Earthquake forecasting using optimized levenberg–marquardt back-propagation neural network

MANOJ KOLLAM
Department of Electrical and Computer Engineering
The University of West Indies
St.Augustine, TRINIDAD AND TOBAGO

AJAY JOSHI
Department of Electrical and Computer Engineering
The University of West Indies
St.Augustine, TRINIDAD AND TOBAGO

Abstract: In this study, an effective earthquake forecasting model is introduced using a hybrid metaheuristic machine learning (ML) algorithm with CUDA-enabled parallel processing. To improve the performance and accuracy of the model, a novel hybrid ML model is developed that utilizes parallel processing. The model consists of a Chaotic Chimp based African Vulture Optimization Algorithm (CCAVO) for feature selection and a Hybrid Levenberg-Marquardt Back-Propagation Neural Network (HLMt-BPNN) for prediction. The proposed model follows a four-step process: preprocessing the raw data to identify seismic indications, extracting features from the preprocessed data, using optimized ML algorithms to forecast the earthquake and its expected time, epicenter, and magnitude, and implementing the model using the Python platform. The model's performance is evaluated using various criteria, including accuracy, precision, recall, F-measure, specificity, false negative ratio, false positive ratio, negative prediction value, Matthew's correlation coefficient, root mean square error, mean absolute error, and mean absolute percentage error. The proposed model achieved an accuracy of 98%, which is higher than the accuracy of existing earthquake prediction methods.

Keywords: Seismic indicators; Earthquake catalogue;Magnitude predictions; African vulture optimization; GPU,Levenberg-Marquardt Back-Propagation Neural Network.

Received: April 27, 2022. Revised: May 25, 2023. Accepted: June 22, 2023. Published: August 3, 2023.

1. Introduction

The most frequent natural disaster is an earthquake, which happens when tectonic plates slide past one another or laterally. This causes significant losses in human lives and material goods by disrupting the seas and land masses. The Richter scale, which ranges from 0 to 9, is used to quantify the intensity of earthquakes [1] [2] [3]. Strong earthquakes are those with a Richter scale value greater than or equal to 6. Furthermore, due to changes in the structure of the region that is prone to rupture, selective release of tension, and a variety of additional flaws, earthquakes do not occur on a regular basis. This proves that the intervals between these seismic occurrences must be unquestionably erratic in character [4] [5]. One of the key factors in an earthquake's categorization is its magnitude. The strength of the earthquake source is shown via a logarithmic scale. Magnitude is utilized in scientific study as well as to quickly educate the public about earthquakes [6] [7] [8].

Numerous research have proposed several forms of magnitude scales ever since the so-called local (ML) or Richter scale, which is used to measure earthquake magnitude. Although these magnitude scales may indicate fundamentally distinct aspects of the source, they are appropriate for a variety of magnitude of earthquakes and the distances between epicenters despite measuring differing seismic wave attributes. Quantity scales are often empirical. Typically, a magnitude is calculated using a formula containing a number of constants [9] [10] from the time and amplitude of a certain type of seismic wave. These constants are chosen such that, at least within a particular magnitude range, a new scale's magnitudes match those of an existing one. On a seismogram, the length of shaking can occasionally be used to estimate magnitude. Because of this, there may be more than one magnitude unit of difference between the values of the various magnitude categories for both very large and very small earthquakes as well as for some specific classes of seismic source. This is due to the complicated physical mechanism that causes an earthquake [11] [12] [13].

The development of an awareness system utilizing ML has been a growing area of research across all sectors of engineering and science as a result of the losses brought on by an earthquake. Numerous studies have advanced in this approach. Geologists and earthquake specialists now have a new, creative technique to assess seismic risk and trigger future earthquakes that exceeds the traditional, established ways they had previously used. Earthquake projections can be divided into two categories: forecast predictions and short-term predictions [14] [15]. In contrast to long-term estimates, which are made months to years before it happened, short-term earthquake predictions are created hours to days beforehand. The main goal of this research is to use various ML methods to forecast whether a significant earthquake would be labelled as a negative or positive event. The model cannot be solved perfectly using ML alone. A new ML model is created in parallel to improve the model's accuracy and performance. Since the parallelism is naturally supplied by employing the architecture for constructing GPU utilizing computational techniques, known as the Compute Unified Device Architecture, the shortcomings of ML using a central processing unit (CPU) may be solved by Graphic Processing unit (GPU) implementation (CUDA). The implementation of hybrid state vector machine (HSVM) algorithm using parallel processing through CUDA is used to forecast earthquakes.

The foremost contribution of the paper is as follows,

- Chaotic Chimp based African Vulture Optimization Algorithm (CCAVO) is used for feature selection.
- The CUDA model is used for train the extracted data. The CUDA model will process the data parallelly. This is the main advantage of CUDA model.
- Then the prediction is performed with the help of HLMt-BPNN model. The accuracy of the prediction model is improved using the ISOA.

- By comparing the CPU and GPU's respective computation times, the performance of the proposed model is compared to that of the present model.

This essay's remaining sections are organized as follows: Section 2 discussed the literature reviews that were completed by earthquake forecasting and the history of the ideas employed in this article. The methodology of the proposed models was explained in Section 3. the outcomes of all the approaches are provided, and the best method is determined by comparison with a few other tried-and-true techniques are described in Section 4. The study was concluded in Section 5. The formatter will need to create these components, incorporating the applicable criteria that follow.

2.Proposed Methodology

In this article, an HLMt-BPNN algorithm is created for use with GPUs and the CUDA programming environment. The computation speed and performance of the forecasting model were increased by the method by adding the GPU to machine learning, which further enhanced resilience. Preprocessing, feature extraction, feature selection, model training, and predictions are eventually made on an unobserved portion of the dataset are other procedures that are involved. The effectiveness of the prediction model is ultimately assessed, and comparisons are made. Fig.1 depicts the overall architecture diagram.

The The eight seismic parameters described in Section 3.1 have been subjected to a variety of ML techniques. The earthquake is classified by considering the threshold value specifically based on magnitude i.e., Magnitude>5.5 illustrates that the earthquake is occurred. Conversely, the Magnitude<5.5 shows that the earthquake is non occurred. This can be stated as 0 or 1 problem which means 0 denotes non-occurred and the 1 denotes occurred. The major indention of the proposed model is to deal with the binary classification issue. After these approaches have been trained, output on unknown data parameters is generated, and performance is then assessed in Section 4. The preprocessing is significant step for enhancing the prediction performance. The subsequent section describes the feature selection step in detail.

2.1 Feature Selection

To select the best optimal features from the extracted features, a Chaotic Chimp based African Vulture Optimization Algorithm (CCAVO) is used.

1) Chaotic Chimp based African Vulture Optimization Algorithm

The AVOA (African Vulture Optimization Algorithm) is a nature-inspired metaheuristic algorithm that was developed as a tool for optimization. It is based on the observed behavior of African vultures, which are known for their ability to find food in a wide range of environments. One potential advantage of AVOA is its ability to effectively search for solutions in a wide range of optimization problems, including those with many variables and complex constraints. It is also relatively simple to implement, as it only requires a few parameters to be set by the user. AVOA has been applied to various optimization problems and has shown to be effective at finding good solutions. AVOA has been tested on a variety of optimization problems and has demonstrated its ability to find high-quality solutions.

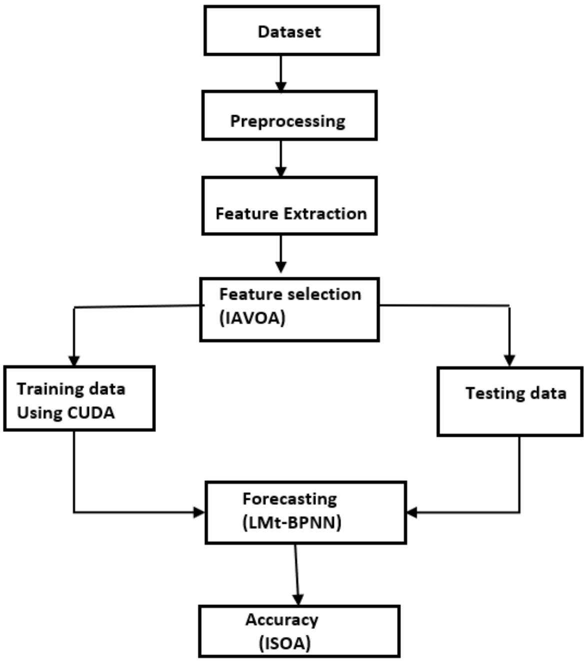

Fig. 1. Block diagram of the proposed methodology.

Stage 1: Vulture Group Formation

In the first phase of the CCAVO method, the initial population of vultures is created and the fitness of all solutions is evaluated. The vulture corresponding to the best solution is identified as the first vulture, the vulture corresponding to the second-best solution is identified as the second-best vulture using the Eq. (11), and all the other vultures are assigned to the third group according to the second criteria. This phase sets the foundation for the subsequent phases of the foraging stage, in which the vultures' positions are updated and their fitness values are reevaluated.

$$B_{vai} = \begin{cases} B_{va1} & \text{if } n_i = r_1 \\ B_{va2} & \text{if } n_i = r_2 \end{cases} \quad (1)$$

In this phase, the variables B_{va1} and B_{va2} represent the best and second-best vultures, respectively and r_1 and r_2are two random values in the range [0,1] such that their sum is 1. The value of n_i is determined using the roulette-wheel technique as shown in Eq. (2).

$$z_i = \frac{Fitn_i}{\sum_{i=1}^{n} Fitn_i} \quad (2)$$

In this phase, $Fitn_i$, which represents the fitness of the first and second groups of vultures, and n, which represents the combined number of vultures in both groups, are used.

Stage 2: Vulture Starvation Level

The CCAVO algorithm uses the hunger level of vultures, as calculated by Eq. (3), to determine their exploration and exploitation behavior. When vultures are not hungry, they have the energy and endurance to fly long distances in search of food. However, if they are hungry, they will not be able to sustain flight for as long and may act aggressively in their search for food. The i^{th} hunger level of the vultures, represented by (Fit_i), serves as an indicator of their transition from exploitation to exploration. In this way, the CCAVO algorithm is able to simulate the real-world behavior of vultures in the search for food and apply it to the optimization process.

$$Fitn_i = (2 \times rana' + 1) \times d \times \left(1 - \frac{iter_i}{\max(iter_i)}\right) + y \qquad (3)$$

Where $Fitn_i$ denotes that the vultures have consumed all of the available food, $rana'$ is a random variable with a value between 0 and 1, d is a random number with a range of [1,1] that varies with each iteration, and t is determined by Eq. (4).

$$x = b \times \left(sin^c\left(\frac{\pi}{2} \times \frac{ite_i}{\max(ite_i)}\right) + cos\left(\frac{\pi}{2} \times \frac{ite_i}{\max(ite_i)}\right) - 1\right) \qquad (4)$$

Where the value c determines the likelihood that the vulture will execute the exploitation stage. In addition, ite_i stands for the current iteration number, $\max(ite_i)$ for the total number of iterations, and b for a random number between -2 and 2. When the value of $Fitn_i$ is greater than 1, the vultures begin the exploration phase and look for new food sources in diverse places. In the absence of this, vultures enter the stage of exploitation and search the nearby area for better food.

Stage 3: Search Stage

Vultures can swiftly seek food and identify dead animals because of their excellent vision in the natural world. But because they spend a lot of time scanning their surroundings before taking off, vultures can have trouble finding food. A long way in search of nourishment. A parameter labelled z_2 in the range [0,1] is used to select which of two different techniques vultures in the CCAVO can use to check numerous random sites.

A random number randp1 between 0 and 1 is used to select one of the strategies during the exploration phase. $Z(i+1)$ represents the position of the vulture in the next iteration of the optimization process

$$Z(i+1) = B_{vai} - L(i) \times Fitn_i \qquad (5)$$

$$Z(i+1) = B_{vai} \text{ - } Fitn_i + rana'2 \times ((U_b - L_b) \times rana'3 + lb) \qquad (6)$$

Where, $rand'2$ is a random integer between 0 and 1, B_{vai} is one of the best vultures selected in the current iteration, $Fitn_i$ is the current iteration's rate of vulture satiation derived using Eq. (6), and L_b and U_b are the variables' lower and upper bounds, respectively. $rana'3$ is used to give a high random coefficient at the search environment scale, increasing diversity and the search for different search space areas. Eq. (7) calculates L_i, which stands for the separation between the vulture and the currently optimal one.

$$L_i = |A \times B_{va}(\mathrm{i}) - \mathrm{Z(i)}| \qquad (7)$$

Here, 'A' is a randomly chosen number between 0 and 2, and L_i denotes the location of the ith vulture.

Stage 4: First Exploitation Stage

The efficiency stage of the CCAVO is investigated at this point. If $Fitn_i$ value is less than 1, the CCAVO initiates the first phase of exploitation. The selected approach is determined by the parameter z_2 in the interval [0,1]. A random integer between 0 and 1 is generated at the beginning of this phase, $rand_{z_2}$. If this $rand_{z_2}$ is greater than or equal to parameter z_2, the siege-fight tactic is employed gradually. If not, the circular flying method is employed. As per Eq. (8),

$$Z(i+1) = \begin{cases} L_i \times (Fitn_i + \ rana'4) - g(t) & if\ z_2 \geq rana_{z_2} \\ B_{vai} - (q_1 + q_2) & if\ z_2 < rana_{z_2} \end{cases} \qquad (8)$$

Where $g(t)$ represents the distance between the vulture and one of the two groups' top vultures, as determined by Eq. (9), and $rana'4$ is a random number between 0 and 1.

$$g(t) = B_{vai} - \mathrm{Z(i)} \qquad (9)$$

$$Vas_1 = B_{vai} \times \left(\frac{rana'5 \times \mathrm{Z(i)}}{2\pi}\right) \times \cos(\mathrm{Z(i)}) \qquad (10)$$

$$Vas_2 = B_{vai} \times \left(\frac{rana'6 \times \mathrm{Z(i)}}{2\pi}\right) \times \sin(\mathrm{Z(i)}) \qquad (11)$$

$$s = chaotic\_value \qquad (12)$$

$$Z(i+1) = B_{vai} - s \cdot (Vas_1 + Vas_2) \qquad (13)$$

$rana5$ and $rana'6$ are random numbers between 0 and 1. Eq. (10) and Eq. (11) are used to determine saturated vulture one Vas_1 and saturated vulture two Vas_2, and s is the chaotic vector based on chimp optimization.

Stage 5: Second Exploitation Stage (Chaotic Chimp based Enhancement in AVO) (proposed)

The chaotic maps listed in Table 1 are used to enhance the performance of CCAVO. These deterministic processes can also produce random behavior. The update process is modeled as follows as per Eq. (12)

$$Y_{vulture}(t+1) = \{Chaotic\_value, \quad |Fitn_i| \geq \ 0.5 \qquad (14)$$

where, μ is the random number in [0,1].

To summarize, the CCAVO algorithm begins by generating a random population of "vultures" (candidate solutions). Each vulture then updates its f coefficients using its own group's strategy. During the iteration, the attacker, barrier, chaser, and driver all estimate the possible locations of the prey. The candidate solutions also update their distance from the prey. The adaptive tuning of the c and m parameters help to avoid local optima and improve the convergence rate. Additionally, the value of f is reduced from 2.5 to 0 to enhance the exploitation process. If the inequality $|a| > 1$ is satisfied, the chimps diverge from the prey, otherwise they eventually converge towards it. Finally, the chaotic maps help to speed up convergence without getting stuck in local minima.

Table 1: Chaotic Maps

S. No	Name	Chaotic Map	Range
1	Bernoulli	$x_{i+1} = 2X_i(mod\ 1)$	(0,1)
2	Quadratic	$x_{i+1} = x_i^2 - c, c = 1$	(0,1)
3	Iterative	$x_{i+1} = \sin\left(\frac{a\pi}{x_i}\right),$ $a = 0.7$	(-1,1)

2.2 Earthquake prediction

The eight seismic parameters have been subjected to various ML techniques. With earthquakes of magnitude 5.5 and bigger being classified as Yes or 1 and earthquakes of lesser magnitude as No or 0, the prediction job is approached as a binary classification issue. These strategies produce results on unknown data parameters after training.

1) Levenberg–Marquardt backpropagation (LMA)

The Levenberg-Marquardt method is implemented here using the usual backpropagation approach. The algorithm bears the names of the researchers who developed it. It is taught how to train feedforward networks using the Levenberg-Marquardt method, and it is made clear how much better neural networks compute when they use this algorithm rather than backpropagation as is often done. Because of this, attempts were made to change the LMA-based backpropagation learning algorithm, which is noteworthy from the perspective of a contribution. The writers of this work described every mathematical formulation and function used to modify the conventional backpropagation along the lines of LMA. The construction of a "Hessian" matrix using this approach has the benefit of using initial derivatives with regard to network weights, which are conveniently handled by the usual backpropagation. So, the algorithm's overall computing complexity decreases. The algorithm is specifically made to reduce the total squared mistakes. A Taylor series can be used to expand the error vector to first order if there is little difference between the old and new weight vectors. The error function can therefore be provided as per Eq. (15),

$$E_f = \frac{1}{2}\left\|e_f(i) + \frac{\partial e_f(k)}{\partial w_f(j)}(w_f(i+1) - w_f(i))\right\|^2 \quad (15)$$

$e_f(i)$ is an error vector and $e_f(k)$ is its element, $w_f(i+1)$ and $w_f(i)$ are new and previous weight vector respectively. When the aforementioned function is minimised with regard to the new weight vector as per Eq. (16)

$$w_f(i+1) = w_f(i) - (C^T C)^{-1} C^T e_f(i)$$
$$where\ C_{kj} = \frac{\partial e_f(k)}{\partial w_f(j)} \quad (16)$$

The formula is based on linear approximation, which is another factor. In order to guarantee the validity of the linear approximation, the step size is kept small in the LMA while the error function is reduced. The error function is somewhat adjusted as a result:

$$E_f = \frac{1}{2}\left\|e_f(i) + \frac{\partial e_f(k)}{\partial w_f(j)}(w_f(i+1) - w_f(i))\right\|^2 + \varphi\left\|w_f(i+1) - w_f(i)\right\|^2 \quad (17)$$

In this, φ determines the step size. Similar to this, reducing the error now in relation to $w_f(i+1)$ results in

$$w_f(i+1) = w_f(i) - (C^T C)^{-1} C^T e_f(i) \quad (18)$$

2) Levenberg–Marquardt backpropagation (LMA)

The topology of the BP neural network is shown in Fig. 2. The network's input layer, hidden layer, and output layer's corresponding node counts are represented by the letters n, T, and m. h_{xy} and h_{yz} are used to represent connection weights. The BP neural network's input value is represented by the letters a_1, a_2, a_3, and a_l, while the predicted value is represented by the letters b_1, b_2, b_3, and b_l. The BP neural network is trained using the following method: the neural network should be started. According to the criteria for real prediction, the values are chosen, and the hidden layer threshold **a** and output layer threshold **b** are initialized. Following that, the neural network's learning rate and the neuron's excitation function are calculated.

1. The chosen implicit layer excitation function in this work, f, is:

$$g(a) = 1/(1 + e^{-a}) \quad (19)$$

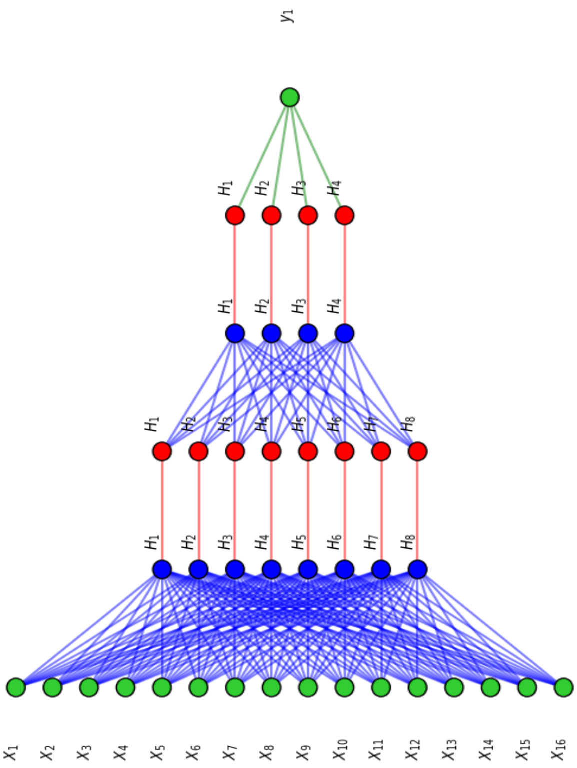

Fig 2: Topological structure of NN

2. Determine the buried layer's output. Given that a, h_{xy}, and p are known, it is possible to calculate the hidden layer's

output Hl_{Out}:

$Hl_{Out} = g(\sum_{x=1}^{l} h_{xy}a_x - p_y)\ \ where\ y = 1,2, \ldots., m$ (20)

The buried layer's N nodes are what make up the formula.

3. Determine what the output layer should be. The expected output Out_z value may be determined from Hl, h_{yz}, and y.

$Out_z = \sum_{y=1}^{n} Hl_{Out}h_{yz} - y_z \quad where\ z = 1,2, \ldots.n$ (21)

4. Determine the inaccuracy in the forecast. By deducting Out_z from the anticipated output b, one may derive the model prediction error e_z.

$e_z = b_z - Out_z, z = 1,2, \ldots m$ (22)

5. Update the weights. Update h_{xy} and h_{yz} in accordance with e_z. The following are the expressions:

$h_{xy} = h_{xy} + \zeta Hl_{Out}(1 - Hl_{Out})a(x) \sum_{z=1}^{n} h_{yz}\, e_z$ (23)

where $x = 1, 2, \ldots.., l$ & $y = 1, 2, \ldots.., m$

$h_{yz} = h_{yz} + \zeta Hl_{Out}e_z$ (24)

where $y = 1, 2, \ldots.., m$ & $z = 1,2, \ldots.., n$

where, h is the learning rate.

6. Update threshold. Update p, q according to e_z

$p_y = p_y + \zeta Hl_{Out}(1 - Hl_{Out}) \sum_{z=1}^{n} h_{yz}\, e_z$ (25)

where $y = 1, 2, \ldots, m$

$q_z = q_z + e_z$ where $z = 1, 2, \ldots, n$ (26)

7. Determine if the model has converged, and if not, return to step 2 to continue the iteration.

3. Result and Discussion

In this section, the performance of the proposed model is compared with the existing models by using the performance metrics. The computation time for the CPU is compared with the GPU-CUDA model. The comparison table is shown in table 2.

3.1 Dataset description

Modern automatic phase pickers have been compared using the dataset, a global collection of more than a million seismic waveforms labelled with both P- and S-arrival (Mousavi et al., 2020). To evaluate how successfully the recommended technique identified phases, we used the same test set (120,000 waveforms). The phase selecting networks were used to choose P- and S-phase arrivals from the waveforms after the feature extraction network processed the waveforms. In order to identify phases and pinpoint arrival timings, we chose the peaks from the predicted activation sequences that were higher than a threshold of 0.5. True positives are those projected selections that are within 0.5 seconds of the manual labelling. The remainder are regarded as false positives.

3.2 Performance Metrics

Define Utilizing performance metrics including accuracy, precision, recall, F-measure, RMSE, MAE, and MAPE, the suggested model's performance is assessed.

Table 2: Comparison of the performance metrics between the proposed and existing techniques

Method	Accuracy	Precision	Recall	F-measure	specificity	FNR
Proposed _HLMt_BPNN	98.00	95.84	97.95	94.57	95.73	0.04
Existing_ LSTM	95.80	92.38	96.58	91.36	92.72	0.07
Existing_ GRU	90.12	90.58	90.99	87.58	90.22	0.07
Existing_ CNN	89.77	86.40	90.10	85.91	86.40	0.09
Existing ANN	85.80	84.50	88.52	84.25	87.00	0.10

Method	FPR	MCC	NPV	RMSE	MAE	MAPE
Proposed _HLMt_ BPNN	0.01	96.21	93.74	0.25	0.41	0.36
Existing_ LSTM	0.05	93.78	91.60	5.84	18.87	13.58
Existing_ GRU	0.08	90.68	90.66	57.90	49.96	46.90
Existing_ CNN	0.08	86.63	88.69	32.69	38.87	34.71
Existing ANN	0.09	83.99	87.95	236.59	183.95	175.78

The tabulated values are shown in the form of graphs. The performance metrics for the proposed model are higher than the existing models which are explained separately.

The accuracy values for the proposed and the existing techniques like LSTM, GRU, CNN, and ANN are 98.00, 95.80, 90.12, 89.77, and 85.80 respectively. The proposed model produces higher accuracy than the other existing techniques.

The precision values for the proposed and the existing techniques, such as LSTM, GRU, CNN, and ANN, are, respectively, 95.84, 92.38, 90.58, 86.40, and 84.50. The suggested model has higher precision than the other methods that are already in use.

The values for recall for the suggested and existing techniques, such as LSTM, GRU, CNN, and ANN, are 97.95, 96.58, 90.99, 90.10, and 88.52, respectively. In comparison to other methodologies, the suggested model produces results with higher recall.

iv) F-Measure

The Proposed HLMt-BPNN's F-measure is compared to the accuracy of existing models such as the LSTM, GRU, CNN, and ANN. Fig. 6 displays a graphical comparison of the F-measure rates.

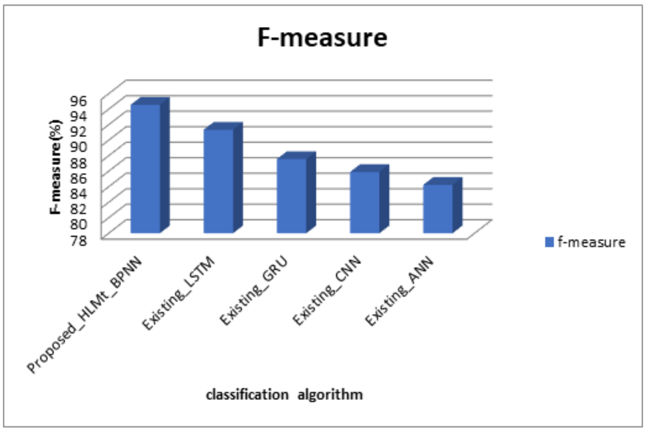

Fig 6: Comparison of F-measure between proposed and existing papers

The values for F-measure for the suggested and the existing techniques, such as LSTM, GRU, CNN, and ANN, are 94.57, 91.36, 87.58, 85.91and 84.25, respectively. In comparison to other methodologies, the suggested model produces results with higher F-measure.

The specificity values for the suggested and the currently employed techniques, such as LSTM, GRU, CNN, and ANN, are 95.73, 92.72, 90.22, 86.40, and 87.00, respectively. Comparing the specificity of the suggested model to other methods currently in use.

The FNR values for the suggested and existing techniques, such as LSTM, GRU, CNN, and ANN, are 0.04, 0.07, 0.07, 0.09, and 0.10 correspondingly. Compared to other methods already in use, the suggested model produces results with low FNR.

The recommended and current techniques, such as LSTM, GRU, CNN, and ANN, have FPR values of 0.01, 0.05, 0.08, 0.09, and 0.09, respectively. The proposed approach yields outcomes with lower FPR compared to existing techniques.

The MCC values for the proposed and existing techniques, such as LSTM, GRU, CNN, and ANN, are, respectively, 96.21, 93.78, 90.68, 86.63, and 83.99. The suggested model generates results with higher MCC compared to other techniques already in use.

The NPV values for the proposed and existing techniques, such as LSTM, GRU, CNN, and ANN, are, respectively, 98.00, 95.80, 90.12, 89.77, and 85.80. The suggested model generates results with higher NPV compared to other techniques already in use.

The FRR values for the proposed and the existing techniques, such as LSTM, GRU, CNN, and ANN, are 98.00, 95.80, 90.12, 89.77, and 85.80, respectively. The proposed model generates results with lower FRR compared to other existing approaches.

The recommended and current techniques, such as LSTM, GRU, CNN, and ANN, have RMSE values of 0.25, 5.84, 57.90, 32.69, and 236.59, respectively. The proposed approach yields outcomes with higher RMSE compared to existing techniques.

The values for F-measure for the suggested and the existing techniques, such as LSTM, GRU, CNN, and ANN, are 94.57, 91.36, 87.58, 85.91and 84.25, respectively. In comparison to other methodologies, the suggested model produces results with higher F-measure.

The MAPE values for the proposed and the currently used techniques, including LSTM, GRU, CNN, and ANN, are 0.36, 13.58, 46.90, 34.71, and 175.78, respectively. The suggested model yields low-loss findings when compared to other techniques currently in use.

In order to determine the proposed model's processing performance on CPU and GPU, 4000 epochs are taken into account because the model's correctness is constant after 2000 epochs. The proposed model is run on an Intel Core i7, 8 GB of RAM, and a GT 1050Ti GPU with 4 GB of RAM and 768 CUDA cores. The training of 4000 epochs for the PSVR model using GPU took 200 seconds, but the same task on the CPU took roughly 900 seconds. As can be observed the proposed model significantly outperformed the CPU in terms of computing speed when training the seismic catalog model.

Table 3: Exploring the Differences in Computation Time between CPU and GPU

Method	CPU (ms)	GPU (ms)
Proposed_HLMt_BPNN	44526	21102
Existing_LSTM	71421	48903
Existing_GRU	74234	55274
Existing_CNN	88706	59001
Existing_ANN	99934	77392

Table 3 shows the computation time differences between CPU and GPU models. The computation time of the CPU is high because the data is processed serially. But in the GPU model, the data is processed parallelly, so the computation time is low. This is the main advantage of the proposed model for forecasting an earthquake.

4. Conclusion

In this study, an effective earthquake forecasting model was presented that employs a hybrid metaheuristic machine learning algorithm with CUDA-enabled parallel processing. A novel hybrid ML model was developed to improve model performance and accuracy, using Chaotic Chimp based African Vulture Optimization Algorithm (CCAVO) for feature selection and a Levenberg-Marquardt Back-Propagation Neural Network for prediction. The Seagull Optimization Algorithm was also utilized to further enhance prediction accuracy. The model follows a four-step process involving preprocessing raw data, extracting features, using optimized ML algorithms to predict earthquakes, and implementing the model using the Python platform. The performance of the proposed model was evaluated using a variety of performance criteria, and the model achieved an accuracy of 98%, outperforming existing earthquake prediction methods. The use of parallel processing in the model's design enables efficient and fast prediction, making it suitable for real-time applications. These findings suggest that the proposed model could be a valuable tool for predicting earthquakes and potentially mitigating their impact.

References

[1] Wu, Y., Hou, G. and Chen, S., 2021. Post-earthquake resilience assessment and long-term restoration prioritization of transportation network. Reliability Engineering & System Safety, 211, p.107612.

[2] Hammad, A. and Moustafa, M.A., 2021. Numerical analysis of special concentric braced frames using experimentally-validated fatigue and fracture model under short and long duration earthquakes. Bulletin of Earthquake Engineering, 19(1), pp.287-316.

[3] Cremen, G., Velazquez, O., Orihuela, B. and Galasso, C., 2021. Predicting approximate seismic responses in multistory buildings from real-time earthquake source information, for earthquake early warning applications. Bulletin of Earthquake Engineering, 19(12), pp.4865-4885.

[4] Tena-Colunga, A., 2021. Conditions of structural irregularity. Relationships with observed earthquake damage in Mexico City in 2017. Soil Dynamics and Earthquake Engineering, 143, p.106630.

[5] Triantafyllou, I., Papadopoulos, G.A. and Lekkas, E., 2020. Impact on built and natural environment of the strong earthquakes of April 23, 1933, and July 20, 2017, in the southeast Aegean Sea, eastern Mediterranean. Natural Hazards, 100(2), pp.671-695.

[6] Zhang, X., Zhang, M. and Tian, X., 2021. Real-time earthquake early warning with deep learning: Application to the 2016 M 6.0 Central Apennines, Italy earthquake. Geophysical Research Letters, 48(5), p.2020GL089394.

[7] Tzouvaras, M., Kouhartsiouk, D., Agapiou, A., Danezis, C. and Hadjimitsis, D.G., 2019. The use of Sentinel-1 synthetic aperture radar (SAR) images and open-source software for cultural heritage: An example from Paphos area in Cyprus for mapping landscape changes after a 5.6 magnitude earthquake. Remote Sensing, 11(15), p.1766.

[8] Khalilpourazari, S. and Arshadi Khamseh, A., 2019. Bi-objective emergency blood supply chain network design in earthquake considering earthquake magnitude: a comprehensive study with real world application. Annals of Operations Research, 283(1), pp.355-393.

[9] Shi, Y., Liao, X., Zhang, D. and Liu, C.P., 2019. Seismic waves could decrease the permeability of the shallow crust. Geophysical Research Letters, 46(12), pp.6371-6377.

[10] Trugman, D.T., Chu, S.X. and Tsai, V.C., 2021. Earthquake Source Complexity Controls the Frequency Dependence of Near-Source Radiation Patterns. Geophysical Research Letters, 48(17), p.e2021GL095022.

[11] Tsai, V.C., Hirth, G., Trugman, D.T. and Chu, S.X., 2021. Impact versus frictional earthquake models for high-frequency radiation in complex fault zones. Journal of Geophysical Research: Solid Earth, 126(8), p.e2021JB022313.

[12] Hutchison, A.A., 2020. Inter-episodic tremor and slip event episodes of quasi-spatiotemporally discrete tremor and very low frequency earthquakes in Cascadia suggestive of a connective underlying, heterogeneous process. Geophysical Research Letters, 47(3), p.e2019GL086798.

[13] Mignan, A., Ouillon, G., Sornette, D. and Freund, F., 2021. Global earthquake forecasting system (GEFS): The challenges ahead. The European Physical Journal Special Topics, 230(1), pp.473-490.

[14] Nandan, S., Kamer, Y., Ouillon, G., Hiemer, S. and Sornette, D., 2021. Global models for short-term earthquake forecasting and predictive skill assessment. The European Physical Journal Special Topics, 230(1), pp.425-449.

[15] Wikelski, M., Mueller, U., Scocco, P., Catorci, A., Desinov, L.V., Belyaev, M.Y., Keim, D., Pohlmeier, W., Fechteler, G. and Martin Mai, P., 2020. Potential short-term earthquake forecasting by farm animal monitoring. Ethology, 126(9), pp.931-941.

[16] Xiong, P., Tong, L., Zhang, K., Shen, X., Battiston, R., Ouzounov, D., Iuppa, R., Crookes, D., Long, C. and Zhou, H., 2021. Towards advancing the earthquake forecasting by machine learning of satellite data. Science of The Total Environment, 771, p.145256.

[17] Gitis, V.G. and Derendyaev, A.B., 2019. Machine learning methods for seismic hazards forecast. Geosciences, 9(7), p.308.

[18] Jena, R., Pradhan, B., Beydoun, G., Alamri, A.M. and Sofyan, H., 2020. Earthquake hazard and risk assessment using machine learning approaches at Palu, Indonesia. Science of the total environment, 749, p.141582.

[19] Mousavi, S.M. and Beroza, G.C., 2020. A machine-learning approach for earthquake magnitude estimation. Geophysical Research Letters, 47(1), p.e2019GL085976.

[20] Rundle, J.B., Donnellan, A., Fox, G., Crutchfield, J.P. and Granat, R., 2021. Nowcasting earthquakes: imaging the earthquake cycle in California with machine learning. Earth and Space Science, 8(12), p.e2021EA001757.

[21] Aslam, B., Zafar, A., Khalil, U. and Azam, U., 2021. Seismic activity prediction of the northern part of Pakistan from novel machine learning technique. Journal of Seismology, 25(2), pp.639-652.

[22] Asim, K.M., Moustafa, S.S., Niaz, I.A., Elawadi, E.A., Iqbal, T. and Martínez-Álvarez, F., 2020. Seismicity analysis and machine learning models for short-term low magnitude seismic activity predictions in Cyprus. Soil Dynamics and Earthquake Engineering, 130, p.105932.

[23] Xiong, P., Long, C., Zhou, H., Battiston, R., Zhang, X. and Shen, X., 2020. Identification of electromagnetic pre-earthquake perturbations from the DEMETER data by machine learning. Remote Sensing, 12(21), p.3643.

[24] Cui, S., Yin, Y., Wang, D., Li, Z. and Wang, Y., 2021. A stacking-based ensemble learning method for earthquake casualty prediction. Applied Soft Computing, 101, p.107038.

[25] Zhang, Y., Burton, H.V., Sun, H. and Shokrabadi, M., 2018. A machine learning framework for assessing post-earthquake structural safety. Structural safety, 72, pp.1-16.

[26] Zhu, W., Tai, K.S., Mousavi, S.M., Bailis, P. and Beroza, G.C., 2022. An End-To-End Earthquake Detection Method for Joint Phase Picking and Association Using Deep Learning. Journal of Geophysical Research: Solid Earth, 127(3), p.e2021JB023283.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.