest classifiers to classify disaster-related English and Italic tweets in eyewitness and non-eyewitness classes. The random forest classifier was found best for English tweets with an F_1-score of 0.79 whereas, the Naive Bayes classifier performed best with an F_1-score of 0.69 for Italic tweets.

Zahra et al. [50] labeled disaster-related tweets into three different classes (i) eyewitness, (ii) non-eyewitness, and (iii) don't know. The eyewitness tweets were further examined and found that the tweets referencing terms such as feeling, seeing, and hearing mostly appeared in the direct eyewitness tweets, while the indirect eyewitness tweets include prayers, feelings, and thoughts. Zahra et al. [49] extracted several features such as n-gram features and domain-specific features from the flood, earthquake, hurricane, and wildfire tweets. Then they used these features with Random forest classifiers and achieved an F_1-scores of 0.57, 0.92, 0.60, and 0.40 for flood, earthquake, hurricane, and wildfire events, respectively. Stefan et al. [36] labeled California wildfire, Iran-Iraq earthquake, and Hurricane Harvey tweets and applied Naive Bayes, Logistic regression, and deep learning-based approaches to classify eyewitness and non-eyewitness tweets. Pekar et al. [31] applied support vector machines, K-nearest neighbor, Naive Bayes, and ensemble methods to identify eyewitness tweets. They found that the identification of eyewitness tweets is very hard and none of the models achieved a F_1-score of more than 50%. Truelove et al. [44] extracted bag-of-visual-words features from the images and applied several machine learning classifiers to classify images posted from the eyewitness account or not. They reported that the support vector machine classifiers achieved the best accuracy of 90.17%.

2.2 Random neural Network Based Models

Jayawardene & Venayagamoorthy [17] utilized Echo State Network and Extreme Learning Machine to solve high photovoltaic power variability. Katuwal et al. [18] proposed deep Random Vector Functional Link and ensemble deep Random Vector Functional Link neural networks by randomly generating weights of the hidden layers while training the output weights. They validated their proposed system with thirteen different datasets and achieved impressive performance. Qiu et al. [32] proposed a hybrid framework composed of Discrete Wavelet Transform (DWT), Empirical Mode Decomposition (EMD), and Random Vector Functional Link network (RVFL) to predict the short-term electric load. Their proposed system outperformed non-Empirical Model Decomposition benchmarks with the confidence of 0.95%. Katuwal et al. [19] proposed an ensemble-based approach consisting of a decision tree and random vector functional link network to a multi-class classification problem. The proposed system was validated with sixty-five different multi-class datasets. They found their ensemble-based approach achieved significantly better performance in comparison to other state-of-the-art models. Tang et al. [39] proposed a novel hierarchical extreme learning machine (H-ELM) framework for the multi-layer perception. They tested their proposed system for computer vision applications and found that it achieved better performance in comparison to other models. A two hidden layered extreme learning network was proposed in [33] by bringing the actual hidden layer output closer to the expected hidden layer output. For many problems of regression and classification, the proposed two hidden layered ELM was found to outperform the original ELM and several current multilayer ELM variants. Cao et al. [3] proposed a hybrid classifier by using the power of the ELM and the Sparse Representation Classification (SRC). They tested their models with handwritten digit recognition, landmark recognition, and face recognition tasks that achieved substantial improvement over others.

Tong & Tanaka [43] used untrained CNN to extract features from the images and used those extracted features with an ESN to classify images. They achieved a state-of-the-art accuracy of 99.25% on the MNIST image dataset. Chang & Futagami [4] proposed a model that uses reinforcement learning with convolutional reservoir computing. They extracted visual and time-series features from the car racing game dataset using randomly fixed weights of the convolutional neural network and reservoir computing model. They achieved the state of the art performance from their model. They also found that the use of the randomly fixed single layer dense network also achieved good performance for the reinforcement learning tasks. Bianchi et al. [2] proposed an unsupervised approach based on reservoir computing to learn vector representation of a multivariate time series dataset. Their model performed comparatively well and achieved comparable computational performance compared to other reservoir computing models. Some excellent surveys of random neural networks can be found in [41, 47, 48, 51, 52].

The success of random neural network models such as ESN and ELM motivated us to explore the usability of these networks for the identification of disaster-related eyewitness tweets. The proposed model uses the convolutional neural network model with randomly assigned weights to extract features from disaster-related tweets. This strategy can reduce the overall training time of the network as we do not have to train these weights through backpropagation. Then the extracted features are used by Echo State Network, Extreme Learning Machine, and trainable dense neural network models to test the efficiency of these models in the disaster-related eyewitness tweets identification.

3 Methodology

This section discusses the proposed RCNN-DNN (Random Convolutional Neural Network-Dense Neural Network) model in detail. The overall system diagram for the proposed system can be seen in Fig. 1. Along with the RCNN-DNN model, we also explored RCNN-ESN (Random Convolutional Neural Network-Echo State Network) and RCNN-ELM (Random Convolutional Neural Network-Extreme Learning Machine). The detailed description of each of the models and the datasets used to validate the models can be seen in the following subsections.

3.1 Data description

The proposed study uses the dataset[1] published by Zahra et al. [49]. Tweets about four separate incidents, such as earthquakes, floods, fires, and hurricanes, are included in the dataset. Three different labels have been assigned to tweets, such as *eyewitness*, *non-eyewitness*, and *don't know*. They provided two sets of datasets. In set-1, authors and their groups labeled the datasets. In set-2, the labeling was done using crowdsourcing. The wildfire event dataset was labeled by crowdsourcing only, while for the other events, both crowdsourced and authors have labeled the datasets. We have combined event-specific (both authors labeled and crowdsourced) datasets into one for earthquake, flood, and hurricane. For training and testing sets, we divided each dataset into a ratio of 80:20. The overall dataset statistics after merging similar events can be found in Table 1.

3.2 Randomized Convolutional Neural Network (RCNN) for Feature Extraction

The overall diagram for the proposed system can be seen in Fig. 1. To provide the input to the system each word of the tweets is represented by a real-value

[1] https://crisisnlp.qcri.org/data/eyewitness_tweets_annotations_14k_public.zip

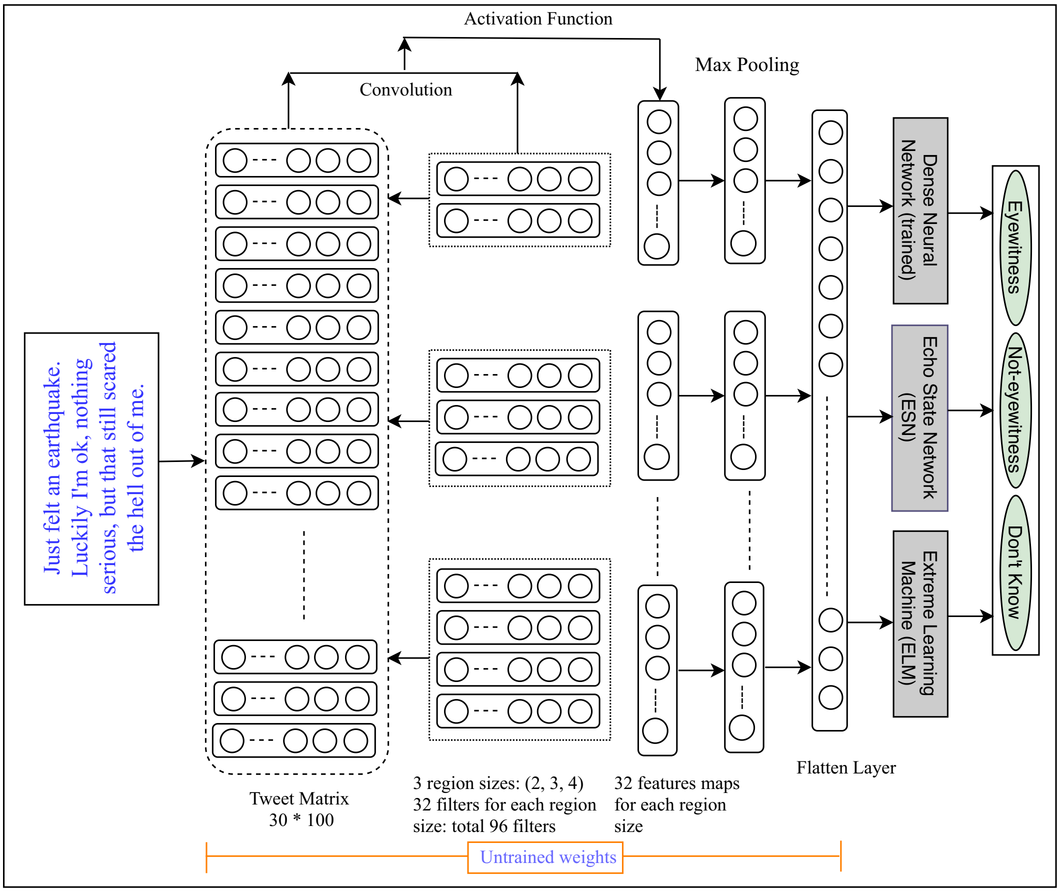

Fig. 1 Flow diagram for the proposed model

fixed-dimensional embedding vector. Embedding vector represents similar word vectors for the words having similar contextual meaning. In our case, we used a pre-trained GloVe[2] word embedding vector to map each word into a 100-dimensional embedding vector. The use of pre-trained GloVe embedding reduces the computation overhead and generally leads to better results because it is trained on a large corpus of texts [9]. The tweet matrix T_i can be represented by (1).

$$T_i = \begin{array}{c} \begin{matrix} W_1 & W_2 & W_3 & \ldots & W_n \end{matrix} \\ \begin{bmatrix} x_{11} & x_{21} & x_{31} & \ldots & x_{n1} \\ x_{12} & x_{22} & x_{32} & \ldots & x_{n2} \\ x_{13} & x_{23} & x_{33} & \ldots & x_{n3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{1k} & x_{2k} & x_{3k} & \ldots & x_{nk} \end{bmatrix} \end{array} \tag{1}$$

[2]http://nlp.stanford.edu/data/glove.twitter.27B.zip

Tweet matrix with n words is represented by T_i. For the tweets having more than n words, we curtail out to make it to n words and for the tweets having less than n words, we padded them to make it of the dimension of n words. In the tweet matrix, $[x_{n1}x_{n2}.....x_{nk}]$ represents the embedding vector for word W_n, where k represented the embedding dimension. In our case, we fixed the length of the tweets (n) to 30 words and

Table 1 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

embedding dimension (k) to 100 for the experiments.

To extract features from the tweet matrix, the convolutional neural network uses various n-gram filters. The convolution process involves a filter with the dimension of $(w \times k)$, where w is the number of the words and k is the word embedding dimension. The filter applied on a tweet matrix first performs the element-wise multiplication with the first w words of the tweet matrix and then the sum of all the multiplied values is then passed through an activation function to get a feature map. This operation is repeated for the next w words of the tweet matrix by leaving the first word. A simple convolution process with a 3-gram filter (i.e., size of filter matrix = $(3 \times k)$) is represented in (2) and (3).

$$T_i = \begin{array}{c} \begin{array}{ccccc} W_1 & W_2 & W_3 & \ldots & W_n \end{array} \\ \begin{bmatrix} x_{11} & x_{21} & x_{31} & \cdots & x_{n1} \\ x_{12} & x_{22} & x_{32} & \cdots & x_{n2} \\ x_{13} & x_{23} & x_{33} & \cdots & x_{n3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{1k} & x_{2k} & x_{3k} & \cdots & x_{nk} \end{bmatrix} \end{array} \bullet \begin{bmatrix} y_{11} & y_{21} & y_{31} \\ y_{12} & y_{22} & y_{32} \\ y_{13} & y_{23} & y_{33} \\ \vdots & \vdots & \vdots \\ y_{1k} & y_{2k} & y_{3k} \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} fm_1 \\ fm_2 \\ fm_3 \\ \vdots \\ fm_{(n-w+1)} \end{bmatrix} = \begin{bmatrix} F\{x_{11}y_{11} + x_{21}y_{21} + x_{31}y_{31} + x_{12}y_{12} + x_{22}y_{22} + x_{32}y_{32} + \cdots + \\ x_{1k}y_{1k} + x_{2k}y_{2k} + x_{3k}y_{3k}\} \\ F\{x_{21}y_{11} + x_{31}y_{21} + x_{41}y_{31} + x_{22}y_{12} + x_{32}y_{22} + x_{42}y_{32} + \cdots + \\ x_{2k}y_{1k} + x_{3k}y_{2k} + x_{4k}y_{3k}\} \\ F\{x_{31}y_{11} + x_{41}y_{21} + x_{51}y_{31} + x_{32}y_{12} + x_{42}y_{22} + x_{52}y_{32} + \cdots + \\ x_{3k}y_{1k} + x_{4k}y_{2k} + x_{5k}y_{3k}\} \\ \vdots \\ F\{x_{(n-2)1}y_{11} + x_{(n-1)1}y_{21} + x_{(n)1}y_{31} + x_{(n-2)2}y_{12} + x_{(n-1)2}y_{22} + \\ x_{(n)2}y_{32} + \cdots + x_{(n-2)k}y_{1k} + x_{(n-1)k}y_{2k} + x_{nk}y_{3k}\} \end{bmatrix} \tag{3}$$

where, $F(x) = max(0, x)$ represents the ReLu activation function which maps negative value to zero. The feature maps obtained by performing the convolutional process is represented by $[fm_1, fm_2, fm_3, \ldots\ldots, fm_{(n-w+1)}]$. This feature map is then passed through a max-pooling operation, that pools the maximum value from a window. A max-pooling operation with the window size of p can be defined by (4).

$$\hat{fm}_1 = max(fm_i, fm_{i+1}, fm_{i+2}, \ldots\ldots, fm_p), i \geq 1 \tag{4}$$

The feature maps obtained from the convolutional and max-pooling operation are then flattened to get the feature vectors. The obtained feature vectors are used by the output layer to predict the probabilities for each of the classes. Recently, Tong et al. [43] and Chang et al. [4] utilized an untrained convolutional neural network to extract features from the given input and achieved state-of-the-art performance to reduce the burden of the backpropagation algorithms. Motivated by their studies [4, 43], in this work we have randomly initialized the weights of the convolutional neural network using *Glorot Uniform*[3] distribution. The *Glorot Uniform* distribution draws a sample from the uniform distribution within $(-GUD, GUD)$, where GUD can be defined by (5):

$$GUD = \sqrt{\frac{6}{\text{Number of input units in the weight tensor} + \text{Number of output units in the weight tensor}}} \tag{5}$$

After initializing weights randomly from the uniform distribution, the CNN network is directly used to extract features from the tweet matrix without any training (see Fig. 1). For extracting the features, we used three CNN layers with 32 filters of 2-gram, 3-gram, and 4-gram and perform the convolutional operation on the (30×100) dimensional tweet matrix. After that, the values are passed through the ReLu activation function and max-pooling operation with a window size of 5. Finally, the obtained feature vector is used by the trainable dense neural network

[3] https://www.tensorflow.org/api_docs/python/tf/keras/initializers/GlorotUniform

(DNN), echo state network (ESN), and extreme learning machine (ELM) to classify tweets into *eyewitness*, *not-eyewitness*, and *don't know* classes, as can be seen in Fig. 1.

3.3 Random Convolutional Neural Network-Dense Neural Network (RCNN-DNN)

In the RCNN-DNN network, the features extracted by the untrained convolutional neural network passed through a dense layer containing 128 neurons followed by an output layer with a softmax activation function. The dense connection of the RCNN-DNN model (i.e., last two layers) is trained using *Adam* [20] backpropagation algorithm. We used *categorical crossentropy* as a loss function, the learning rate of 0.001, batch size of 100 and trained this network for the 100 epochs.

3.4 Random Convolutional Neural Network-Echo State Network (RCNN-ESN)

In the RCNN-ESN network, the features extracted from the untrained convolutional neural network are used by the echo state network (ESN) to predict the probabilities for each of the classes. The Echo State Network (ESN) consists of a network of sparsely connected neuron nodes as a reservoir. A temporal sequential signal drives the reservoir and creates spatiotemporal patterns in high-dimensional space. To produce the outputs, these spatiotemporal patterns are transformed by a linear mapping. Equations 6 and 7 represent the states of the reservoir nodes and those of the output nodes.

$$p_t = (1-\mu).p_{t-1} + \mu.\phi(W_{in}.u_t + W_{rec}.p_{t-1}) \quad (6)$$

$$q_t = W_{out}.p_t \quad (7)$$

where, t represents the time steps, p_t is a vector which represents states of the nodes of reservoir at time t, q_t is the output vector at time t, u_t is the input vector at time t, ϕ represents the element-wise sigmoid activation function, and μ $(0 < \mu \leq 1)$ is the parameter for controlling the speed of the state update. The weight matrices W_{in}, W_{rec}, and W_{out} represent the weight between the input and reservoir, recurrent connection weights between the nodes within the reservoir, and trainable weights between the reservoir and output, respectively. The weight matrices W_{in} and W_{rec} are initialized randomly and fixed during the training process. For the echo state network, the weight matrix W_{out} training can be performed by solving a simple regression problem. To train the weight parameter, the mean square error between the actual out to the reservoir output is minimized using (8).

$$W_{out} = QP^T(PP^T + \lambda.I)^{-1} \quad (8)$$

where I represents the identity matrix, λ represents the regularization factor, P is the matrix of all the states of the reservoir nodes in response to all the training input data, and Q represents the matrix of desired output for all the training data. The detailed description of the echo state network can be seen in [27]. In our case, we have performed extensive experiments by varying the number of reservoir processing units. The model performed best with the 1,000 reservoir processing units.

3.5 Random Convolutional Neural Network-Extreme Learning Machine (RCNN-ELM)

In the RCNN-ELM network, the features extracted from the untrained convolutional neural network are used by the extreme learning machine to classify disaster-related tweets into different classes. Similar to the echo state network, the parameters of hidden nodes and input weights are randomly assigned to the network. The output weight is calculated by a simple generalized inverse operation. The output of the hidden layer of the extreme learning machine can be defined by (9).

$$H_{out} = Activation(W_{in}.x + b) \quad (9)$$

where, W_{in} and b are the input weights, x is the input batch, *Activation* is the activation function, and H_{out} represents the hidden output. The output of the ELM can be represented by $y_{out} = H_{out}.\beta$, where β is the output weights. β can be calculated on the labeled target 'T' using (10).

$$\beta = (H_{out}.H_{out}^t)^{-1}.H_{out}^t.T \quad (10)$$

where $(H_{out}.H_{out}^t)^{-1}.H_{out}^t$ the pseudo-inverse of the matrix H_{out}. The detailed description of the extreme learning machine can be seen in [13]. In our case, we have extensively performed the experimentation by varying the number of hidden layers for the ELM. We achieved the best performance when we used 8,000 hidden layers in the ELM network.

Table 2 Results for each of the models

	Class	Hurricane P	Hurricane R	Hurricane F_1	Earthquake P	Earthquake R	Earthquake F_1	Flood P	Flood R	Flood F_1	Wildfire P	Wildfire R	Wildfire F_1
RCNN-DNN	Eyewitness	0.67	0.52	0.59	0.89	0.88	0.89	0.69	0.56	0.62	0.75	0.47	0.58
	Not-eyewitness	0.82	0.83	0.82	0.92	0.78	0.85	0.82	0.63	0.71	0.87	0.96	0.91
	Don't know	0.82	0.89	0.85	0.80	0.85	0.82	0.82	0.93	0.88	0.86	0.72	0.79
	Weighted Avg.	**0.79**	**0.80**	**0.79**	**0.86**	**0.86**	**0.86**	**0.80**	**0.80**	**0.79**	**0.86**	**0.86**	**0.85**
RCNN-ESN	Eyewitness	0.64	0.37	0.47	0.84	0.88	0.86	0.65	0.35	0.46	0.69	0.24	0.35
	Not-eyewitness	0.72	0.74	0.73	0.92	0.76	0.83	0.74	0.59	0.66	0.84	0.93	0.88
	Don't know	0.75	0.88	0.81	0.76	0.76	0.76	0.78	0.94	0.85	0.70	0.66	0.68
	Weighted Avg.	0.72	0.73	0.71	0.82	0.82	0.82	0.74	0.76	0.73	0.79	0.80	0.78
RCNN-ELM	Eyewitness	0.52	0.50	0.51	0.85	0.84	0.85	0.59	0.51	0.55	0.78	0.47	0.59
	Not-eyewitness	0.76	0.77	0.76	0.95	0.80	0.87	0.72	0.67	0.70	0.85	0.94	0.89
	Don't know	0.79	0.80	0.80	0.75	0.80	0.77	0.82	0.88	0.85	0.77	0.68	0.72
	Weighted Avg.	0.73	0.73	0.73	0.83	0.82	0.82	0.76	0.77	0.76	0.83	0.83	0.82
DNN	Eyewitness	0.65	0.51	0.57	0.83	0.85	0.84	0.52	0.42	0.45	0.54	0.44	0.49
	Not-eyewitness	0.80	0.81	0.80	0.73	0.75	0.74	0.67	0.72	0.74	0.84	0.89	0.86
	Don't know	0.81	0.87	0.84	0.73	0.71	0.72	0.79	0.89	0.84	0.71	0.67	0.69
	Weighted Avg.	0.77	0.78	0.77	0.78	0.79	0.79	0.71	0.75	0.73	0.78	0.80	0.79
CNN	Eyewitness	0.66	0.51	0.58	0.88	0.87	0.87	0.67	0.54	0.60	0.73	0.45	0.56
	Not-eyewitness	0.82	0.83	0.82	0.90	0.79	0.84	0.81	0.61	0.70	0.84	0.93	0.88
	Don't know	0.81	0.87	0.86	0.78	0.87	0.82	0.80	0.91	0.85	0.83	0.70	0.76
	Weighted Avg.	0.78	0.78	0.78	0.85	0.86	0.85	0.78	0.78	0.77	0.83	0.83	0.82
KNN	Eyewitness	0.59	0.48	0.53	0.72	0.95	0.82	0.53	0.35	0.42	0.37	0.42	0.40
	Not-eyewitness	0.77	0.72	0.74	0.81	0.77	0.79	0.75	0.55	0.63	0.78	0.93	0.84
	Don't know	0.77	0.87	0.81	0.86	0.47	0.61	0.78	0.93	0.85	0.78	0.31	0.44
	Weighted Avg.	0.73	0.74	0.73	0.78	0.76	0.74	0.72	0.74	0.72	0.74	0.73	0.71
Naive Bayes	Eyewitness	0.30	0.47	0.36	0.75	0.69	0.72	0.38	0.50	0.43	0.37	0.58	0.45
	Not-eyewitness	0.63	0.62	0.63	0.46	0.78	0.58	0.62	0.55	0.58	0.83	0.81	0.82
	Don't know	0.73	0.56	0.63	0.72	0.64	0.68	0.78	0.73	0.75	0.47	0.40	0.43
	Weighted Avg.	0.61	0.56	0.58	0.71	0.69	0.69	0.67	0.65	0.66	0.70	0.69	0.69
Decision Tree	Eyewitness	0.51	0.46	0.48	0.81	0.83	0.82	0.50	0.38	0.43	0.50	0.34	0.41
	Not-eyewitness	0.73	0.77	0.75	0.72	0.74	0.73	0.66	0.60	0.63	0.82	0.88	0.85
	Don't know	0.80	0.81	0.81	0.73	0.69	0.71	0.78	0.87	0.82	0.67	0.61	0.64
	Weighted Avg.	0.72	0.72	0.72	0.77	0.77	0.77	0.70	0.72	0.71	0.75	0.77	0.76

Bold indicates "Best performance"

Along with three different random neural network-based models such as RCNN-DNN, RCNN-ESN, and RCNN-ELM, we implemented K-nearest neighbor, Naive Bayes, Decision tree, fully trained 3-layered DNN (DNN), and fully-trained 2-layered CNN (CNN) model to compare the performance of the proposed random neural network-based models. We extracted TF-IDF (Term Frequency and Inverse Document Frequency) features from the tweets and used it as an input to the K-nearest neighbor, Naive Bayes, Decision tree, and DNN models to classify tweets into *eyewitness*, *not-eyewitness*, and *Don't know* classes.

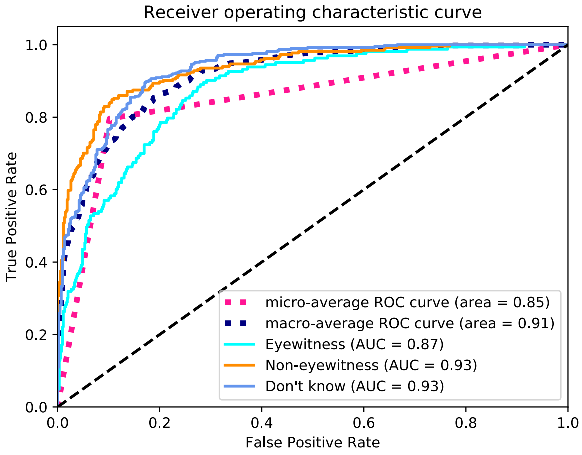

Fig. 2 ROC curve for RCNN-DNN for hurricane event

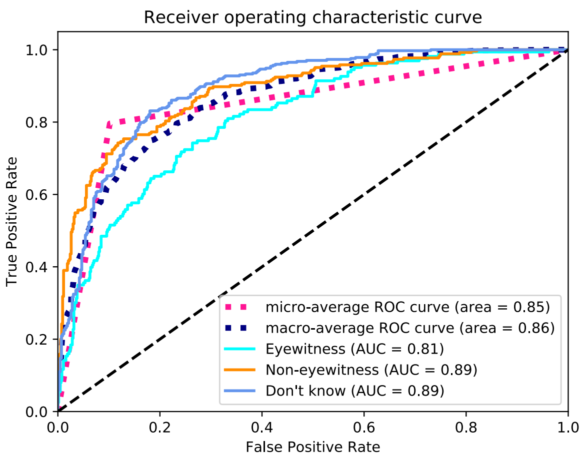

Fig. 3 ROC curve for RCNN-ESN for hurricane event

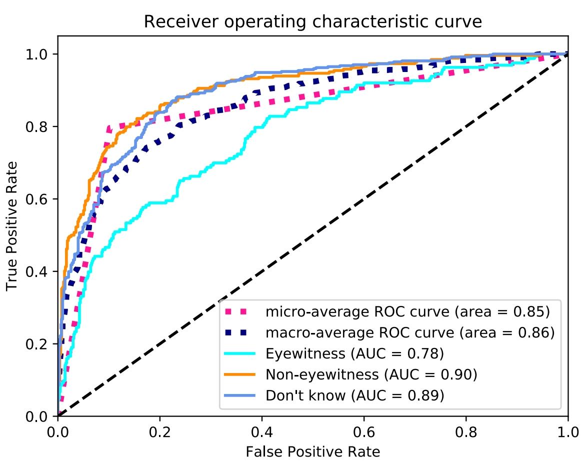

Fig. 4 ROC curve for RCNN-ELM for hurricane event

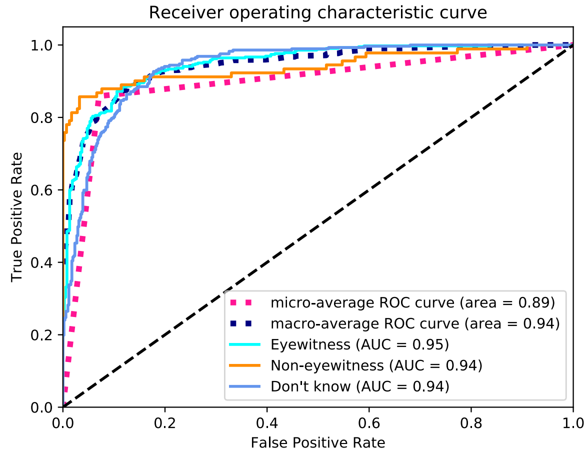

Fig. 5 ROC curve for RCNN-DNN for earthquake event

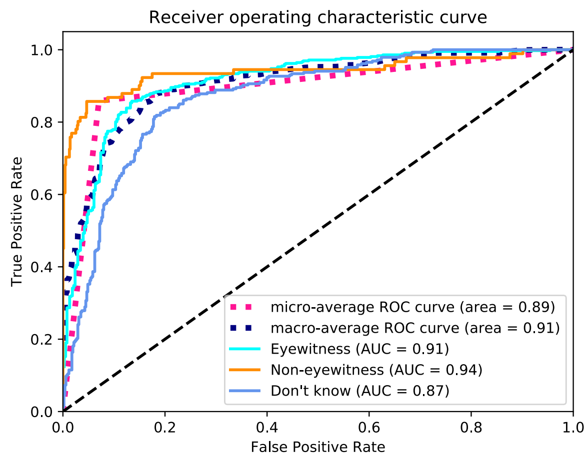

Fig. 6 ROC curve for RCNN-ESN for earthquake event

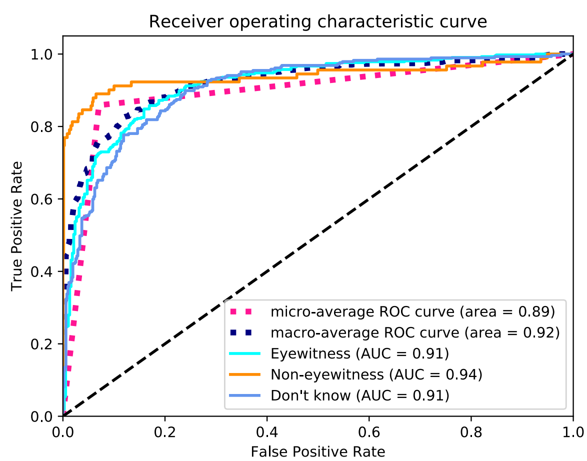

Fig. 7 ROC curve for RCNN-ELM for earthquake event

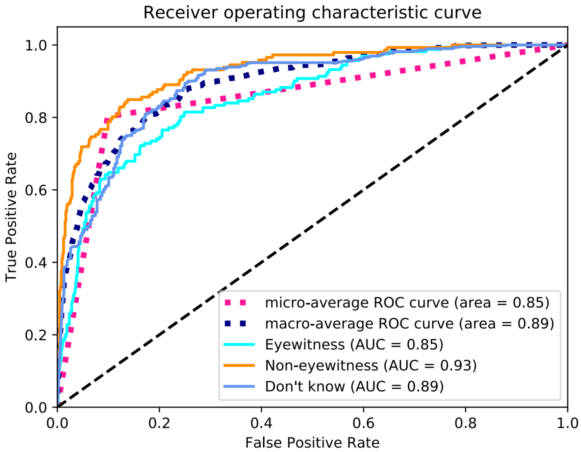

Fig. 8 ROC curve for RCNN-DNN for flood event

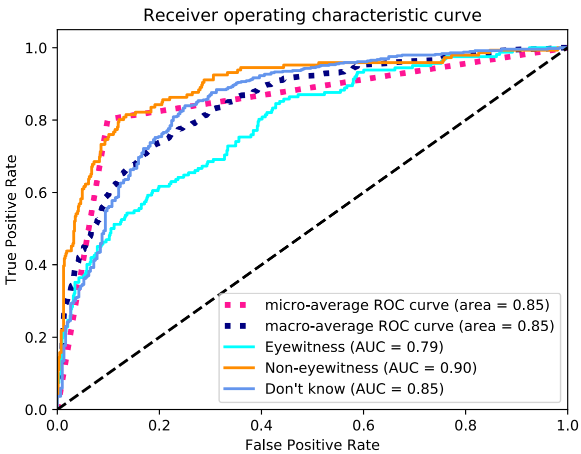

Fig. 9 ROC curve for RCNN-ESN for flood event

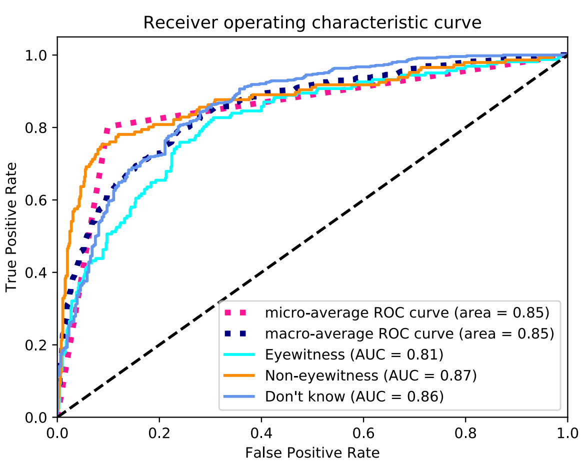

Fig. 10 ROC curve for RCNN-ELM for flood event

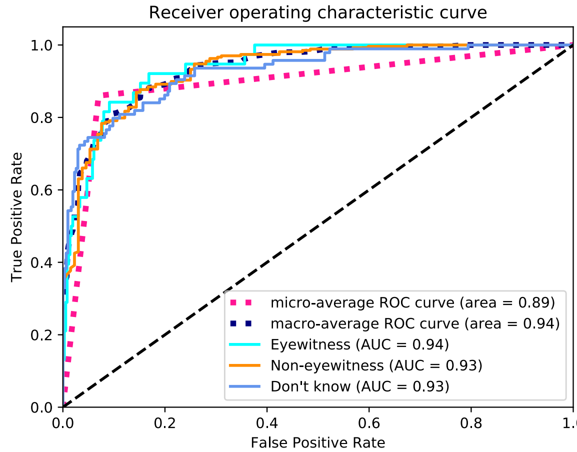

Fig. 11 ROC curve for RCNN-DNN for wildfire event

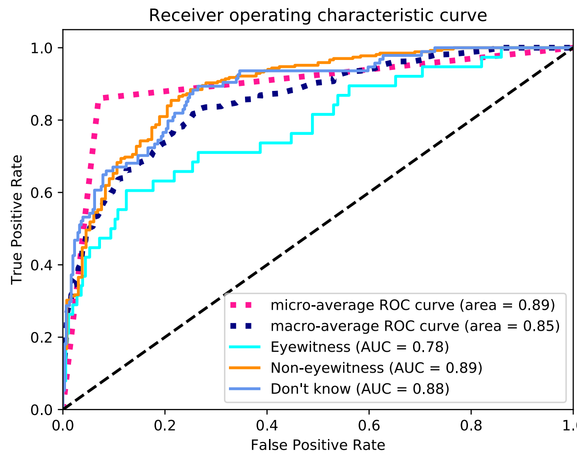

Fig. 12 ROC curve for RCNN-ESN for wildfire event

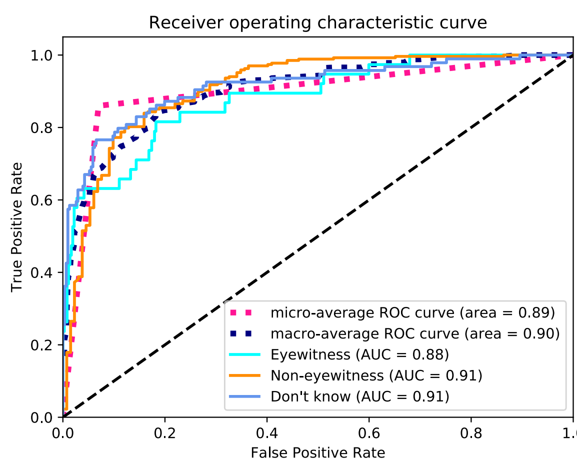

Fig. 13 ROC curve for RCNN-ELM for wildfire event

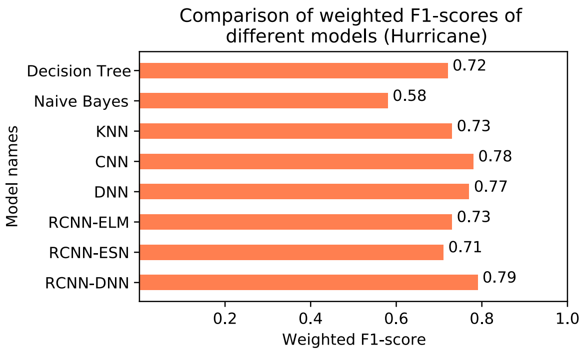

Fig. 14 Models comparison for hurricane event

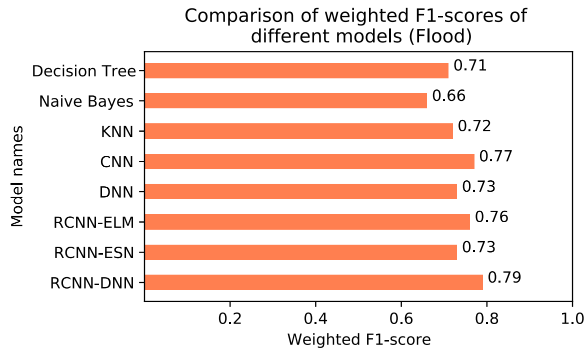

Fig. 16 Models comparison for flood event

DNN model contains 512, 256, and 3 neurons at the first, second, and third layers, respectively. In the case of CNN, the tweet matrix is used as an input to the model. The CNN model contains 32 filters of 2-gram, 3-gram, and 4-gram filters at the first CNN layer whereas at the second CNN layer, 16 filters of 2-gram, 3-gram, and 4-gram filters were used. To train DNN and CNN models, *Adam* as the optimizer, *categorical crossentropy* as a loss function, a learning rate 0.001, a batch size of 100, and epoch of 100 was used.

4 Results

The performance of the proposed models is evaluated using precision, recall, F_1-score, and Receiver Operating Characteristic (ROC) curve. The detailed description of each of the evaluation metrics can be seen in [6]. The extensive experiments were performed with the hurricane, flood, earthquake, and wildfire disaster-related datasets. We performed our experimentation with eight different models: (i) RCNN-DNN, (ii) RCNN-ESN, (iii) RCNN-ELM, (iv) K-nearest neighbor, (v) Naive Bayes, (vi) Decision tree, (vii) Dense Neural Network (DNN), and (viii) Convolutional Neural Network (CNN).

The performance of each of the models with four different disaster datasets is listed in Table 2. The ROC curves for all the three random neural network-based models, RCNN-DNN, RCNN-ESN, RCNN-ELM for hurricane event can be seen in Figs. 2, 3, and 4, respectively. For the earthquake event, ROC curves for RCNN-DNN, RCNN-ESN, RCNN-ELM can be seen in Figs. 5, 6, and 7, respectively. For the flood event, ROC curves for RCNN-DNN, RCNN-ESN, RCNN-ELM can be seen in Figs. 8, 9, and 10, respectively. Similarly, for wildfire event, ROC curves for RCNN-DNN, RCNN-ESN, RCNN-ELM can be seen in Figs. 11, 12, and 13, respectively. For hurricane event, RCNN-DNN, RCNN-ESN, RCNN-ELM, DNN, CNN, KNN, Naive Bayes, and Decision tree achieved a weighted F_1-scores of 0.79, 0.71, 0.73, 0.77, 0.78, 0.73, 0.58, and 0.72, respectively. The comparison of all the implemented models for a hurricane

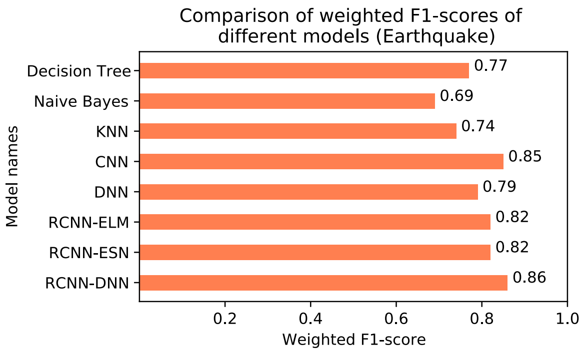

Fig. 15 Models comparison for earthquake event

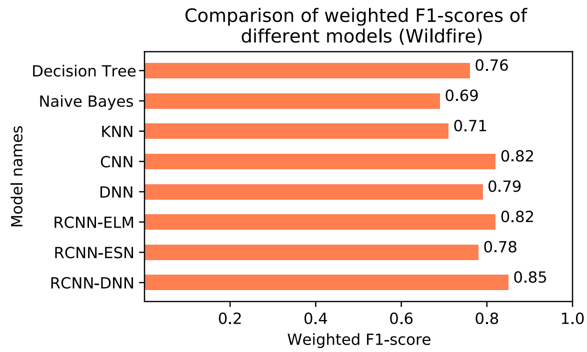

Fig. 17 Models comparison for wildfire event

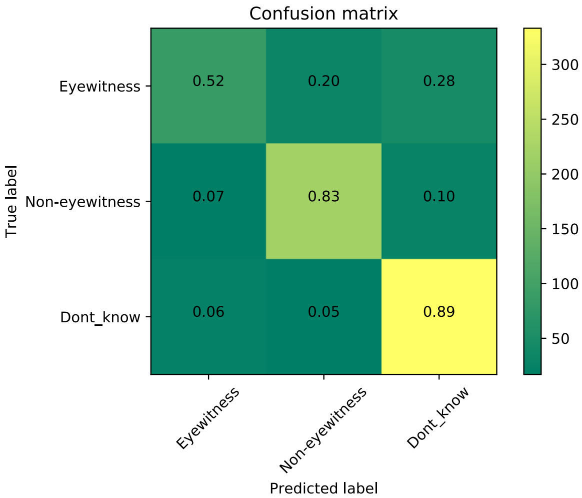

Fig. 18 Confusion matrix for RCNN-DNN for hurricane event

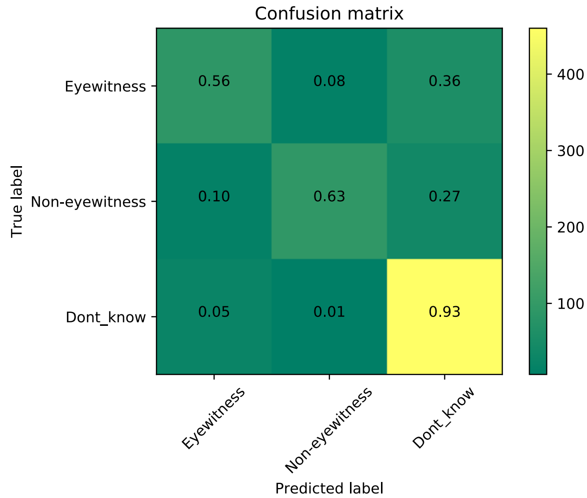

Fig. 20 Confusion matrix for RCNN-DNN for flood event

event in terms of weighted F_1-score is plotted in Fig. 14. For earthquake event, RCNN-DNN, RCNN-ESN, RCNN-ELM, DNN, CNN, KNN, Naive Bayes, and Decision tree achieved a weighted F_1-scores of 0.86, 0.82, 0.82, 0.79, 0.85, 0.74, 0.69, and 0.77, respectively. The comparison of all the implemented models for an earthquake event in terms of weighted F_1-score is plotted in Fig. 15. For flood event, RCNN-DNN, RCNN-ESN, RCNN-ELM, DNN, CNN, KNN, Naive Bayes, and Decision tree achieved a weighted F_1-scores of 0.79, 0.73, 0.76, 0.73, 0.77, 0.72, 0.66, and 0.71, respectively. The comparison of all the implemented models for a flood event in terms of weighted F_1-score is plotted in Fig. 16. For wildfire event, RCNN-DNN, RCNN-ESN, RCNN-ELM, DNN, CNN, KNN, Naive Bayes, and Decision tree achieved a weighted F_1-scores of 0.85, 0.78, 0.82, 0.79, 0.82, 0.71, 0.69, and 0.76, respectively. The comparison of all the implemented models for the wildfire event dataset in terms of weighted F_1-score is plotted in Fig. 17. Among all the implemented models, RCNN-DNN outperformed other models in all the disaster-related datasets. We have plotted the confusion matrix for the best performed RCNN-DNN model in Figs. 18, 19, 20, and 21 for hurricane, earthquake, flood, and wildfire events, respectively.

Fig. 19 Confusion matrix for RCNN-DNN for earthquake event

Fig. 21 Confusion matrix for RCNN-DNN for wildfire event

5 Discussion

The major finding of this work is that the randomly initialized untrained weights with a convolutional neural network (RCNN) extract enough expressive features from the disaster-related tweets to identify eyewitness tweets. The extracted features from the RCNN network with a one-layered trainable dense network performed best throughout all the event datasets. The extracted features from the RCNN with echo state network (ESN) and extreme learning machine (ELM) performed comparatively better than the conventional machine learning classifiers such as KNN, Naive Bayes, and Decision tree. The confusion matrix of the best performed RCNN-DNN model for the hurricane event can be seen from Fig. 18, which shows recall values of 0.52, 0.83, and 0.89 for eyewitness, not-eyewitness, and don't know classes, respectively. The confusion matrix of the best performed RCNN-DNN model for the earthquake event can be seen from Fig. 19, which shows recall values of 0.88, 0.78, and 0.85 for eyewitness, not-eyewitness, and don't know classes, respectively. Similarly, the confusion matrix for flood and wildfire events can be seen from Figs. 20 and 21, respectively. The confusion matrix for the flood event shows recall values of 0.56, 0.63, and 0.93 for eyewitness, not-eyewitness, and don't know classes, respectively. The confusion matrix for the wildfire event shows recall values of 0.42, 0.96, and 0.72 for eyewitness, not-eyewitness, and don't know classes, respectively. As our target is to identify disaster-related eyewitness tweets, the recall for the eyewitness class in the case of the RCNN-ELM model achieved comparable results in comparison to the RCNN-DNN model as can be seen from Table 2. But, RCNN-ESN models fail to perform well for the eyewitness class as can be seen from Table 2. It means the features extracted from the randomly initialized untrained weights from the convolutional neural network, RCNN-DNN and RCNN-ELM learned better than the RCNN-ESN.

In identifying eyewitness tweets, the proposed RCNN-DNN network only uses textual content from disaster-related social media posts, so this framework can be easily incorporated with any social media website. An android application can also be made to recognize eyewitness disaster-related tweets from the live streaming of Twitter posts with the proposed RCNN-DNN model to help people become more aware of the situation during the disaster. The proposed framework can be used effectively in a wide variety of applications, such as better event identification, reliable information detection, and in journalism, to find relevant first-hand information. The limitation of this work is that the proposed system is trained with English-language tweets only, but during the crisis, a significant number of tweets are also posted in other regional languages. In such situations, the proposed model is yet to be evaluated.

6 Conclusion

The identification of crisis-related social media eyewitness posts provides the floor reality of the disaster. In this work, we have extracted textual features from the tweets using randomly initialized untrained weights for the convolutional neural network (RCNN) and proposed three different random neural network-based models such as RCNN-DNN, RCNN-ESN, and RCNN-ELM to identify disaster-related eyewitness tweets. The proposed models were validated with four different disaster-related datasets such as hurricane, earthquake, flood, and wildfire. The RCNN-DNN model outperformed RCNN-ESN, RCNN-ELM, and other conventional machine learning and deep learning models such as KNN, Naive Bayes, Decision tree, DNN, and CNN. The proposed RCNN-DNN model achieved a weighted F_1-scores of 0.79, 0.86, 0.79, and 0.85 for hurricane, earthquake, flood, and wildfire events, respectively.

Other random neural network-based models such as random vector functional link (RVFL) and liquid state networks (LSN) can also be explored in the future for the eyewitness identification task. In the current work, we did not verify the authenticity of the posted tweets in the proposed work, which is another limitation of the proposed work. Therefore, a method can be used to analyze the authenticity of the posted tweets in the future and a multi-lingual framework can be developed to solve the multi-linguality issue. The proposed RCNN-DNN network uses only textual contents of the tweets to identify eyewitness messages. Therefore, in the future, the other clues such as images, videos, and meta-data information of the social media post can be integrated to extract features from the RCNN to investigate the roles of other clues in identifying eyewitness social media messages.

References

1. Al Khushi, N., Côté, A.: Apparent life-threatening events: assessment, risks, reality. Paediatr. Respir. Rev. **12**(2), 124–132 (2011)
2. Bianchi, F.M., Scardapane, S., Løkse, S., Jenssen, R.: Reservoir computing approaches for representation and classification of multivariate time series. IEEE Transactions on Neural Networks and Learning Systems, 1–11. https://doi.org/10.1109/TNNLS.2020.3001377 (2020)
3. Cao, J., Zhang, K., Luo, M., Yin, C., Lai, X.: Extreme learning machine and adaptive sparse representation for image classification. Neural Netw. **81**, 91–102 (2016)
4. Chang, H., Futagami, K.: Convolutional reservoir computing for world models. arXiv:1907.08040 (2019)
5. Darabian, H., Homayounoot, S., Dehghantanha, A., Hashemi, S., Karimipour, H., Parizi, R.M., Choo, K.K.R.: Detecting cryptomining malware: a deep learning approach for static and dynamic analysis. J. Grid. Comput., 1–11 (2020)
6. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: Proceedings of the 23rd International Conference on Machine Learning, 233–240 (2006)
7. Doggett, E., Cantarero, A.: Identifying eyewitness newsworthy events on twitter. In: Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media, 7–13 (2016)
8. Fang, R., Nourbakhsh, A., Liu, X., Shah, S., Li, Q.: Witness identification in twitter. In: Proceedings of the Fourth International Workshop on Natural Language Processing for Social Media, 65–73 (2016)
9. Goldberg, Y.: A primer on neural network models for natural language processing. J. Artif. Intell. Res. **57**, 345–420 (2016)
10. Gupta, A., Sahu, H., Nanecha, N., Kumar, P., Roy, P.P., Chang, V.: Enhancing text using emotion detected from eeg signals. J. Grid. Comput. **17**(2), 325–340 (2019)
11. Huang, C.F.: Evaluation of system reliability for a stochastic delivery-flow distribution network with inventory. Ann. Oper. Res. **277**(1), 33–45 (2019)
12. Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multiclass classification. IEEE Transactions on Systems, Man, and Cybernetics Part B (Cybernetics) **42**(2), 513–529 (2011)
13. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing **70**(1-3), 489–501 (2006)
14. Imran, M., Castillo, C., Diaz, F., Vieweg, S.: Processing social media messages in mass emergency: a survey. ACM Computing Surveys (CSUR) **47**(4), 1–38 (2015)
15. Imran, M., Ofli, F., Caragea, D., Torralba, A.: Using ai and social media multimodal content for disaster response and management: Opportunities, challenges, and future directions Information Processing & Management 57(5). https://doi.org/10.1016/j.ipm.2020.102261 (2020)
16. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. Bonn, Germany:, German National Research Center for Information Technology GMD Technical Report **148**(34), 13 (2001)
17. Jayawardene, I., Venayagamoorthy, G.K.: Comparison of Echo State Network and Extreme Learning Machine for Pv Power Prediction. In: 2014 IEEE Symposium on Computational Intelligence Applications in Smart Grid (CIASG), 1–8 (2014)
18. Katuwal, R., Suganthan, P.N., Tanveer, M.: Random vector functional link neural network based ensemble deep learning. arXiv:1907.00350 (2019)
19. Katuwal, R., Suganthan, P.N., Zhang, L.: An ensemble of decision trees with random vector functional link networks for multi-class classification. Appl. Soft Comput. **70**, 1146–1153 (2018)
20. Kingma, D.P., Ba, J.: Adam:, A method for stochastic optimization. arXiv:1412.6980 (2014)
21. Kumar, A., Singh, J.P.: Location reference identification from tweets during emergencies: a deep learning approach. Int. J. Disaster Risk Reduction **33**, 365–375 (2019)
22. Kumar, A., Singh, J.P., Dwivedi, Y.K., Rana, N.P.: A deep multi-modal neural network for informative twitter content classification during emergencies. Annals of Operations Research 1–32. https://doi.org/10.1007/s10479-020-03514-x (2020)
23. Kumar, A., Singh, J.P., Saumya, S.: A Comparative Analysis of Machine Learning Techniques for Disaster-Related Tweet Classification. IEEE Region 10 Humanitarian Technology Conference, 222–227 (2019)
24. Li, D., Han, M., Wang, J.: Chaotic time series prediction based on a novel robust echo state network. IEEE Transactions on Neural Networks and Learning Systems **23**(5), 787–799 (2012)
25. Liu, D., Chen, L., Wang, Z., Diao, G.: Speech expression multimodal emotion recognition based on deep belief network. J. Grid. Comput. **19**(2), 1–13 (2021)
26. Loyola-González, O., Medina-Pérez, M.A., Choo, K.K.R.: A review of supervised classification based on contrast patterns: Applications, trends, and challenges. J. Grid. Comput., 1–49 (2020)
27. Lukoševičius, M.: A Practical guide to applying echo state networks, 659–686. Springer, Berlin (2012)
28. Ma, Q., Shen, L., Chen, W., Wang, J., Wei, J., Yu, Z.: Functional echo state network for time series classification. Inform. Sci. **373**, 1–20 (2016)
29. Morstatter, F., Lubold, N., Pon-Barry, H., Pfeffer, J., Liu, H.: Finding eyewitness tweets during crises. In: Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science, 23-27. Association for Computational Linguistics, Baltimore, MD, USA (2014). https://doi.org/10.3115/v1/W14-2509
30. Nilsang, S., Yuangyai, C., Cheng, C.Y., Janjarassuk, U.: Locating an ambulance base by using social media: a case study in bangkok. Ann. Oper. Res. **283**(1), 497–516 (2019)
31. Pekar, V., Binner, J., Najafi, H., Hale, C., Schmidt, V.: Early detection of heterogeneous disaster events using social media. J. Assoc. Inf. Sci. Technol. **71**(1), 43–54 (2020)
32. Qiu, X., Suganthan, P.N., Amaratunga, G.A.: Ensemble incremental learning random vector functional link network for short-term electric load forecasting. Knowl.-Based Syst. **145**, 182–196 (2018)
33. Qu, B.Y., Lang, B., Liang, J.J., Qin, A.K., Crisalle, O.D.: Two-hidden-layer extreme learning machine for regression and classification. Neurocomputing **175**, 826–834 (2016)
34. Saeed, Z., Abbasi, R.A., Maqbool, O., Sadaf, A., Razzak, I., Daud, A., Aljohani, N.R., Xu, G.: What's happening

around the world? a survey and framework on event detection techniques on twitter. J. Grid. Comput. **17**(2), 279–312 (2019)
35. Singh, J.P., Dwivedi, Y.K., Rana, N.P., Kumar, A., Kapoor, K.K.: Event classification and location prediction from tweets during disasters. Ann. Oper. Res. **283**(1), 737–757 (2019)
36. Stefan, I., Rebedea, T., Caragea, D.: Classification of Eyewitness Tweets in Emergency Situations. In: RoCHI, 46–52 (2019)
37. Subasi, A., Khateeb, K., Brahimi, T., Sarirete, A.: Human Activity Recognition Using Machine Learning Methods in a Smart Healthcare Environment. In: Innovation in Health Informatics, 123–144. Elsevier (2020)
38. Tanev, H., Zavarella, V., Steinberger, J.: Monitoring Disaster Impact: Detecting Micro-Events and Eyewitness Reports in Mainstream and Social Media. In: ISCRAM (2017)
39. Tang, J., Deng, C., Huang, G.B.: Extreme learning machine for multilayer perceptron. IEEE Trans. Neural Netw. Learn. Syst. **27**(4), 809–821 (2015)
40. Tanisaro, P., Heidemann, G.: Time series classification using time warping invariant echo state networks. In: 2016 15Th IEEE International Conference on Machine Learning and Applications (ICMLA), 831–836. IEEE (2016)
41. Timotheou, S.: The random neural network: a survey. Comput. J. **53**(3), 251–267 (2010)
42. Tong, M.H., Bickett, A.D., Christiansen, E.M., Cottrell, G.W.: Learning grammatical structure with echo state networks. Neural Netw. **20**(3), 424–432 (2007)
43. Tong, Z., Tanaka, G.: Reservoir Computing with Untrained Convolutional Neural Networks for Image Recognition. In: 2018 24Th International Conference on Pattern Recognition (ICPR), 1289–1294. IEEE (2018)
44. Truelove, M., Khoshelham, K., McLean, S., Winter, S., Vasardani, M.: Identifying witness accounts from social media using imagery. ISPRS International Journal of Geo-Information **6**(4), 120 (2017)
45. Truelove, M., Vasardani, M., Winter, S.: Towards credibility of micro-blogs: characterising witness accounts. GeoJournal **80**(3), 339–359 (2015)
46. Wamba, S.F., Edwards, A., Akter, S.: Social media adoption and use for improved emergency services operations: the case of the nsw ses. Ann. Oper. Res. **283**(1–2), 225–245 (2019)
47. Yin, Y.: Deep learning with the random neural network and its applications. arXiv:1810.08653 (2018)
48. Yin, Y.: Random neural network methods and deep learning. Probability in the Engineering and Informational Sciences, 1–31. https://doi.org/10.1017/S026996481800058X (2019)
49. Zahra, K., Imran, M., Ostermann, F.O.: Automatic identification of eyewitness messages on twitter during disasters. Inf. Process. Manag. **57**(1), 102107 (2020)
50. Zahra, K., Imran, M., Ostermann, F.O., Boersma, K., Tomaszewski, B.: Understanding eyewitness reports on twitter during disasters. In: Proceedings of the of the ISCRAM (2018) (2018)
51. Zhang, L., Suganthan, P.N.: A comprehensive evaluation of random vector functional link networks. Inf. Sci. **367**, 1094–1105 (2016)
52. Zhang, L., Suganthan, P.N.: A survey of randomized algorithms for training neural networks. Infor. Sci. **364**, 146–155 (2016)
53. Zola, P., Ragno, C., Cortez, P.: A google trends spatial clustering approach for a worldwide twitter user geolocation. Inf. Process. Manag. **57**(6), 102312 (2020). https://doi.org/10.1016/j.ipm.2020.102312