Heliyon 9 (2023) e21172

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Hybrid deep learning model with enhanced sunflower optimization for flood and earthquake detection

Phalguna Krishna E S [a], Venkata Nagaraju Thatha [b], Gowtham Mamidisetti [c], Srihari Varma Mantena [d], Phanikanth Chintamaneni [e], Ramesh Vatambeti [f,*]

[a] Department of Computer Science and Engineering, GITAM School of Technology, GITAM Deemed to Be University, Bengaluru Campus, India
[b] Department of Information Technology, MLR Institute of Technology, Hyderabad, India
[c] Department of CSE, Malla Reddy University, Hyderabad, India
[d] Department of Computer Science and Engineering, SRKR Engineering College, Bhimavaram, 534204, India
[e] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, 522302, India
[f] School of Computer Science and Engineering, VIT-AP University, Vijayawada, India

ARTICLE INFO

Keywords:
Enhanced sunflower optimisation
Hybrid deep learning
Disasters identification
Flood and earthquake
Solar energy
Warning and alert system

ABSTRACT

Natural catastrophes may strike anywhere at any moment and cause widespread destruction. Most people do not have the necessary catastrophe preparedness knowledge or awareness. The combination of a flood and an earthquake can cause widespread destruction. Natural catastrophes have a domino effect on a country's economy, first by damaging infrastructure and then by taking human lives and other resources. The mortality tolls of both humans and animals have decreased as a result of recent natural disasters. So, we need a mechanism to identify and monitor floods and earthquakes. The suggested system uses a hybrid deep learning analysis to keep an eye on earthquake- and flood-affected areas. In order to boost the efficiency of the presented model, this research presents the improved sunflower optimisation (ESFO). In polynomial time, it determines the best time to schedule events. In view of the need for real-time monitoring of regions vulnerable to flooding and earthquakes, as well as the associated costs and precautions, this study focuses on systems. The suggested technology also sends a notification to the proper authorities whenever a person is detected in the area. In the event of an emergency, it can be used as a backup source of solar power. We then offer the best suitable depth and enable real-time earthquake detection with reduced false alarm rates through practical evaluation. Finally, we demonstrate that the projected model can be successfully deployed in a real-world, dynamic situation after being trained on a range of datasets.

1. Introduction

In a short period of time, natural disasters like fires, earthquakes, and floods can cause devastating losses of life and property [1]. Most individuals don't know what to do to prepare for a disaster or how to stay safe once it starts. If there is no suitable plan in place, the situation will worsen, and the effects will be terrible when a catastrophe strikes and the operations of identifying, evacuating, and rescuing people are carried out. Thus, it is essential to develop strategies for handling this issue in order to lessen its impact as much as

* Corresponding author.
E-mail addresses: phalgunakrishna@gmail.com (P.K. E S), nagaraju.thatha@gmail.com (V.N. Thatha), mamidisetti.gowtham@gmail.com (G. Mamidisetti), mhsv@srkrec.ac.in (S.V. Mantena), phanikanth.ch@gmail.com (P. Chintamaneni), v2ramesh634@gmail.com (R. Vatambeti).

https://doi.org/10.1016/j.heliyon.2023.e21172
Received 25 July 2023; Received in revised form 17 October 2023; Accepted 17 October 2023
Available online 18 October 2023

possible [2]. The actions taken by authorities, particularly government officials, at the time can, to varying degrees, mitigate damage from natural disasters like floods and earthquakes. The Internet of Things (IoT) is a relatively new innovation that allows virtually any object to be linked to the web [3]. (IoT) is a technology that is quickly spreading as its application areas expand. One innovation made possible by these technological advancements is monitoring natural disasters. The use of Internet of Things technology in monitoring the occurrence of catastrophes like floods and earthquakes has saved lives [4] (see).

Extreme weather events have reached crisis proportions [5], particularly floods and earthquakes. A warning system is necessary because it can save lives, protect the environment, and prevent economic damage. Recent devastating floods in Kerala killed people, killed animals, and ruined property, all of which had an impact on the state's economy [6–8]. An early warning, detection, and rescue system for natural disasters like floods and earthquakes is one possible solution. Researchers still confront several obstacles, despite the fact that a lot of effort has been made in the past decade towards creating an effective flood and earthquake observatory and warning scheme [8]. A rapid, efficient, simple, and accurate flood and earthquake observatory and warning scheme is needed. Researchers are expressing a lot of interest in the Internet of Things (IoT) as a means of implementing an effective flood and earthquake building and warning scheme with high security [9]. In addition to saving lives and money, this may be used to build an earthquake and flood observatory and warning system.

One popular approach for earthquake detection and identification is the short-time average (STA or LTA) [10]. Nevertheless, STA and LTA have a number of limitations, such as the fact that erroneous initialization values result in a number of false alarms and that they do not work well when a great deal of noise is present [11]. The Akaike information criterion (AIC) is also commonly used as a benchmark method for the earthquake detection issue [12]. When the amount of background noise is large, however, AIC gives inaccurate estimates of the global minimum. A number of machine learning-based automatic earthquake detection methods have been suggested, such as the K-means algorithm, template matching, support vector machine, fuzzy algorithm, wavelet transformation, and thresholding algorithm [13].

In recent years, deep learning has emerged as a useful tool for seismologists. Several different deep learning approaches, including CNNs, RNNs, and autoencoders, have been suggested to detect earthquakes [14]. CNN is quite good at identifying important features in data. There are, however, a number of limitations. The pooling layer, one of the most important layers in CNN design, decreases the network's dimensionality [15]. The network's efficiency is hampered, however, because data may be lost at the pooling layer. CNN's generalisation ability is further limited since it does not take into account the spatial correlations between the retrieved features [16]. The CNN is not only rotationally invariant but also translationally invariant.

In this study, we make a new hybrid model called the attention-based deep residual gated recurrent unit (RGRU-ESFO) that can predict both floods and earthquakes. This paper makes the following primary innovations and contributions: First, we present a series-connected deep learning method that employs an enhanced residual network to extract static data and a GRU to extract additional temporal information. To further improve prediction accuracy, the authors suggest a residual network to be used in conjunction with GRU; they call it RGRU. 2) To further enhance the prediction model's generalisation capabilities, a high-performance ESFO method is used to re-educate the output layer of the projected RGRU model.

Below is how the breakdown of the paper is organised. Associated works are presented in Section 2, and the overall framework and specifics of the suggested hybrid model's implementation are introduced in Section 3. Section 4 presents the trial testing and comparison findings. Section 5 provides a conclusion.

2. Related works

Jackson et al. [17] used a learning strategy on a dataset called FloodNet to compare the effectiveness of SOTA flood picture categorization. We did this by making changes to SOTA models that we had already trained. To learn more about how the uncertainty offset affects how well the models work, we also found a new way to change it. Loss and ROC-AUC were used to measure the models' efficacy as classification tools. This research quantifies the influence of various uncertainty offsets and compares the performance of several CNN architectures for flood picture categorization. These results have the potential to improve disaster response and recovery schemes, which in turn might lessen the devastation caused by natural catastrophes.

Xiang et al. [18] offer a dense pyramid pooling module (DensePPM) that can be used to get information from a distribution. Ablation experiments demonstrated that models trained with the densePPM achieved better values than those trained with the pyramid pooling method (PPM), indicating The DensePPM was subsequently used as the foundation for a new deep learning network, DensePPMUNet-a, which was suggested to separate bodies of water in aerial photographs. Experimental results on many datasets showed that, compared to other state-of-the-art deep learning representations, DensePPMUNet-a performed better.

Hernández et al. [19] have made a lot of progress towards making UAVs that can fly themselves and help in emergencies like floods. To automatically predict the locations most damaged by rainfall, we use a collection of UAV photos related to several floods in Spain and an AI-based strategy that relies on three popular deep neural networks (DNNs) for semantic segmentation of images (flooded areas). To facilitate classification and ensure that only the algorithm output is sent to the cloud for real-time platforms. This makes the process greener and more resilient to connection outages by decreasing reliance on infrastructure and lowering network resource use. Advanced UAV imaging utilising sophisticated DNN-based solutions is conceivable, as demonstrated by experimental findings employing a variety of hardware and architectures.

Yang et al. [20] used deep learning models to create a way for UAV aerial photos to automatically identify flooded structures. As a case study, the Kangshan levee on Poyang Lake was used to see how well the strategy worked. The flood of the focus buildings and vegetation may be determined from the photos with an accuracy of 88 % and 85 %, respectively, according to the experimental results. We could also use the UAV photos and flight data to figure out how much of the structures will be flooded. This research demonstrates

the potential benefit to end users in the flood emergency response sector of accurate and timely visualisation of the geographical delivery of flooding at the object level.

Lenka et al. [21] show two ways to find places that are flooded in UAV aerial photos. In the first method, a texture-based method is used, while a network is trained on the texture data to determine if a picture is flooded or not. This study examines the suggested model's performance in recognising flooded regions across geographical regions, as opposed to prior works where the representations were trained and evaluated on photos from the same geographical locations. Compared to classifiers, the suggested segmentation-based method achieves a higher F1 score of 0.89. The suggested method has been shown to be reliable enough to be used to detect flooded areas in any location with little to no human intervention.

Li and Bortolot [22] made an object-based picture classification to figure out how the amount of impermeable land cover affects the peak flood rate and total flood volume in a given system. With an average accuracy of 92 %, objects measured the temporal shifts on impervious surfaces. Four storm events were used to calibrate and test a drainage model based on stormwater at 0.69 for calibration and 0.46 for validation. From 1950 to 2018, the results reveal an increase in impermeable areas of up to 32 % due to the effects of urban redevelopment. Maximum increases in flooding peak and volume due to land cover imperviousness occurred during storm events with 10- to 100-year return periods. Through the use of remote sensing image categorization, this study's implications hope to inform homeowners and engineers about the likelihood of floods in human-modified landscapes.

Priyatna et al. [23] used multi-Landsat-8 satellite images to investigate land cover change, flood detection, and flood dispersion in the Barito watershed. Sentinel-1 imaging floodplains were identified using a mixture of change detection and the Otsu algorithm. Using techniques from both change detection and machine learning, namely the random forest method, we can identify shifts in LULC. The spread of floodplains was analysed using the overlay method. A total of 109,623 acres were determined to be floodplains in the research region. The change detection approach finds a decline of 13,020 ha (ha) in primary forest, 116,235 ha (ha) in secondary forest, 259 ha (ha) in fields, 146,696 ha (ha) in rice fields, 47,308 ha (ha) in shrubs, and 9601 ha (ha) in ponds. Settlements, barren land, and bodies rise by 218,916 ha and 34,768 ha, respectively. The most flooded ecosystem types included rice paddies, lakes, and virgin woods.

He et al. [24] created the heterogeneous dataset CAU-Flood, which has 18 research plots and underwent careful picture pre-processing and human annotation. In addition, a novel deep network (CNN) was presented for flood detection utilising multispectral and SAR pictures; it was given the name CMCDNet. The projected network is encoder-decoder-based and uses gating and self-attention modules to achieve feature fusion across many stages. In addition, the network fixes the misaligned feature problem at decoding by including an alignment module in the upsampling process. As compared to SOTA approaches, the suggested CMCDNet approach fared better in terms of accuracy, with an IoU of 89.84%.

In order to detect floods with great precision, Zhao et al. [25] created a network called Siam-DWENet. Siam-DWENet is a new way to learn how to switch between tasks because it is based on Siamese architecture. With this method, we can use a small number of training samples to identify flooding based on what we already know. Experiments comparing the efficacy of Siam-DWENet and other approaches for flood detection using two flooding SAR datasets. The experimental findings show that Siam-DWENet achieves an average OA of 0.887 and F1 of 0.865 in flood inundation detection tasks, outperforming previous change detection techniques and improving the accuracy of the inundation area edge while working with complicated backdrops.

3. Proposed system

In this section, we'll present how the research uses two separate datasets—an earthquake dataset and a flood dataset—to make predictions.

3.1. Dataset for earthquake

There are two types of label data in the dataset we used for the suggested DL models. The time series earthquake dataset is one type of information that may be found in the aforementioned [26,27] reports from the National Research Institute for Earth Science and Disaster Prevention (NIED) and the United States Survey. Using the NIED database, 385 earthquakes that happened between April 2009 and May 2019 were chosen. The earthquakes ranged in magnitude from 4 to 8. In addition, 120 stations' worth of data from three earthquakes (Tottori, 2000; magnitude 6.61; Niigata, 2004) were entered into the database. Pre-processing and unit conversion of the NIED earthquake data (g) All earthquake data was sampled at a rate of 100 Hz. East-West (EW) and North-South (NS) are horizontal components, and Up-Down (UD) is a vertical component, of the data shown here.

The second type of information is a several-hour-long time series of data that was not taken during an earthquake but was instead captured by mobile phones. Human activity data and noise data were used in the tests instead of data from actual earthquakes. Information about human actions includes riding bicycles, buses, and cars (in the hand), jumping rope, jogging (hand, pocket), shaking a desk (with a mobile device on top), ascending and descending stairs (bag, hand, pocket), strolling, standing motionless, and working. On the other hand, noise data includes things like machine noises and floor noises (like footsteps) from a variety of heights. As this sensor noise data comes from an uncontrollable external source, we additionally incorporate seismic signal tail data.

For the third dataset, earthquake data from shake table tests gathered with multiple accelerometers with varied HW specifications and prices is used to assess the models' generalisation features. We gathered acceleration data from inexpensive accelerometers that were mounted on the shaking table at Pusan University to record two simulated earthquakes, the Pohang and the El Centro.

3.2. Flood dataset

The case study for this study is the Swat Valley, which is in the Khyber Pakhtunkhwa province of northern Pakistan (Fig. 1). This area is particularly vulnerable to flooding since it is at the confluence of two rivers, the Daral and the Swat. Thirty people lost their lives and another 38 were injured in this area's flash floods last year. More than 130 homes were damaged or destroyed, along with a bridge and a place of worship, due to the flash floods. In addition, flash floods have destroyed or washed away a number of highways, cutting off access to several settlements [28] (see Fig. 2).

Swat is a high-risk area for flooding, according to the Pakistan Disaster Management Authority (PDMA). If another major flood were to occur, this region may become entirely inaccessible, the PDMA warns. This statement was put to the test by the recent flash flooding, and it passed with flying colours, as rescue efforts were significantly slowed down because the area was inaccessible and connecting bridges were destroyed. Because of its importance, this region will serve as a case study for the ongoing investigation.

Multispectral aerial photos were taken of the area for the purpose of creating a complete dataset to address flood hazards in the region. Fig. 1c displays a UAV-captured Red-Green-Blue (RGB) picture of a neighbourhood in Swat. The image's spatial resolution is 0.23 m, making it possible to spot flooded regions with pinpoint accuracy. The image's dimensions are 19,956 b y 12,444, and its field of view is around 11 km 2. Three hundred photographs were taken by the UAV, while the rest were culled from PDMA's local databases, both before and after the flood.

3.2.1. Image acquisition

A tiny UAV, known as River Map, was used for observation and inundation detection. Aerial photographs of the research region were taken using a Go-Pro® digital camera. Fig. 1 depicts the results of real-time monitoring conducted on the case study region on September 2, 2020, using this setup, showing the devastating effects of the flood on this location. Due to the enormous intensity of the floodwater, the photographs indicate that most roads have fallen into the water and many structures have been misshapen, making relief operations more difficult, time-consuming, and maybe impossible.

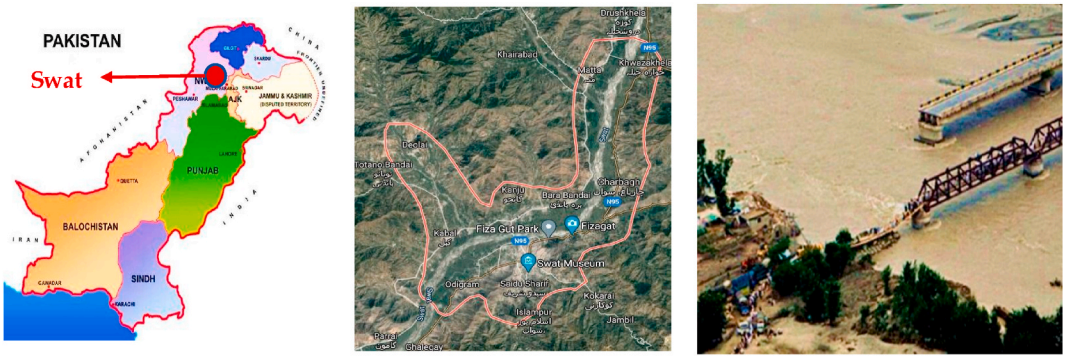
3.3. Pre-processing

In our approach, we scale the features to a range of 0–1 as part of the pre-processing phase. Due to the detrimental effects that unbalanced datasets have on machine learning model performance [29], balancing is essential. In this situation, the dataset that does not include earthquakes (noise and human activity) is substantially bigger. To do this, we employed the K-mean clustering technique [30] to normalise the data that did not involve earthquakes. We employed cluster centroids to represent the non-earthquake data, which were clustered using the K-Mean to account for the total sum of earthquake data points. In addition, we scaled data points to the range of scalers as shadows, which helped to enhance the prediction model:

$$d_{scaled} = \frac{d - d_{min}}{d_{max} - d_{min}} \tag{1}$$

3.4. Prediction stage

In this part of the research, the RGRU hybrid reduction model is shown. This model combines the residual network and GRU. Combining these separate prediction models has two benefits: By combining shallow and deep characteristics, we may: 1) draw attention to the input's internal coupling connection. 2) To go further towards uncovering those elusive temporal nuances. Modelling and implementing each part of RGRU is covered in depth below.



(a) (b) (c)

Fig. 1. (a) Swat Region on a chart of Pakistan; (b) Swat area; (c) aerial copy of a flood-hit zone in Swat.

(a) (b)

Fig. 2. Flooded regions of Swat Valley. (a) A village (b) A damaged bridge.

3.4.1. The residual network

Because the network is made up of cross-layer links, it is possible to improve the accuracy of the residual network by feeding shallow characteristics into deeper layers [31]. In the meantime, the residual network's outstanding feature representation and extraction capacity are shared with the rest of the CNN architecture. Thus, this research employs the residual network structure developed in this work to conduct in-depth feature extraction from data collected using BA. Assuming that the input to the unit is y m and the output is y_{m+1}, we have a residual network built by stacking residual units:

$$y_{m+1} = f(y_m + F(y_m)) \tag{2}$$

People often use the ReLU function, where $F(y_m)$ is the remaining function, which is shown by the residual unit, f(*) is the nonlinear start function, and m is the number of observations.

In this study, a residual network is proposed that is built on top of the residual unit. In this architecture, the activation of each of the three convolutional layers with a 1 × 1 kernel size is always set to ReLU. The small-size filter has the benefit of requiring fewer calculation parameters than larger-size filters like the 33, and it can also facilitate the extraction of more complex abstract characteristics. In addition, in Eq., instead of the ReLU function being used as the activation function for the unit output, the tan function is substituted (3). This function is utilised for two reasons: 1) to solve the zero-slope problem that causes ReLU to die out, and 2) to map the feature map recovered by the residual unit to the range [−1, 1]. In this study, we employ the aforementioned residual unit as the building block for the residual network, with a total of 1 unit.

$$\tanh(x) = \frac{exp^{x} - exp^{-x}}{exp^{x} + exp^{-x}} \tag{3}$$

In this research, the input X is fed into a residual network in order to generate an output X (res out) of the same size. The residual network's output is transformed from three dimensions to two, yielding a sequence Y= (y_1, y_2,.,y_T) of length T, where y t[-1, 1] ((1F)) is a vector with dimensions F. The resulting Y is then sent into a cascading GRU network.

3.4.2. Gated recurrent unit

RNNs have demonstrated exceptional performance in several machine learning models for solving sequence issues. Unfortunately, when training an RNN, you might quickly run into the issues of gradient vanishing and explosion. To address these issues, one RNN variation, the GRU, has a more straightforward architecture than the LSTM. In addition, GRU is more computationally efficient and has fewer training parameters. Hence, GRU is taken into account here to help extract such temporal characteristics. The heart of the GRU system consists of the reset gate and the update gate. When a fresh input is received by a network, the two data sets are fused under the direction of the reset gate r_t. The quantity of data kept in memory is controlled by the update gate z_t. Using the F-dimensional vector y_t at time t in Y as input, this article introduces the comprehensive updating mechanism of the GRU unit.

$$\begin{cases} r_t = \sigma\left(y_t W_{ry} + h_{t-1} W_{rh} + b_r\right) \\ z_t = \sigma\left(y_t W_{zy} + h_{t-1} W_{zh} + b_z\right) \\ \widetilde{h}_t = \tanh\left(y_t W_{hy} + r_t \odot h_{t-1} W_{hh} + b_h\right) \\ h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h}_t \end{cases} \tag{4}$$

The GRU output at time t is denoted by h_t. b is the bias vector, while W is the weight matrix. The sigmoid function is denoted by, and the equivalent dot product operation on the element is 1.

In this study, we construct a two-layer GRU network with 4–8 neurons per layer and apply the ReLU activation function. Next, a regression prediction is constructed using a two-layer FC architecture, with 32 neurons per layer and 10 prediction steps per layer. The RGRU model's final prediction value is derived from the GRU output via regression prediction. The model for forecasting wind power has been developed thus far. Adam is an optimisation technique used in this research to hone the prediction model.

3.4.3. Retraining stage

The popular Adam optimizer has been shown to produce a prediction model with quick early-stage convergence, but a too-low

learning degree in the late stages of training might hinder the model's effective convergence and lead to generalisation issues [32]. In this study, a high-performance ESFO is used to retrain the limits of the yield layer based on the projected model trained by Adam, with the goal of improving the model's generalisation performance. This idea is inspired by the use of population-based optimisation procedures to optimise the parameters of networks [33].

3.4.3.1. The ESFO algorithm for optimal solution. Sunflower optimisation (SFO) [34] is an algorithm that models sunflower behaviour in order to maximise energy absorption from sunlight. The SFO comprises two stages: the pollination phase and the migration phase. During the pollination phase, sunflowers work together to create a fertile pollen gamete. In this stage of the process, the sunflowers are "in motion," and their random steps are directed towards the strongest sunflower (the "sun"). The most critical step in locating the global optimum of an optimisation issue is doing a thorough search of the available solutions. In order to address this problem, we provide an improved version of SFO called ESFO. A new pollination operator has been added to the ESFO, allowing for a more even distribution of exploitative and exploratory capabilities. The best solution is described below using the ESFO algorithm.

Make a person: There were Z sunflowers in the original population.

$$Pop = \{S_1, S_2, .., S_z\} \tag{5}$$

Each sunflower $Si = \{s_{i,1}, s_{i,2}, \ldots, s_{i,M}\}$ in the populace is collected of M statistics, each one denotes the identifier. s_{ij} is defined as

$$s_{i,j} = s_{min} + \varphi_{ij}(s_{max} - s_{min}) \tag{6}$$

s min and s max are the values of $s_{i,j}$, whereas ij is a uniformly random number among 0 and 1. The best solution begins with s min and s max set to 1 and M, respectively. Sunflowers' components are discretized before being used with ESFO.

Estimating Power: Each sunflower's strength, S_i, may be determined using either Eq. (5) or (Eq (6). Power in the context of the task scheduling problem stands for solution cost. After doing the maths on power, the cheapest sunflower (S*) is chosen as the sun.

Standard SFO models pollination by selecting pZ sunflowers at random from the population and then updating them as though they were actually being pollinated.

$$S_i^{t+1} = r_i.\left(S_i^t - S_j^t\right) + S_j^t \tag{7}$$

where p is a pollination index. Sunflower i's and j's current locations, S_i^t and S_j^t, respectively, t. r i is an arbitrary integer between zero and one. The typical SFO uses p = 0.5. The following equation was suggested in ESFO to characterise the pollination stage.

$$S_i^{t+1} = a^t \times A + (1 - a^t) \times B \tag{8}$$

$$A = S_j^t + U(-1 + 1) \times \left(S_i^t - S_j^t\right) \tag{9}$$

$$B = S_i^t + \beta \times \left(S^* - S_i^t\right) \tag{10}$$

where t is the switching chance at iteration t that determines the balance between regional and worldwide pollination. The greatest sunflower variety discovered to yet is S*. Adding additional wiggle room to the search radius, U (−1, +1) produces a uniform random sum in the range [-1,1]. The magnitude of the search direction can be adjusted by adjusting the scaling factor Si, (S*- S_i^t). To get a normal random sum in the range [0, 1], we substituted the value 3 for. Here is how we characterise t.:

$$\alpha^t = a_{max} - \frac{t}{I}(a_{max} - a_{min}) \tag{11}$$

the maximum sum of iterations is I. Maximum and minimum values are 0.6 and 0.4. In this case, the plan is for ESFO to do more comprehensive searches in the early iterations and then narrow their focus to more specific regions as time goes on.

Introducing the new pollination operator was driven mostly by a desire to boost exploration and exploration in tandem. By modelling the local pollination process, Term A in Eq. (9) enhances ESFO utilisation. Adding Term B to Eq. (10) and modelling the global pollination mechanism enhances the investigation.

Movement: The following procedure is used to pick and update (1- p) Z sunflowers:

$$S_i^{t+1} = S_i^t + r.\frac{S^* - S_i^t}{\left\|S^* - S_i^t\right\|} \tag{12}$$

The sunflower Si's directional shift towards the sun S is modelled by Eq. (12). The sun is finally swapped out with the healthiest sunflower.

$$S^* = S_k \left| P(S_k) > P\left(S_j\right) \forall j = 1, 2, \ldots, Z \right. \tag{13}$$

As long as the maximum sum of iterations is not achieved, the algorithm continues to execute.

4. Results and discussion

4.1. Performance metrices

Performance matrices are used to compare various machine learning and deep learning methods. The confusion matrix provides a table of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) and serves as the basis for the classification performance measures (False Negative).

The confusion matrix is used to derive common classification performance metrics including accuracy, precision, and recall. Accuracy is calculated as:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{14}$$

Accuracy is a simple way to judge how well a model works. It is calculated by dividing the number of correct predictions by the number of observations.

Precision is computed as:

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

how well a model performs among those successful predictions is evaluated by looking at its precision. To rephrase, how many true positives may be expected among all predicted positives? A low proportion of false positives indicates high accuracy.

Recall is computed as:

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

The recall is the number of times out of all the earthquake incidences that the model correctly identifies an earthquake.

The F1 score is the harmonic mean of the accuracy and recall scores. It considers both over- and under-reporting.

$$F1 - Score = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \tag{17}$$

Lastly, a receiver operating characteristics (ROC) curve may be used to depict the false-positive and true-positive rates produced by the classification model.

4.2. Evaluation

The test is carried out in both a static and a dynamic setting. Since the sensors in a static setting are always in the same place, there's no need to train a model using data from a wide range of different human actions. Nonetheless, we used examples of human actions like strolling and waiting to train the model appropriately for the static setting. This is due to the fact that the model cannot learn the underlying patterns of the data, notably seismic patterns, if only noisy data is present. We compared models based on a variety of characteristics, and then we put the model that performed best in a static setting through its paces in a dynamic one to gauge how well it would transfer.

The analysis of various models on the earthquake dataset is represented in Table 1. In this analysis, different methodologies are used. In DBN, we achieved an accuracy of 94.24 % and a precision of 84.40. and also, the recall value of 69.11, and finally the F1 score of 75.99. CNN achieved an accuracy of 95.13 and a precision of 89.27, as well as a recall value of 69.75. RNN achieved the accuracy of 96.92 and the precision of 58.42; also, the recall value of 60.69; and finally, the F1 score of 67.76. LSTM achieved the accuracy of 97.97 and the precision of 78.83; also, the recall value of 67.82; and finally, the F1 score of 72.91. RGRU achieved an accuracy of 97.89, a precision of 72.11, a precision of 77.33, and finally the F1 score of 74.63. RGRU-ESFO achieved an accuracy of 98.96, a precision of 93.40, a recall value of 79.90, and finally the F1 score of 86.1 See Figs. [3-9].

The analysis of various models on the flood dataset is represented in Table 2. In this analysis, different methods are used as DBN reached the accuracy score of 92.82 and also reached the precision rate of 92.42, the recall value of 93.75, and finally the F1-score of 93.08. CNN reached the accuracy score of 2.19 and also reached the precision rate of 33.29, the recall value of 93.96, and finally the

Table 1
Analysis of various model on earthquake dataset.

Methodology	Accuracy	Precision	Recall	F1
DBN	94.24	84.40	69.11	75.99
CNN	95.13	89.27	53.76	69.75
RNN	96.92	58.42	60.69	67.76
LSTM	97.97	78.83	67.82	72.91
RGRU	97.89	72.11	77.33	74.63
RGRU-ESFO	98.96	93.40	79.90	86.13

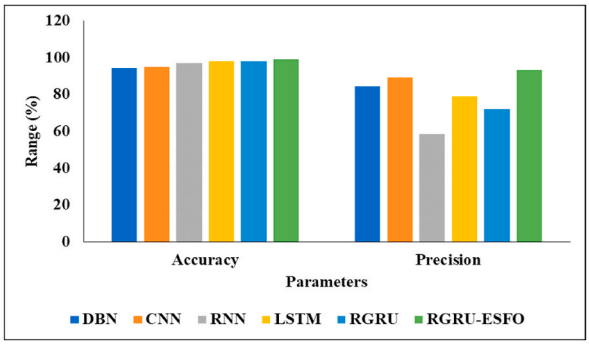


Fig. 3. Graphical description of proposed model.

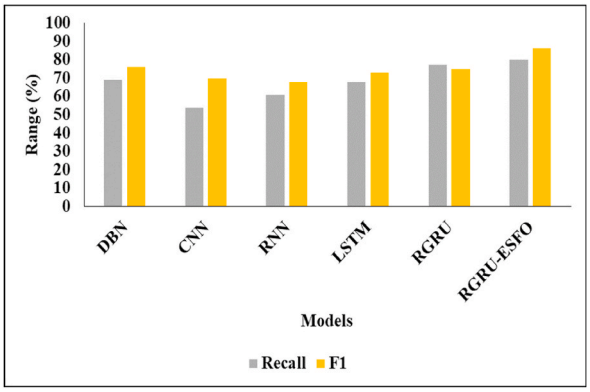


Fig. 4. Analysis of projected model.

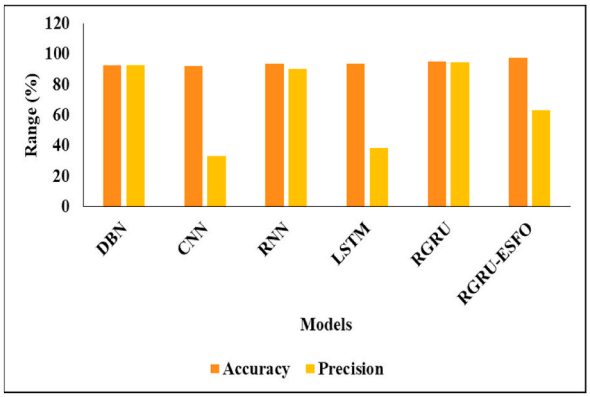


Fig. 5. Flood dataset analysis.

F1-score of 49.16. RNN reached the accuracy score of 93.54 and also reached the precision rate of 90.19, the recall value of 97.47, and finally the F1-score of 93.69. LSTM reached the accuracy score of 93.61 and also reached the precision rate of 38.39, the recall value of 97.43, and finally the F1-score of 55.08. RGRU reached the accuracy score of 94.80 and also reached the precision rate of 94.58, the recall value of 95.05, and finally the F1-score of 94.82. RGRU-ESFO reached the accuracy score of 97.55 and also reached the precision rate of 62.97, the recall value of 94.95, and finally the F1-score of 75.72, respectively.

The comparative analysis of the proposed model on the overall dataset is represented in Table 3 above. In this analysis, different methods are used, as DBN reached the accuracy score of 90.84 and also reached the precision rate of 91.03, as well as the recall value of 88.27, and finally the F1-score of 91.65. CNN reached the accuracy score of 91.60 and also reached the precision rate of 88.14, the recall value of 89.24, and finally the F1-score of 84.47. RNN reached the accuracy score of 91.59 and also reached the precision rate of 93.19, the recall value of 89.45, and finally the F1-score of 91.28. LSTM reached the accuracy score of 85.33 and also reached the

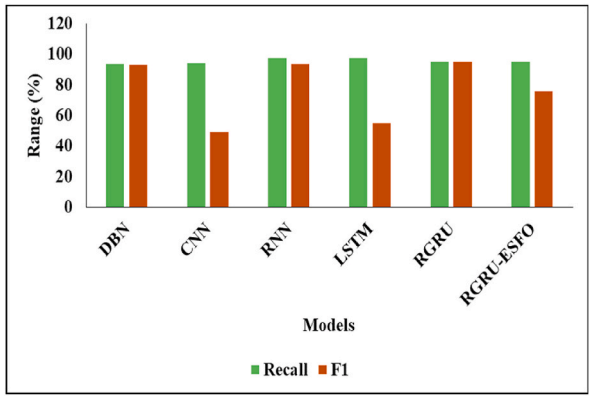


Fig. 6. Analysis of various classifiers.

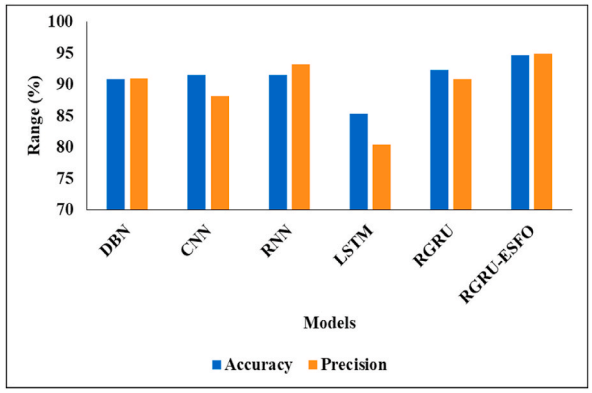


Fig. 7. Graphical Description of Proposed Model on Overall dataset.

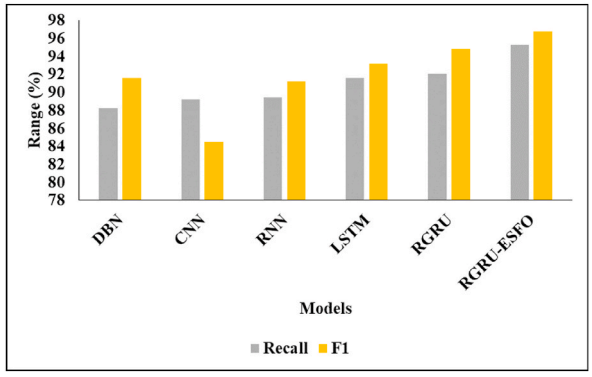


Fig. 8. Analysis of models.

precision rate of 80.44, the recall value of 91.59, and finally the F1-score of 93.18. RGRU reached the accuracy score of 92.31 and also reached the precision rate of 90.91, the recall value of 92.06, and finally the F1-score of 94.81. RGRU-ESFO reached the accuracy score of 94.68, the precision rate of 94.95, the recall value of 95.32, and finally the F1-score of 96.78, respectively.

Table 4 shows the comparison of the proposed model to the existing technique for 80%–20 %. In this comparison analysis, DBN reached an accuracy of 80.66 and CNN reached an accuracy of 89.54. and also, the RNN model reached an accuracy of 88.3. Additionally, the LSTM reached an accuracy of 92.05. The RGRU reached an accuracy of 91. And finally, the RGRU-ESFO model reached an accuracy of 94.

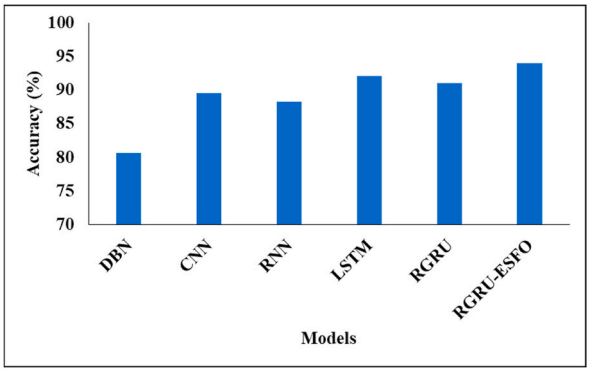


Fig. 9. Accuracy comparison.

Table 2
Validation analysis of model on flood dataset.

Methodology	Accuracy	Precision	Recall	F1
DBN	92.82	92.42	93.75	93.08
CNN	92.19	33.29	93.96	49.16
RNN	93.54	90.19	97.47	93.69
LSTM	93.61	38.39	97.43	55.08
RGRU	94.80	94.58	95.05	94.82
RGRU-ESFO	97.55	62.97	94.95	75.72

Table 3
Comparative analysis of proposed model on overall dataset.

Methodology	Accuracy	Precision	Recall	F1
DBN	90.84	91.03	88.27	91.65
CNN	91.60	88.14	89.24	84.47
RNN	91.59	93.19	89.45	91.28
LSTM	85.33	80.44	91.59	93.18
RGRU	92.31	90.91	92.06	94.81
RGRU-ESFO	94.68	94.95	95.32	96.78

Table 4
Analysis of the proposed model with existing techniques for 80%–20 %.

Methodology	Accuracy (%)
DBN	80.66
CNN	89.54
RNN	88.3
LSTM	92.05
RGRU	91
RGRU-ESFO	94

5. Limitations

While a hybrid deep learning model with increased Sunflower Optimisation has advantages for flood and earthquake detection, it also has limitations and challenges:

Data Availability and Quality: The availability and quality of data are critical to the performance of deep learning models. It might be difficult to obtain a big and high-quality dataset for flood and earthquake detection, especially for unusual or intense events. In some circumstances, the data may be incomplete or noisy, affecting the model's performance.

Computing Resources: Deep learning models, particularly big ones, necessitate extensive computing resources, such as high-performance GPUs and RAM. Adding Sunflower Optimisation on top of this can raise the computational cost and time required for hyperparameter optimisation even higher.

Hyperparameter Sensitivity: While hyperparameter optimisation is an important part of deep learning, it can be time-consuming

and dependent on the optimisation algorithm and search space used. Enhanced Sunflower Optimisation may not always result in the optimum hyperparameters, and manual tuning may still be necessary.

Interpretability: Deep learning models are frequently seen as black boxes, making it difficult to decipher the rationale behind their predictions. Interpretability is vital for confidence and decision-making in critical applications such as disaster detection.

6. Conclusion

One of the most challenging issues in catastrophe detection right now is real-time early detection and notification. Because of this issue, people who live in disaster zones will have to deal with a number of problems. Smart technology has been used to make plans for rescuing people in the event of a flood or earthquake. In order to forecast natural disasters like floods and earthquakes, an RGRU model is suggested here. The results of the experiment show that combining the residual network with the GRU for data relationship extraction is a good idea. Finally, it is proposed to retrain the RGRU fully-connected layer using the ESFO method. The ESFO optimises the pollination operator used by the standard SFO algorithm so that it can effectively explore the space of possible solutions. In polynomial time, the suggested approach locates the best answer. We repurposed the static-environment-tested model and trained it on new data spanning a wide range of human behaviours to use in a dynamic setting. The chosen model performed better than the alternatives and generated fewer false positives. Our method retains the same model and characteristics, whether applied to a static or dynamic setting. In the future, we plan to investigate novel features and models that have a lower computing burden while still being capable of doing well in detection tasks on difficult non-earthquake datasets.

Data availability statement

Data will be made available on request.

Additional information

No additional information is available for this paper.

Research involving human participants and/or animals

Not applicable.

Ethics approval

Not applicable.

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

CRediT authorship contribution statement

Phalguna Krishna E S: Data curation, Conceptualization. **Venkata Nagaraju Thatha:** Methodology, Investigation. **Gowtham Mamidisetti:** Software, Resources. **Srihari Varma Mantena:** Writing – original draft, Visualization, Validation. **Phanikanth Chintamaneni:** Writing – original draft. **Ramesh Vatambeti:** Writing – review & editing, Supervision.

Declaration of competing 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

[1] H.S. Munawar, F. Ullah, S. Qayyum, A. Heravi, Application of deep learning on uav-based aerial images for flood detection, Smart Cities 4 (3) (2021) 1220–1242.
[2] M. Hasanlou, R. Shah-Hosseini, S.T. Seydi, S. Karimzadeh, M. Matsuoka, Earthquake damage region detection by multitemporal coherence map analysis of radar and multispectral imagery, Rem. Sens. 13 (6) (2021) 1195.
[3] K. Shoyama, Q. Cui, M. Hanashima, H. Sano, Y. Usuda, Emergency flood detection using multiple information sources: integrated analysis of natural hazard monitoring and social media data, Sci. Total Environ. 767 (2021), 144371.
[4] L. Belcastro, F. Marozzo, D. Talia, P. Trunfio, F. Branda, T. Palpanas, M. Imran, Using social media for sub-event detection during disasters, Journal of big data 8 (1) (2021) 1–22.
[5] I. Suwarno, A. Ma'arif, N.M. Raharja, A. Nurjanah, J. Ikhsan, D. Mutiarin, IoT-based lava flood early warning system with rainfall intensity monitoring and disaster communication technology, Emerging Science Journal 4 (0) (2021) 154–166.

[6] M. Aamir, T. Ali, M. Irfan, A. Shaf, M.Z. Azam, A. Glowacz, F. Brumercik, W. Glowacz, S. Alqhtani, S. Rahman, Natural disasters intensity analysis and classification based on multispectral images using multi-layered deep convolutional neural network, Sensors 21 (8) (2021) 2648.
[7] S. Kaur, S. Gupta, S. Singh, Hurricane damage detection using machine learning and deep learning techniques: a review, in: IOP Conference Series: Materials Science and Engineering, 1022, IOP Publishing, 2021, 012035. No. 1.
[8] A. Vecere, M. Martina, R. Monteiro, C. Galasso, Satellite precipitation–based extreme event detection for flood index insurance, Int. J. Disaster Risk Reduc. 55 (2021), 102108.
[9] Y. Pi, N.D. Nath, A.H. Behzadan, Detection and semantic segmentation of disaster damage in UAV footage, J. Comput. Civ. Eng. 35 (2) (2021), 04020063.
[10] S. Bernabé, C. González, A. Fernández, U. Bhangale, Portability and acceleration of deep learning inferences to detect rapid earthquake damage from vhr remote sensing images using intel openvino toolkit, IEEE J. Sel. Top. Appl. Earth Obs. Rem. Sens. 14 (2021) 6906–6915.
[11] A. Pravin, T.P. Jacob, R. Rajakumar, Enhanced flood detection system using IoT, in: 2021 6th International Conference on Communication and Electronics Systems (ICCES), IEEE, 2021, July, pp. 507–510.
[12] A. Sharifi, Development of a method for flood detection based on Sentinel-1 images and classifier algorithms, Water Environ. J. 35 (3) (2021) 924–929.
[13] M.S. Sruthi, M.J. Poovathingal, V.N. Nandana, S. Lakshmi, M. Samshad, V.S. Sudeesh, YOLOv5 based open-source UAV for human detection during search and rescue (SAR), in: 2021 International Conference on Advances in Computing and Communications (ICACC), IEEE, 2021, October, pp. 1–6.
[14] K. Sharma, D. Anand, M. Sabharwal, P.K. Tiwari, O. Cheikhrouhou, T. Frikha, A disaster management framework using internet of things-based interconnected devices, Math. Probl Eng. 2021 (2021) 1–21.
[15] E.E.P. Talplacido, M.J.M.B. Zuñiga, M.D.T. Yusi, S.A. Loria Jr., A.G. Galang, Real-time and automated flood detection &early warning systems for the municipality of san leonardo, nueva ecija: a case study, American Journal of Multidisciplinary Research & Development (AJMRD) 3 (11) (2021) 28–38.
[16] V. Menon, R. Arjun Rathya, A. Prasad, A. Gopinath, N.B. Sai Shibu, G. Gayathri, Exploring IoT-enabled multi-hazard warning system for disaster-prone areas, in: Advances in Computing and Network Communications: Proceedings of CoCoNet 2020, vol. 1, Springer Singapore, 2021, pp. 405–422.
[17] J. Jackson, S.B. Yussif, R.A. Patamia, K. Sarpong, Z. Qin, Flood or non-flooded: a comparative study of state-of-the-art models for flood image classification using the FloodNet dataset with uncertainty offset analysis, Water 15 (5) (2023) 875.
[18] D. Xiang, X. Zhang, W. Wu, H. Liu, DensePPMUNet-a: a robust deep learning network for segmenting water bodies from aerial images, IEEE Trans. Geosci. Rem. Sens. (2023).
[19] D. Hernández, J.M. Cecilia, J.C. Cano, C.T. Calafate, Flood detection using real-time image segmentation from unmanned aerial vehicles on edge-computing platform, Rem. Sens. 14 (1) (2022) 223.
[20] K. Yang, S. Zhang, X. Yang, N. Wu, Flood Detection Based on Unmanned Aerial Vehicle System and Deep Learning, Complexity, 2022, 2022.
[21] S. Lenka, P. Kerhalkar, P. Shetty, H. Gupta, B. Vidyarthi, U. Verma, Cross-Geography Generalization of Machine Learning Methods for Classification of Flooded Regions in Aerial Images, 2022 arXiv preprint arXiv:2210.01588.
[22] J. Li, Z.J. Bortolot, Quantifying the impacts of land cover change on catchment-scale urban flooding by classifying aerial images, J. Clean. Prod. 344 (2022), 130992.
[23] M. Priyatna, S.K. Wijaya, M.R. Khomarudin, F. Yulianto, G. Nugroho, P.M. Afgatiani, A. Rarasati, M.A. Hussein, The use of multi-sensor satellite imagery to analyze flood events and land cover changes using change detection and machine learning techniques in the Barito watershed, Journal of Degraded and Mining Lands Management 10 (2) (2023) 4073–4080.
[24] X. He, S. Zhang, B. Xue, T. Zhao, T. Wu, Cross-modal change detection flood extraction based on convolutional neural network, Int. J. Appl. Earth Obs. Geoinf. 117 (2023), 103197.
[25] B. Zhao, H. Sui, J. Liu, Siam-DWENet: flood inundation detection for SAR imagery using a cross-task transfer siamese network, Int. J. Appl. Earth Obs. Geoinf. 116 (2023), 103132.
[26] National research Institute for Earth science and disaster prevention. Available online: http://www. kyoshin.bosai.go.jp (accessed on 31 January 2020)..
[27] Peer Ground Motion Database, Pacific earthquake engineering research center, Available online: https://ngawest2.berkeley.edu/. (Accessed 31 January 2020).
[28] Relief Web, Pakistan: floods and landslides—august 2020, Available online: https://reliefweb.int/disaster/fl-2020-000185-pak-0. (Accessed 6 October 2020).
[29] B. Krawczyk, Learning from imbalanced data: open challenges and future directions, Prog. Artif. Intell. 5 (2016) 221–232.
[30] H.H. Bock, Clustering methods: a history of k-means algorithms, in: Selected Contributions in Data Analysis and Classification, Springer, Berlin, Germany, 2007, pp. 161–172.
[31] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising, IEEE Trans. Image Process. 26 (2017) 3142e55, https://doi.org/10.1109/TIP.2017.2662206.
[32] N.S. Keskar, R. Socher, Improving Generalization Performance by Switching from ADAM to SGD. ArXiv, 2017.
[33] H. Yin, Z. Dong, Y. Chen, J. Ge, L.L. Lai, A. Vaccaro, et al., An effective secondary decomposition approach for wind power forecasting using extreme learning machine trained by crisscross optimization, Energy Convers. Manag. 150 (2017) 108e21, https://doi.org/10.1016/j.enconman.2017.08.014.
[34] G. Ferreira Gomes, S.S. da Cunha, A. Carlos Ancelotti, A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates, Eng. Comput. 35 (2) (2019) 619–626.