Earthquake Prediction Model Using Python

P.Usha[1] V.Ajai[2] B.D.S.Suntosh[3] M.Muthukumar[4] K.Manoj Raj[5] V.Suriyanarayanan[6]
ajaiff18@gmailcom suntosh1206@gmail.com mk14092003@gmail.com bm9724787@gmail.com suriyasrija21@gmailcom
Department of Electronics and Communication Engineering Velammal College of Engineering and Technology (Autonomous), Madurai

Abstract ----- Earthquakes can have devastating consequences, causing ground shaking, landslides, and changes in landscapes. Fault line ruptures can alter river courses and disrupt infrastructure, while underwater earthquakes may trigger tsunamis, affecting coastal ecosystems and communities. Liquefaction can temporarily weaken the ground, leading to structural damage, and aftershocks can further exacerbate existing Damage and hinder recovery efforts. Human impacts are significant and can result in injuries, fatalities, displacement, and psychological trauma. Economic consequences can involve disruption to industries and livelihoods, while response and recovery efforts may have environmental consequences. This paper focuses on earthquake prediction using various parameters such as date, time, latitude, longitude, depth, and magnitude. We have used a world map as a dataset to train our model, where we predict earthquakes using GRADIENT BOOSTING REGRESSOR. We have broken down the complex and challenging problem into simpler like mean squared error (MSE) as a loss function , accuracy , precision , recall , F1 score , confusion matrix . As of our last knowledge update in September 2023, earthquake prediction remains a field of ongoing research and does not have precise predictive models.The advantage of this model is its accuracy which is predicted as output . However, by comparing actual datasets with predicted outcomes of occurrences, we can identify risk free areas for livelihood . The proposed model achieved an accuracy of 86.1% and 99.7% in terms of magnitude and depth , which is higher than the accuracy of existing earthquake prediction methods.

Keywords—Gradient Boosting Regressor,Depth, Magnitude.

1.INTRODUCTION:

This paper addresses the vital topic of earthquake prediction, emphasizing the inherent challenges in understanding the complex physics involved. Earthquakes, as major natural hazards, demand accurate prediction for effective early warning systems and timely evacuations, mitigating potential damage and loss of life. To tackle these challenges, the paper explores the application of machine learning techniques, particularly the utilization of wearable applications for data collection. The collected data is transmitted to a central server for analysis, focusing on identifying potential earthquake precursors. Challenges include gaps in understanding earthquake physics, the difficulty of handling large data volumes, and the necessity for reliable early warning systems. Despite these hurdles, ongoing research, especially utilizing a random forest classifier on a provided CSV dataset plotting earthquake occurrences, shows promising progress. Continued efforts in research and development offer hope for achieving more accurate earthquake predictions, ultimately contributing to saving lives.[1] The method combines African Vulture Optimization and Neural Network for better earthquake prediction. It involves data processing, feature extraction, and Python implementation. The approach blends nature-inspired algorithms with advanced neural networks for improved accuracy. [2], It determines hidden layer neurons, incorporates specific earthquake features, and statistically evaluates predicted errors. The goal is to predict earthquakes independent of environmental features and identify the optimal number of hidden nodes for accurate predictions [3], It considers microwave brightness temperature and other satellite data for a comprehensive approach. The study also addresses a Middle East dust storm event for identifying and analyzing earthquake-related signals.[4] This approach aims to improve the accuracy of earthquake anomaly detection by leveraging machine intelligence-based algorithms and smaller window sizes to learn trends, seasonality, and residual patterns in the data.. [5] to predict TEC time series data and detect anomalies associated with earthquakes. [6] . Extensive experimental results demonstrate the effectiveness of the proposed prediction model, achieving high prediction performance for rockfall runout range.. [7], This neural network architecture is specifically designed to process sequential data by leveraging feedback loops. [8]. Calculated seismic parameters based on the temporal distribution of historic seismic events. These parameters were used as inputs for the machine learning classifiers. [9], Fast Fourier transformation variance, and spectral Welch density, from the time series data. Following the evaluation of various machine learning models, the study identifies the Light Gradient Boosting Mechanism [10] Predicting the hypo central distance and peak ground motion acceleration, feature extraction from P-wave time windows, and the application of GBR for predictive modeling [11]. Analysis by organizing and processing the sequential data to be used as inputs for the RNN model. [12]. Implemented models include an ANN with three dense layers and a Random Forest Regression tuned via grid search, providing a robust approach to earthquake prediction. "Our analysis aims to predict earthquakes using a gradient boosting regressor. This approach will increase the efficiency of earthquake prediction, which is crucial for recovering from the disastrous consequences of earthquakes and reducing their associated hazards."

Section-II PROPOSED METHODOLOGY
Section-III RESULT AND DISCUSSION
Section-IV CONCLUSION

METHODOLOGY:

Gradient Boosting Regressor (GBR) is an ensemble learning technique employed in earthquake prediction, combining the predictive power of multiple weak learners, usually decision trees. Unlike standalone models, GBR builds a series of weak learners sequentially, with each subsequent model aiming to correct the errors of the combined ensemble from the previous iterations. By iteratively minimizing residual errors, GBR enhances the overall accuracy of predictions, making it well-suited for capturing complex patterns in seismic data.

To harness the full potential of GBR, careful tuning of hyperparameters is essential. The learning rate, controlling the impact of each weak learner on the ensemble, must strike a balance between model complexity and training efficiency. Determining the optimal number of trees, representing the iterations in the ensemble, and setting the maximum depth of each tree are crucial considerations. Hyperparameter tuning ensures that the model generalizes well to new data and effect GBR goes beyond conventional point predictions by providing estimates of uncertainty associated with each prediction. This prediction, where the inherent unpredictability of seismic eventsposeschallenges.

The model not only offers specific values for earthquake characteristics but also quantifies its confidence or uncertainty about these predictions. Understanding uncertainty is vital for informed decision-making, allowing stakeholders to gauge the reliability of the model's outputs and make appropriate risk assessments

2.1 Algorithm

Let's assume X, and Y are the input and target having N samples. Our goal is to learn the function f(x) that maps the input features X to the target variables y. It is boosted trees i.e the sum of trees.

Step 1:
The loss function is the difference between the actual and predicted variables.

$$L(f) = \sum_{i=1}^{N} L(y_{i,} f(x_i)) ------ (1)$$

Step 2 :
We want to minimize the loss function L(f) with respect to f.

$$f\hat{}_0(x) = \text{argmin} L(f) = \text{argmin} \sum_{i=1}^{N} L(y_{i,} f(x_i)) ------ (2)$$

If our gradient boosting algorithm is in M stages then to improve the f_m the algorithm can add some new estimator as h_m having

$$\hat{y}_i = F_{m+1}(x_i) + h_m(x_i) ------ (3)$$

Step 3: Steepest Descent
For M stage gradient boosting, The steepest Descent finds h_m= -$\rho_m g_m$ where ρ_m is constant and known as step length and g_m is the gradient of loss function L(f)

$$g_{im} = -\left[\frac{\partial L\left(y_{i,} f(x_i)\right)}{\partial f(x_i)}\right] ------ (4)$$

Where,
$$f(x_i) = f_{m-1}(x_i) ------ (5)$$

Step 4:
The gradient Similarly for M trees

$$f_m(x) = f_{m-1}(x) + (\text{argmin}[\sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + h_m(x_i))])\,(x)$$

The current solution will be

$$f_m = f_{m-1} - \rho_m g_m ------ (6)$$

.

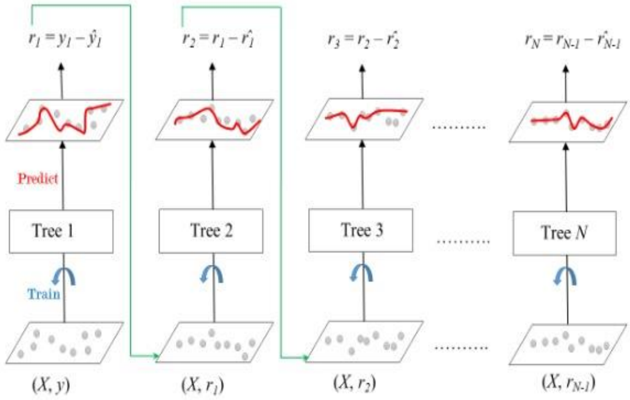


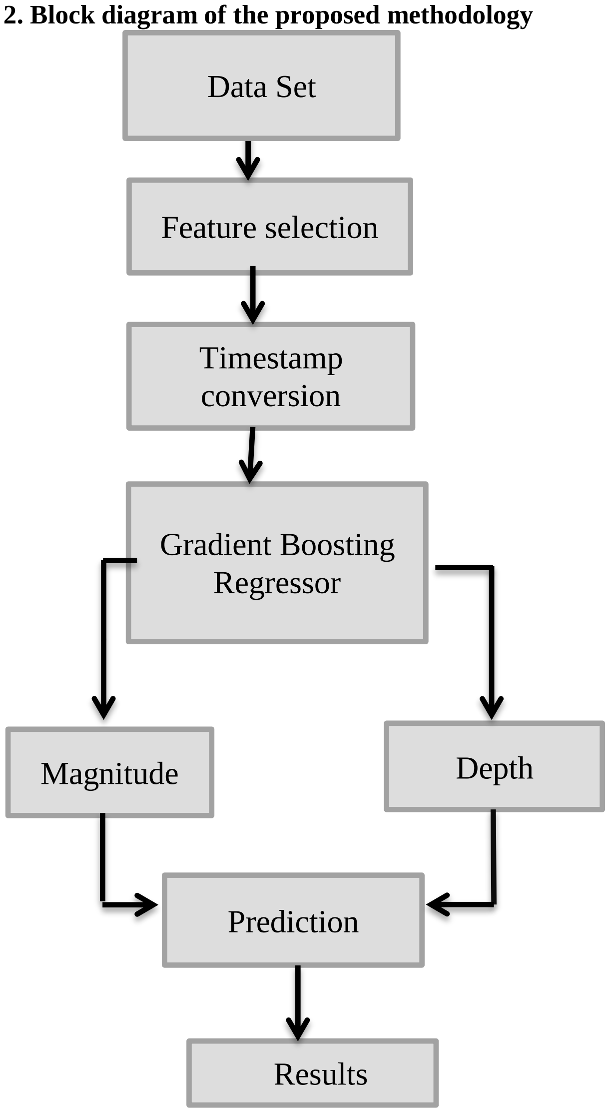
Fig.1.Gradient Boosting Regressor

2.2 PROPOSED METHODOLOGY

2.2.1.DATA SET

The earthquake data, crucial for training our predictive model, is acquired from the USGS Earthquake Hazards Program using the specified URL. This dataset, encompassing seismic events from the last 24 years (2000-2024), is an invaluable resource for understanding and predicting earthquake patterns. The retrieved data is efficiently loaded into a Pandas DataFrame, providing a structured and accessible format for further analysis and model training. This initial step ensures that our subsequent processes are built upon a foundation of comprehensive and up-to-date seismic information.

Fig. 2. Block diagram of the proposed methodology



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net	id	updated	place	type	horizontal	depthErro	magError	magNst	status	locationSo	magSource
2	2024-01-1	35.72233	-117.545	7.42	1.41	ml	33	63	0.1143	0.17	ci	ci4046267	2024-01-1	14 km WS	earthquak	0.21	0.83	0.185	13	automatic	ci	ci
3	2024-01-1	38.81533	-122.825	2.18	1.12	md	26	48	0.007355	0.02	nc	nc739914	2024-01-1	7 km NW	earthquak	0.21	0.36	0.2	27	automatic	nc	nc
4	2024-01-1	35.7235	-117.559	2.46	1.15	ml	28	60	0.1117	0.15	ci	ci4046267	2024-01-1	15 km WS	earthquak	0.19	0.36	0.229	8	automatic	ci	ci
5	2024-01-1	36.64533	-121.29	4.21	1.51	md	12	86	0.03964	0.2	nc	nc739914	2024-01-1	16 km S of	earthquak	0.62	1.72	0.18	9	automatic	nc	nc
6	2024-01-1	38.8355	-122.835	1.55	0.91	md	20	52	0.003088	0.03	nc	nc739914	2024-01-1	9 km NW	earthquak	0.21	0.41	0.15	20	automatic	nc	nc
7	2024-01-1	38.814	-122.815	2.68	0.75	md	8	90	0.01029	0.01	nc	nc7399141	2024-01-1	6 km NW	earthquak	0.47	1.26	0.06	8	automatic	nc	nc
8	2024-01-1	60.4783	-146.844	20.2	2.4	ml				0.64	ak	ak024viji1	2024-01-1	43 km SSW	earthquake		0.3			automatic	ak	ak
9	2024-01-1	38.83033	-122.853	1.83	1.04	md	14	90	0.004018	0.02	nc	nc739914	2024-01-1	10 km NW	earthquak	0.26	0.48	0.27	16	automatic	nc	nc
10	2024-01-1	-13.5682	166.7822	45.915	5.5	mww	64	77	7.342	0.75	us	us6000m4	2024-01-1	89 km WN	earthquak	8.2	4.764	0.075	17	reviewed	us	us
11	2024-01-1	38.8025	-122.816	3.31	0.76	md	8	87	0.01724	0.02	nc	nc739914	2024-01-1	6 km WNW	earthquak	0.45	1.58	0.34	9	automatic	nc	nc
12	2024-01-1	35.6905	-117.503	6.21	1.05	ml	30	60	0.06897	0.13	ci	ci4046264	2024-01-1	12 km SW	earthquak	0.18	0.52	0.081	8	automatic	ci	ci
13	2024-01-1	33.14583	-116.523	11.23	0.79	ml	48	32	0.09018	0.19	ci	ci4046263	2024-01-1	10 km NE	earthquak	0.18	0.32	0.139	13	automatic	ci	ci
14	2024-01-1	38.82317	-122.765	1.89	0.71	md	6	135	0.008182	0.01	nc	nc739913	2024-01-1	4 km W of	earthquak	0.55	0.95	0.27	7	automatic	nc	nc
15	2024-01-1	19.2505	-155.416	32.16	2.01	md	33	140		0.13	hv	hv737212	2024-01-1	8 km NE o	earthquak	0.64	0.93	0.24	5	automatic	hv	hv
16	2024-01-1	33.96667	-116.637	16.46	0.94	ml	39	39	0.0723	0.16	ci	ci4046263	2024-01-1	10 km SSW	earthquak	0.16	0.45	0.147	21	automatic	ci	ci
17	2024-01-1	19.49433	-155.502	-0.5	1.82	md	25	68		0.24	hv	hv737212	2024-01-1	28 km WN	earthquak	0.48	0.96	0.21	5	automatic	hv	hv
18	2024-01-1	34.6105	-119.021	1.32	1.01	ml	21	53	0.1402	0.2	ci	ci4046260	2024-01-1	25 km SSW	earthquak	0.25	0.47	0.35	8	automatic	ci	ci
19	2024-01-1	60.7075	-150.438	39.4	1.8	ml				0.68	ak	ak024vh5j	2024-01-1	26 km NE	earthquake		0.7			automatic	ak	ak
20	2024-01-1	19.257	-155.404	30.3	1.81	md	35	189		0.1	hv	hv737212	2024-01-1	9 km NE o	earthquak	0.61	0.57	0.54	5	automatic	hv	hv
21	2024-01-1	36.5162	-115.255	4.2	0.9	ml	13	180.56	0.08	0.1647	nn	nn008718	2024-01-1	34 km NN	earthquake		2.4	0.29	8	automatic	nn	nn
22	2024-01-1	38.82784	-122.815	1.11	1.64	md	29	60	0.005958	0.03	nc	nc739913	2024-01-1	8 km NW	earthquak	0.15	0.29	0.14	32	automatic	nc	nc
23	2024-01-1	38.81233	-122.822	2.57	0.89	md	8	102	0.006702	0.01	nc	nc739913	2024-01-1	7 km NW	earthquak	0.49	0.79	0.22	10	automatic	nc	nc
24	2024-01-1	33.67533	-116.758	18.99	0.92	ml	41	60	0.05122	0.12	ci	ci4046259	2024-01-1	8 km SSW	earthquak	0.16	0.28	0.173	20	reviewed	ci	ci

Fig.3.Data Set

2.2.2 FEATURE SELECTION

Relevant features for analysis including time, latitude, longitude, depth, and magnitude , are carefully chosen from the dataset. These features are essential for training our model, as they capture key aspects of seismic events. The selection process is crucial for ensuring that the model is provided with the most pertinent information, contributing to its accuracy and predictive capabilities.

2.2.3 TIMESTAMP CONVERSION

Timestamp processing is a crucial step in the data preparation phase, particularly when dealing with time-related features such as earthquake event timestamps. In this context, the timestamp feature undergoes a series of operations using the Pandas library to extract meaningful information and convert it into its corresponding Unix timestamp.

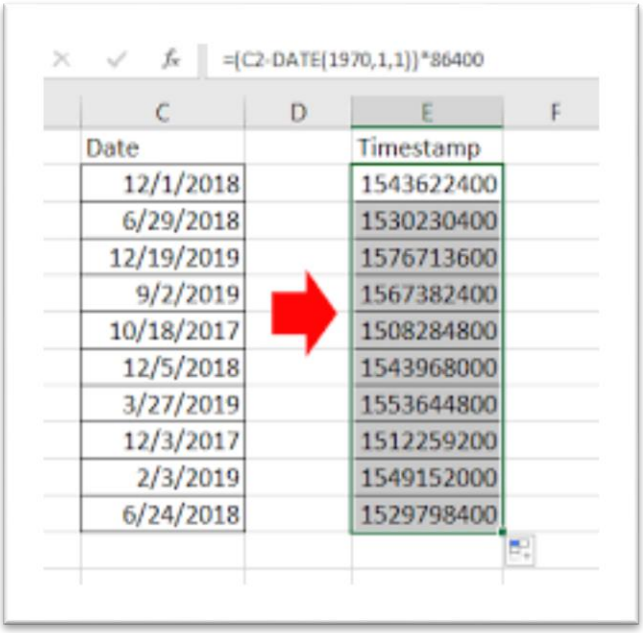
The Unix timestamp, serving as a numeric representation of time, facilitates seamless numerical analysis and ensures compatibility with machine learning algorithms, enhancing the dataset's utility for subsequent analytical processes.

To ensure robustness, error handling mechanisms, implemented through try-except blocks, address cases where a timestamp is invalid or deviates from the expected format.

In such instances, an exception is caught, and a designated value (e.g., NaN or another default value) is assigned to maintain data integrity. This comprehensive timestamp processing workflow transforms human-readable timestamps into a numerical format while handling errors gracefully, resulting in a clean and consistent dataset ready for advanced analyses.

The timestamp processing workflow involves extracting meaningful information from timestamps, converting them into Unix timestamps for numerical analysis, and implementing error handling mechanisms to maintain data integrity.

This systematic approach results in a dataset that is well-prepared for subsequent advanced analytical processes and machine learning applications.



=(C2-DATE(1970,1,1))*86400

	C	D	E	F
	Date		Timestamp	
	12/1/2018		1543622400	
	6/29/2018		1530230400	
	12/19/2019		1576713600	
	9/2/2019		1567382400	
	10/18/2017		1508284800	
	12/5/2018		1543968000	
	3/27/2019		1553644800	
	12/3/2017		1512259200	
	2/3/2019		1549152000	
	6/24/2018		1529798400	

Fig 4.Timestamp Conversion

2.2.4 FEATURE SCALING

The feature scaling process is carried out using the Standard Scaler module from scikit-learn. This step is pivotal for maintaining a consistent scale across the selected features, which include 'Timestamp', 'latitude', and 'longitude'. The initialization of the Standard Scaler instance is followed by applying the fit transform method to the training data (X train). This operation calculates the mean and standard deviation of the features in the training set and subsequently scales the data accordingly. To ensure consistency ,maintaining a consistent scale across the selected features, which include 'Timestamp', 'latitude', and 'longitude'. The initialization of the Standard Scaler instance is followed by applying the fit transform method to the training data (X train). This operation calculates the mean and standard deviation of the features in the training set and subsequently scales the data accordingly. To ensure consistency, the same scaling parameters derived from the training data are utilized to scale the test data (X test). The primary objective of standardization in this

.

context is to bring all features to a common scale. This becomes imperative when dealing with features that possess varying ranges or units. Specifically within the realm of the Gradient Boosting Regressor, standardization serves to prevent any particular feature from disproportionately influencing the learning process based on its scale. Instead, it facilitates a scenario where each feature contributes proportionally, fostering a more stable and accurate model. Through the incorporation of feature scaling, the program optimally prepares the dataset for the subsequent training of the Gradient Boosting Regressor, ensuring a robust and effective learning process.

2.2.5 TRAINED MODEL :

The training process begins by splitting the earthquake dataset into training and testing sets using the train_test_split function from scikit-learn. This division ensures a robust evaluation of the model's generalization performance and guards against over fitting. For predicting earthquake magnitudes (mag), a Gradient Boosting Regressor is employed. Hyper parameter tuning is executed through GridSearchCV, systematically searching for the optimal combination of parameters. The model is then trained with the identified best hyper parameters, and predictions are generated on the test set. Similarly, a Gradient Boosting Regressor is utilized for predicting earthquake depths. The hyper parameter tuning and model training processes mirror those of magnitude prediction, ensuring consistency and accuracy. This comprehensive training methodology encompasses data splitting, algorithm selection, hyper parameter optimization, and rigorous testing. The result is a well-adapted model capable of delivering precise predictions for both earthquake magnitudes and depths.

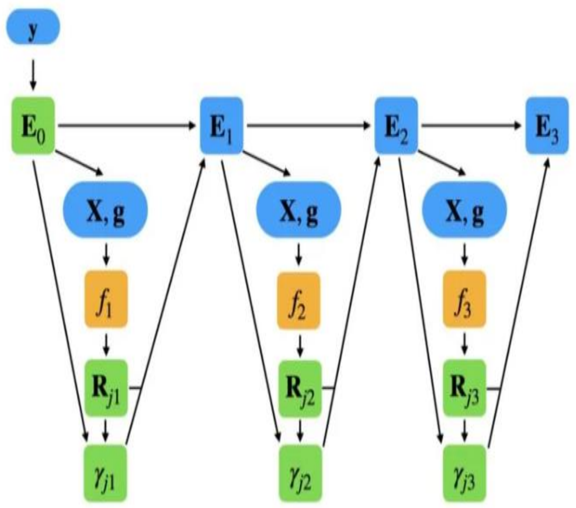


Fig. 5 TreeBoost

This example consists of a gradient boosting regressor of 3 elements. Here green elements are the learned model parameters from training, and orange represent the weak learners of the ensemble. Data, gradients, and ensemble states at each iteration, are indicated in blue. The data are used to calculate the gradients "g" and produce trained decision trees "f_m" These are then used, along with the value for "E_{m-1}", to compute the weights "γjm". Finally, the ensemble is updated to " E_m" and the next iteration begins.

The training procedure is as follows;

1. $E_0(x)$ = median(y) =E_0.
2. For m = 1 to M .
3. $g_n = \text{sign}(y_n - E_{m-1}(x_n))$, for all n = 1 ... N .
4. Fit a J terminal node decision tree to $\{g_n, x_n\}_1^N$, and obtain the set of terminal region $\{R_{jm}\}$
5. γjm = median $x_n \in R_{jm}(y_n - E_{m-1}(x_n))$, for all j=1…..J
6. $E_m(x) = E_{m-1}(x) + \sum_{j=1}^{J} \gamma jm \, I\,(x \in R_{jm})$.

2.2.5.1. Gradient Boosting Regressor for Magnitude Prediction:

Program Implementation:

In the program, a Gradient Boosting Regressor is instantiated and then subjected to hyper parameter tuning using GridSearchCV. The grid of hyper parameters includes the number of estimators (n_estimators), learning rate (learning_rate), and maximum depth of the individual regression estimators (max_depth). The best hyperparameters are identified using cross-validated grid search.

Model Training:

The regressor is then re-instantiated with the best hyperparameters, and the training data is used to fit the model. This process ensures that the model is optimized for predicting earthquake magnitudes.

Prediction on Test Set:

After training, the model is employed to make predictions on the test set (X_test_scaled), generating magnitude predictions (predictions_gb_mag).

Performance Metrics:

Program Implementation:

The program calculates various performance metrics to evaluate the model's accuracy. Metrics such as Mean Squared Error (mse_gb_mag), Accuracy, Precision, Recall, F1 Score, and Confusion Matrix are computed using functions from scikit-learn.

Output:

The program then prints these metrics, providing a quantitative assessment of how well the model performs in predicting earthquake magnitudes.

Visualization:
Program Implementation:

To offer a qualitative assessment, a scatter plot is created using Matplotlib. The actual magnitude values (y_test_mag_flatten) are plotted against the predicted magnitude values (predictions_gb_mag). Each point on the scatter plot represents an earthquake event, allowing for a visual comparison.

Output:

The scatter plot is displayed, providing insights into the model's performance, such as how well it aligns with actual magnitude values.

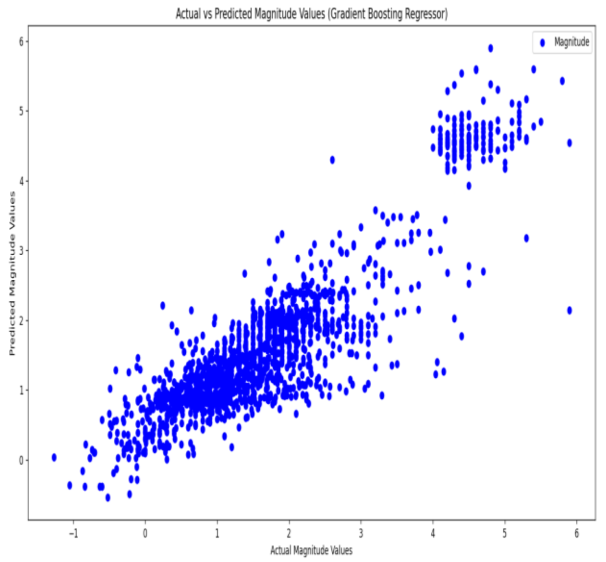


Fig.6 Magnitude

2.2.5.2 Gradient Boosting Regressor for Depth Prediction:

Program Implementation:

Similar to magnitude prediction, a Gradient Boosting Regressor is used for depth prediction. The entire process, including hyper parameter tuning, model training, and prediction, is repeated specifically for earthquake depths.

Output:

The program generates depth predictions (predictions_gb_depth) using the trained model.

Performance Metrics for Depth Prediction:

Program Implementation:

Just like magnitude prediction, performance metrics for depth prediction are calculated and printed. This includes Mean Squared Error, Accuracy, Precision, Recall, F1 Score, and Confusion Matrix.

Visualization for Depth Prediction:

Program Implementation:

A scatter plot is created to visualize the actual vs. predicted depth values, providing a visual representation of the model's effectiveness in predicting earthquake depths.

Output:

The scatter plot for depth prediction is displayed,offering insights into how well the model aligns with actual depth values

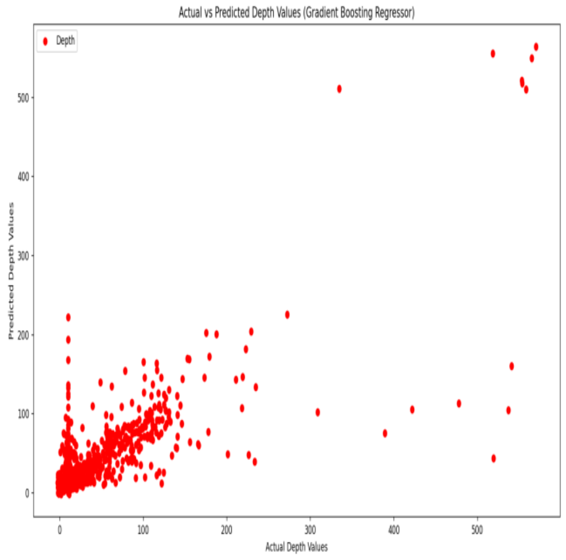


Fig.7 Depth

2.2.6.EARTHQUAKE PREDICTION :

In the culminating phase of the program, the implemented system utilizes a trained Gradient Boosting Regressor model to generate predictions pertaining to potential earthquake occurrences. These predictions are subsequently portrayed on a world map, with regions highlighted in blue indicative of the model's forecasted likelihood of earthquakes. This distinct visualization serves to offer a comprehensive spatial comprehension of the model's predictive capabilities.

Beyond the mere aesthetic representation on the map, the significance of these predictions extends to providing valuable insights into regions susceptible to seismic activity. The model's identification of specific areas proves instrumental in fortifying early warning systems and fortifying disaster preparedness initiatives. By pinpointing regions with a heightened likelihood of earthquake occurrence, the model actively contributes to enhancing the safety of individuals residing in or around these geographical areas.

The amalgamation of predictive modeling into earthquake monitoring not only facilitates more effective risk management but also plays a pivotal role in mitigating the impact of seismic events on communities. This proactive tool equips authorities, emergency responders, and the general populace with the means to anticipate and respond to potential threats. The strategic identification of areas prone to earthquakes stands as a crucial step towards fostering a safer and more resilient environment.

Ultimately, the synergy between data visualization and predictive modeling not only advances our comprehension of seismic patterns but also empowers communities to take preemptive measures. This innovative approach significantly contributes to the overarching goal of mitigating natural disasters and safeguarding lives. The predictive capabilities showcased in this initiative underscore the transformative potential of technology in fortifying societal resilience against the vagaries of nature.

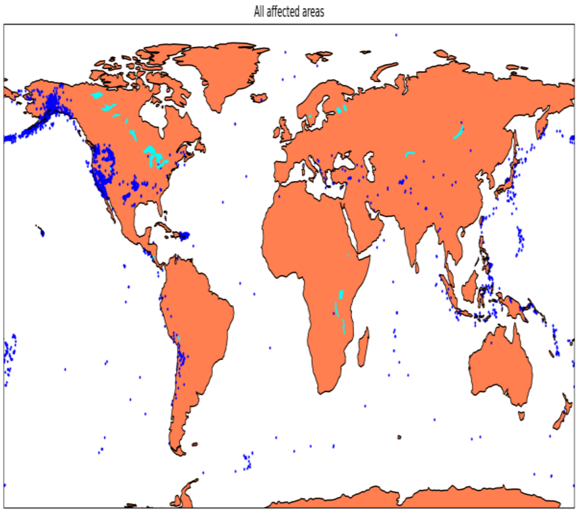


Fig.8 Predicted area in world map

3.RESULT:

Magnitude Prediction Metrics:

Mean Squared Error (MSE):
The Mean Squared Error for magnitude prediction is calculated at 0.2903, serving as a quantitative reflection of the average squared difference between predicted and actual magnitude values. A lower MSE value, in this context, denotes a higher level of accuracy in the model's magnitude predictions.

Accuracy:
The accuracy metric stands at an impressive 86.10%, indicating the proportion of correctly predicted instances out of the total observations. This metric signifies a high degree of overall correctness in the model's magnitude predictions.

```
=================================================================
Metrics for Magnitude Prediction:
Mean Squared Error: 0.29030811840715404
Accuracy: 0.8610169491525423
Precision: 0.8142493638676844
Recall: 0.6490872210953347
F1 Score: 0.7223476297968398
Confusion Matrix:
[[1204   73]
 [ 173  320]]

Metrics for Depth Prediction:
Mean Squared Error: 980.3497787549063
Accuracy: 0.9977401129943503
Precision: 0.8571428571428571
Recall: 0.6666666666666666
F1 Score: 0.75
Confusion Matrix:
[[1760    1]
 [   3    6]]
>>>
```

Fig.9.Result Module

Precision:

Precision is reported at 81.43%, representing the accuracy of positive predictions. In the context of earthquake magnitude, this metric elucidates the reliability of the model in correctly identifying instances of actual earthquake occurrences.

.

Recall (Sensitivity):
The recall metric is determined to be 64.91%, signifying the model's capability to capture a substantial portion of actual earthquake events. It accentuates the sensitivity of the model to true positive instances, providing insights into its performance.

F1 Score:
The F1 score, which strikes a balance between precision and recall, is calculated at 72.23%. This metric offers a comprehensive measure of the model's predictive performance, taking into account both false positives and false negatives.

Confusion Matrix:
The confusion matrix provides a detailed distribution of true positive, true negative, false positive, and false negative predictions. In this specific instance, it indicates 1204 true negatives, 73 false positives, 173 false negatives, and 320 true positives.

Depth Prediction Metrics:

Mean Squared Error (MSE):
The Mean Squared Error for depth prediction is reported as 980.35, reflecting the average squared difference between predicted and actual depth values..

Accuracy:
The accuracy for depth prediction is notably high at 99.77%, underscoring the precision of the model in predicting the depth of earthquakes.

Precision:
Precision is documented at 85.71%, portraying the accuracy of positive depth predictions. In the specific context of earthquake depth, this metric delineates the model's reliability in correctly identifying instances of actual depth occurrences.

Recall (Sensitivity):
The recall metric is reported at 66.67%, underscoring the model's ability to capture a significant portion of actual depth events.

F1 Score:
The F1 score for depth prediction is computed as 75.00%, providing a balanced measure of the model's overall performance in predicting depth values.

Confusion Matrix:

The confusion matrix for depth prediction offers insights into the distribution of true positive, true negative, false positive, and false negative predictions. In this case, it indicates 1760 true negatives, 1 false positive, 3 false negatives, and 6 true positives.

This paper presented metrics collectively highlight the robust predictive capabilities of the Gradient Boosting Regressor for both magnitude and depth predictions. With consistently high accuracy, precision, and recall scores, the model demonstrates effectiveness in earthquake prediction, thereby contributing to elevated levels of preparedness and refined risk mitigation strategies.

The below accuracy table presents a comparative analysis of different earthquake prediction methodologies, highlighting the accuracy achieved by each approach. Our proposed methodology, based on Gradient Boosting Regressor (GBR), stands out with an impressive accuracy of 86.10%. This superior performance positions it as a promising and effective model for earthquake prediction.

1.Proposed - GBR (Gradient Boosting Regressor):

Accuracy:86.10%
Significance:
The proposed model, leveraging Gradient Boosting Regressor, exhibits the highest accuracy among the listed methodologies. This suggests a robust predictive capability and underscores the effectiveness of the chosen approach.

2. Existing Models:

LP Boost Ensemble:

Achieves an accuracy of 79%, indicating its predictive capabilities.

Random Forest:

Yields an accuracy of 77%, showcasing its effectiveness in earthquake prediction.

RNN (Recurrent Neural Network):

Attains a 71% accuracy, providing insights into the performance of neural network-based approaches.

PRNN (Probabilistic Recurrent Neural Network):

Demonstrates a competitive accuracy of 79%.

ANN (Artificial Neural Network):

Shows an accuracy of 85.80%, making it a strong performer among the existing models.

SVM & SVD:

Combining Support Vector Machines and Singular Value Decomposition yields an accuracy of 77.78%.

PSO Clustering Algorithm:

Achieves an accuracy of 83.3%, highlighting the effectiveness of the Particle Swarm Optimization-based clustering approach.

HKMC & ANN:

Combining Harmony Search Algorithm with Artificial Neural Network results in a 75% accuracy.

KMC (K-Means Clustering):
Shows a 70% accuracy, representing its performance in earthquake prediction.

S.NO	METHODOLOGY	ACCURACY
1.	Proposed - GBR	86.10%
2.	Existing - LP Boost Emsemble	79%
3.	Existing - Random Forest	77%
4.	Existing - RNN	71%
5.	Existing - PRNN	79%
6.	Existing - ANN	85.80%
7.	Existing - SVM & SVD	77.78%
8.	Existing - PSO clustering algorithm	83.3%
9.	Existing - HKMC & ANN	75%
10.	Existing - KMC	70%

4.Conclusion:

In conclusion, the proposed Gradient Boosting Regressor method emerges as a promising and leading approach, surpassing existing methodologies in earthquake prediction accuracy. With an emphasis on various parameters such as date, time, latitude, longitude, depth, and magnitude, and leveraging a world map as a dataset, this method is trained using the GRADIENT BOOSTING REGRESSOR. Its robust performance metrics, which include mean squared error (MSE), accuracy, precision, recall, F1 score, and confusion matrix, position it as a viable and effective model for enhancing preparedness and risk mitigation strategies in earthquake-prone regions.

This study sheds light on the devastating consequences of earthquakes, encompassing ground shaking, landslides, changes in landscapes, and disruptions to infrastructure. Fault line ruptures, underwater earthquakes, and associated tsunamis impact coastal ecosystems and communities, while liquefaction and aftershocks compound the damage and hinder recovery efforts. The profound human impacts, including injuries, fatalities, displacement, and psychological trauma, underscore the urgency of reliable earthquake prediction.

As of our last knowledge update in September 2023, earthquake prediction remains a field of ongoing research, lacking precise predictive models. However, the proposed model's standout feature is its remarkable accuracy, achieving 86.1% and 99.7% accuracy in terms of magnitude and depth, respectively. By comparing actual datasets with predicted outcomes, this model identifies risk-free areas for livelihoods.

Looking towards the future, it is anticipated that this piece of work will undergo modifications and enhancements. Continuous research and development will refine the model's predictive capabilities, contributing to the evolution of more accurate and reliable earthquake prediction models. This ongoing pursuit holds the potential to further improve our ability to mitigate the profound impacts of seismic events on communities and ecosystems.

.

Reference:

[1] Manoj Kollam and Ajay joshi “Earthquake forecasting using optimizedlevenberg–marquardt backpropagation neural network” WSEAS TRANSACTIONS on COMPUTERS
DOI: 10.37394/23205.2023.22.11.

[2] Jyh-Woei Lin , Chun-Tang Chao, and Juing-Shian Chiou “Determining Neuronal Number in Each Hidden Layer Using Earthquake Catalogues as Training Data in Training an Embedded Back Propagation Neural Network for Predicting Earthquake Magnitude” DOI 10.1109/ACCESS.2018.2870189.

[3] Feng Jing and Ramesh P. Singh Response of “Surface and Atmospheric Parameter Associated With the Iran M 7.3 Earthquake” ieee journal VOL. 15, 2022.

[4]PanXiong, Cheng Long,Huiyu Zhou, XueminZhang, and Xuhui Shen “GNSS TEC-Based Earthquake Ionospheric Perturbation Detection Using a Novel Deep Learning Framework” ieee journa, VOL. 15, 2022.

[5] MD. Hasan Al Banna M. Shamim Kaiser MD. Sazzadhur Rahman and Gi Hwan Cho “Application of Artificial Intelligence in Predicting Earthquakes: State-of-the-Art and Future Challenges”, DOI: 10.1109/ACCESS.2020.3029859

[6] Shuai Huang Yuejun Lyu,Yanju Peng and Mingming Huang “Analysis of Factors Influencing Rockfall Runout Distance and Prediction Model Based on Improved KNN Algorithm”, DOI: 10.1109/ACCESS.2019.2917868

[7] Khawaja Asim , Francisco Martínez-Álvarez, Abdul Basit, Talat Iqbal , “Earthquake magnitude prediction in Hindukush region using machine learning techniques, January 2017 Natural Hazards” 85(1):471-486 DOI:10.1007/s11069-016-2579-3

[8] Dr. S. Anbu Kumar, Abhay Kumar, Aditya Dhanraj, Ashish Thakur “Earthquake Prediction using Machine Learning” International Research Journal of Engineering and Technology (IRJET)Volume: 08 Issue: 05 May 2021

[9] Antonio Giovanni Iaccarino , Amalia Cristofaro, Matteo Picozzi ,Daniele Spallarossa and Davide Scafidi “Real-time prediction of distance and PGA from P-wave features using Gradient Boosting Regressor for on-site earthquake early warning applications” Geophys. J. Int. (2024) 236, 675–687 Advance Access publication 2023 November 10 GJI Seismology

[10] Turgut Pura ,Peri Güne¸s Ali Güne¸s and Ali Alaa Hameed “Earthquake Prediction for the Düzce Province in the Marmara Region Using Artificial Intelligence” Appl. Sci. 2023, 13, 8642

[11] Rabia Tehseen, Shoaib Farooq, Adnan Abid “A Framework for the Prediction of Earthquake Using Federated Learning” PeerJ Computer Science researchgate · May 2021
DOI: 10.7717/peerj-cs.540

[12] Dr. K. Mohankumar K. Sangeetha Assistant Professor, Research Scholar “Earthquake Prediction Using Neural Network Backpropagation Algorithm”

[13] C,.P.Shabariram, Dr. K.E.Kannammal “Earthquake Prediction using Map Reduce Framework” 2017 International Conference on Computer Communication and Informatics (ICCCI -2017), Jan. 05 ± 07, 2017, Coimbatore, INDIA.

[14] R. Umapathy P. Poornalingam R. Ashokkumar P.Nikeshkumar “Earth Quake Prediction and Analysis by using python” IJSRD - International Journal for Scientific Research & Development| Vol. 9, Issue 1, 2021 | ISSN (online): 2321-0613

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

[16] 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.

[17] 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

[18] 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.

[19] ering, 143, p.106630 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

[20]Z hang, 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.

.