iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.68283

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Major Earthquake Event Prediction Using Various Machine Learning Algorithms

Dr. M.A. Manivasagam[1], M Ramya[2], K Sri Vidya[3], K Siva Rami Reddy[4], T Surya Narayana[5], S Surendra Naidu[6]
[1]*Professor, Department of Computer Science and Engineering, Siddartha institute of science and technology, Puttur-517583, A.P., India*
[2, 3, 4, 5, 6]*Department of Computer Science and Engineering, Siddartha institute of science and technology, Puttur-517583, A.P., India*

Abstract: ***According to the extent or time period taken to foretell the occurrence of an earthquake, two primary classifications of methods are used. In other words, short-term maps are generated over the course of the few hours or, at most, days while maps that are made months or even years in advance are referred to as forecasts. The majority of research is aimed at the predictions that take into account the history of the earthquake in the given regions or countries. The objective of the present investigation is to identify if an earthquake event is positive or negative using different algorithms. The implementation of Random Forest, Naïve Bayes, Logistic Regression, Multi-Layer Perceptron, AdaBoost, K-Nearest Neighbors, SVC using Support Vector Machine, and Classification and Regression Trees has been performed to a real earthquake data set. Different hyperparameters for each algorithm were selected and the accuracy of each of them was compared based on several measures. The features were identified to have average values of 0.89, 0.89 and 0.83 for significant earthquake activities by three of the models. Some of the other algorithms that have been used in conjunction with the proposed approach are Random Forest, Naïve Bayes, Logistic Regression, AdaBoost, KNN, Support Vector Machine and Multi-Layer Perceptron Classifier.***
Keywords: Earthquake Regions, Machine Leaning, Random Forest, Support Vector Machine, Classification.

I. INTRODUCTION

Earthquakes make up some of the most destructive forces of nature-botheredly followed by structural damages, economic effects, as well as loss of human lives. Yet these events remain under even the highest possible prediction levels due to their unpredictability, which makes them major areas of research in science currently. Traditionally, there are two approaches to earthquake forecasts: short-term, pertaining to forecasts within hours or days, and long-term, relating mainly to predicting an event for months or years. Short-term predictions are mainly derived from activities manifestation like frequency, intensity, and location of tremors, while long-term forecasts assess earthquake risk based on geological data with historical trends. Geological studies that deal with past earthquake patterns, fault lines, and seismic waves have been mainly used for earthquake predictions. The new emergence of machine learning (ML) techniques adds another promising way to improve earthquake prediction accuracy. Very critical patterns are expected to be obtained through this exploitation of the capability that machine-learning-based methods have to analyze historical earthquake data and tend to understand the trends associated with seismic activity due to its ability to uncover very complex patterns within the vast datasets. Such research aims to study the feasibility of using the classification of seismic events occurrence or nonoccurrence. Such algorithms are expected to have the ability and capacity to process large datasets and perform complex classification tasks resulting from their ability to disclose insights into factors that influence the occurrence of earthquakes. The real earthquake data was utilized in testing the models, and comparative performance metrics were observed. Hyperparameters for each algorithm were optimized to obtain maximum performance. The Random Forest, Naive Bayes, and SVM models gave the best results when significant seismic events were detected. This indicates that machine learning may succeed in improving earthquake prediction and allows providing an alternative or complementary method to those currently employed. As research progresses, further strengths and weaknesses of each of the algorithms will be emphasized in the earthquake prediction context with regards to machine learning in overcoming the unpredictability issue of natural disasters. Fortified disaster preparedness and mitigation methods are attainable if such machine learning models are coupled with any of the traditional geological models to build the real-time systems for more accurate earthquake prediction.

A. Problem Statement

Earthquake event prediction is among the most difficult problems that seismology and geophysics present because the factors influencing seismic activity are so complex. Although much progress has been made in the technology of monitoring, it is, however, very difficult to predict accurately when and where an earthquake would occur.

Methods used in predicting earthquakes traditionally include statistical analysis of historical records and geological observations, which are subjected to insufficient precision and reliability.

Emerging need exists to go beyond these conventional techniques mainly toward utilizing machine learning models that would improve the prediction of earthquake occurrence. This research primarily aims to determine whether an earthquake event will occur, based on previous earthquake data, using several machine-learning algorithms. Using algorithms, such as Random Forest, Naïve Bayes, Logistic Regression, AdaBoost, K-Nearest Neighbors, Support Vector Machine, and Multi-Layer Perceptron, this research tries to evaluate whether earthquake prediction would be possible with improved accuracy. The main challenge is to select the best features and hyperparameter optimization of the algorithms and then compare their model outcome in forecasting major seismic events in different geographical regions. This is focused on improving the prediction methods for earthquakes concerning preparedness and mitigation of associated risks.

II. LITERATURE SURVEY

Earthquake prediction has always generated curiosities in seismological and geophysical areas, with several researchers proposing various methods to enhance prediction efficacy. While classical earthquake[1] prediction methods can be differentiated into long-term predictions providing forecasts for seismic activity over months and years and short-term predictions forecasting[2] earthquakes occurring in the near future (ranging from hours to days), knowledge advancement has not helped much in via predictions [3]and the nature of seismic events perceived to involve a lot of intricacy and the ambiguous understanding of the processes involved. [4]

Forecasting tremors has proved difficult for several decades as seismic activity has extremely complex and unpredictable phenomena. [5]Several efforts have been made to develop a tool to forecast earthquakes, most of which are studies of historical data to predict the occurrence of earthquakes. [6]Short-term predictions would range from a few hours to several days defined by the classification of earthquake prediction methods, while long-term predictions could last from a number of months even to years. Most of the research in this field is centered around short-term earthquake prediction using machine learning techniques to analyze seismic data and historical events of earthquakes occurring in different regions.[7]

Different machine learning methods have been implemented to improve the predictiveness of earthquakes. Random forests have been largely used in many classification studies with the seismic signals or historical patterns of seismic records to predict event occurrences. [8]The other classifier used widely in seismic prediction models is Naïve Bayes, which has been applied for analyzing the probability factor[9] occurrence of the earthquake event according to historical data. Also, logistic regression that courses the relationships between earthquake activity and other external environmental variables has produced encouraging results for predicting seismic events.[10]

Other studies utilized advanced techniques of machine learning including Multi-Layer Perceptron (MLP) which is basically a model corresponding[11] to neural networks principles but employs a non-linear input-output mapping for capturing complex degrees of association between input features with earthquake occurrences. For the prediction prowess improvement by readingjusting [12]the weights of misclassified cases for emphasis in prediction, AdaBoost a boosting algorithm was also used with the classifiers stated above. [13]. [14]

These are just some of the algorithms, on top of these CART which is useful in decision boundary[15] separation and interpretation regarding the decision made [16] the accuracy of the model while evaluations are done[17] using metrics of precision, recall, F1-score, and accuracy.

The results compared models giving finding that combinations of these techniques can yield an even better prediction than possible with just one This adds to the work of research done towards exploring machine learning as an improving feature of the advance of future earthquake prediction .

To conclude, much has been achieved in applying machine learning techniques toward earthquake prediction, and challenges still arise seismic activity's ever-changing nature. However, with improvement in computational capacities and accessibility of better datasets, one expects machine learning methods to significantly contribute to the restructuring of earthquake prediction systems through timely alerts to mitigate damage. This study intends to enhance research in this area by testing several algorithms: Random Forest, Naïve Bayes, Logistic Regression, and so on, to predict earthquakes and have more accurate and reliable systems for earthquake forecasting.

III. METHODOLOGY

A. *Proposed System*

This application deals with what may be called earthquake prediction through machine learning techniques to increase the accuracy and time of prediction. The intended incorporation of Random Forest, Naïve Bayes, Logistic Regression, Multi-Layer Perceptron, A daBoost, K-nearest neighbors (KNN), Support Vector Machine (SVM), and Classification and Regression Trees (CART) into a system has been developed by using actual earthquake datasets. In addition to that, it will employ different evaluation measures to compare the achieved accuracies of prediction made out by these algorithms. It aims for the development of a strong prediction mechanism for earthquake instances through the combination of various machine learning models improving preparedness and responses in earthquake-vulnerable areas. A friendly interface will also be incorporated into the system for easy immediate visual availability of prediction and in other earthquake-related data. Thus producing forecasts and real-time alerts that would depend on the predictions and timing of actions. Besides, the system will include the feedback loop through which the system could learn to improve model accuracy whenever new incoming data and predictive models were refined.

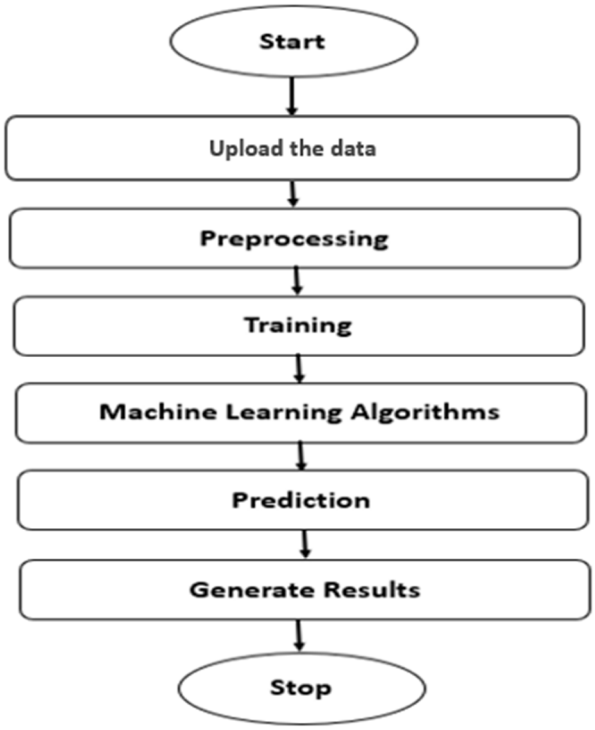
IV. IMPLEMENTATION

A. *Dataset*

These types of historical events data relevant for this research are earthquake event sources. Seismic activities and earthquakes include various features in the database. Sources of the data include geologic surveys by government institutions, a network that collates data from ongoing and historical earthquakes, and scientific research institutions. Another example of an attribute would be an earthquake's coordinates, how deep it was, or at what time it occurred. They vary from one place to another. Such a dataset usually spans many years and covers many significant and non-significant events. The entire picture of earthquake occurrences is known at different locations. All features are pre-processed according to missing data, outliers, and normalization. Gist of the dataset: divided into training Several machine learning algorithms can, therefore, be applied for developing patterns and predicting earthquakes using this heterogeneous dataset.

B. *Pre-processing Steps*

Machine learning algorithms are to be applied, so the next step will be towards actualizing the processes that help assure quality and consistency in data. The process will consist of the following:



1) Data Cleaning: For the cleaning of data, all missing values were treated using imputation techniques. Unwanted features were eliminated, retaining only the most relevant.
2) Data Transformation: Normalization of numerical features is done in a standard scale. This makes sure that for the algorithm working on its features, the different scales may introduce bias into the algorithms, which can very much depend on the feature scale.
3) Categorical Variable Encoding: Categorical variables such as the different regions of earthquakes are converted into relevant forms for machine learning models using techniques like one-hot and label encodings.
4) Feature Selection: Some features selected might have less correlation with the outcome or be too redundant; hence, these features are discarded so that model performance can be boosted with reduced dimensionality.
5) Data Splitting: The dataset was generally partitioned into a training and testing .

C. *Model Training*

1) *Logistic Regression*

Logistic regression has been around since the early twentieth century, where it found its place in research in the biological sciences, followed later by applications in the social sciences. Logistic regression is supposed given the categorical nature of the dependent variable (target). Examples:
To determine whether the email is spam (1) or not (0)
To determine whether the tumor is malignant (1) or not (0)
Consider a situation to classify the email as spam. In this exercise, if we use linear regression, the problem will be arise in fixing a threshold for classification. Consider, if this data point is an actual malignant, with predicted continuous value 0.4 and threshold value 0.5, the data point will be classified as not malignant, which would transaction a winning ticket in real time.
From that one could obviously conclude that linear regression cannot do classification. Linear regression provides estimates outside the limits of 0 to1; hence on comes the motivation for the application of logistic regression."

Reasons and examples of logistic regression usage:
Logistic regression is extensively used in most machine learning algorithms related to binary classification problems with two class values-the necessary output will be a prediction, such as either this or that, yes or no, and A-or-B.
where the predictor has a continuous outcome in terms of hours of study and the response has two values, namely pass or fail.
Then the organization can use these insights from logistic regression results to fine-tune their own business strategies on how to meet the several business goals-either lowering expenditures or losses or increasing the ROI on marketing campaigns.
So consider an e-commerce company that sends costly promotional offers to customers. It would want to know whether a particular customer would respond to these offers or not. The most important part of response prediction that they would be interested in would be to know if that person will be a responder or non-responder.This is simple propensity to respond modeling in marketing.

2) *Random Forest Classifier*

The random forest method is, in extremely simple terms, one of the techniques of machine learning that can be applied to regression and classification problems.
For difficult problems to solve, it is classified as an ensemble learning technique and is realized in random forests as a collection of decision trees. The training part carried out in random forests is done using a technique referred to as bagging or bootstrap aggregating.
Bagging is an ensemble meta-algorithm that reduces variance on algorithms with supervision in machine learning. Random forest output depends on the predictions of decision trees.
It works on the principles of averaging the output from different trees. Therefore, it can also be said that more trees contributed, the better would be the accuracy of the output.
Random forest has been potentially effective in addressing the drawbacks of decision tree algorithms concerning overfitting tendencies of datasets and thus improving accuracy. The random forest gives more accurate predictions with reduced tuning in Scikit-learn packages and without the necessity of tuning.

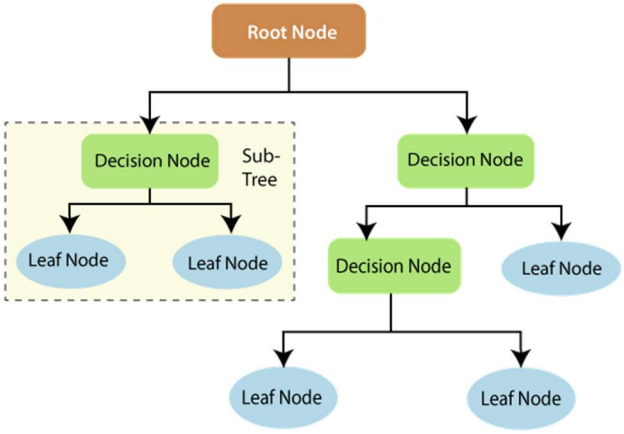
Properties of a Random Forest Algorithm:

- It has much higher accuracy than the decision tree algorithm
- Its performance towards missing data is exceptional
- Reasonable predictions are available without tuning any hyper-parameters
- It solves overfitting with decision trees
- A random subset of features is selected at the node's splitting point for each tree in a random forest

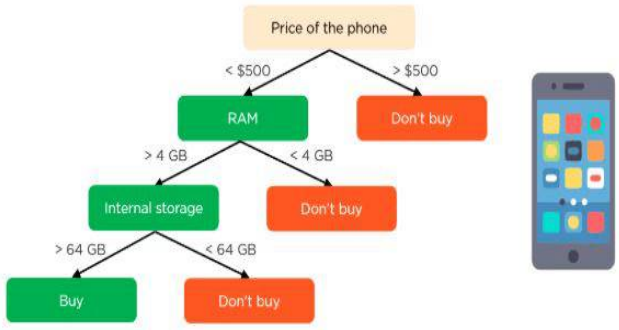
The trees are essentially the building blocks of such an algorithm, with the kind of model capable of functioning as a decision support system that uses a tree-like graph. Familiarizing oneself with the basics of decision trees would help understand the underlying mechanism of random forest.

A decision tree is comprised of three parts: decision nodes, leaf nodes, and a root node. The decision-tree algorithm branches a training data set into a number of smaller sets while also incrementally developing the tree associated with them. This process goes on until there is a leaf node that is no longer dividable.

The decision tree contains nodes that represent the values, which are features that are used to predict the outcome. Decision nodes are directly connected to the leaves. The following diagram shows the three types of nodes in a decision tree.



This is sometimes referred to as informational theory, and it is this basic concept through which one can understand how decision trees function. The tree is erected on the twin pillars called target variable (i.e., class) using its independent variables (i.e., features). This can be calculated by considering the value of entropy of the target variable (Y) and the conditional entropy of Y given X. Conditional entropy is subtracted from the entropy of Y. Information gain thus contributes to the training of decision trees and, as such, helps in resolving the uncertainties of that tree. High information gain means it reduced a lot of uncertainty (information entropy). Hence, entropy and information gain have become critical properties defining how to split nodes-a crucial step in building decision trees. For instance, a decision tree could tell if a customer will buy a phone or not. The relevant attributes, therefore, are those on which the customer decides whether or not to buy. That is more easily shown in the graphical representation of a decision tree: the root node is decision nodes, which index customer characteristics about which he/she responded concerning the phone. The leaf node indicates the final answer to buying versus not buying that product. The important parameters influencing the decision are price, internal storage, and RAM. This is how the decision tree would look like.



The four features which may affect customers' decisions are represented by the root nodes-the price, internal memory, camera, and RAM. The random forest would divide the nodes based on these features selected randomly. The result of the four trees will make the predicted result.

The consequence of most decision trees will make the final decision. Here, out of the three trees predicting buying and one predicting not buying, the final decision will be buying. It then predicts that the customer will buy the phone.

3) Naïve Bayes Classifier

Naive Bayes classifier is a probabilistic model in machine learning used to solve classification problems. The model defines the Bayes theorem as its core.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

According to Bayes' theorem, one can calculate the probability of A given B. In this case, it considers B to be evidence and A to be hypothesis. The other major assumption in outline here is the assumption of independence among the predictors or the features. That is, the presence of one specific feature would not lend any influence on another feature. Therefore, it is called 'naive.'

Let us take an instance to illustrate this. Below is a dataset of training weather data and its corresponding target variable `Play', which suggests possibilities of playing. Now on the basis of weather, we need to classify whether the players will play or not. To perform this let us follow the steps below.

Step 1: Convert the data set into a frequency table.

Step 2: In the Likelihood table, probabilities will be found by calculating Overcast probability = 0.29 and probability of playing = 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Step 3: Now breathe again, where you will use the Naive Bayesian upon the posterior for every class. The output of that class will be the one that happens to have the highest posterior probability.

Issue: Clear is the condition. Players will only play when the weather is sunny. Is this true statement?

The above formula serves the purpose of getting the posterior probability as previously explained. P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P(Sunny).

Here, we have P(Sunny | Yes) = 3/9 = 0.33; P(Sunny) = 5/14 = 0.36; P(Yes) = 9/14 = 0.64. And so we get P(Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has a higher probability.

Naive Bayes will predict probabilities of different classes against different attributes. Thus, it finds its applications in text classification and also in cases of more than two classes.

• Thus, it will be an easy and quick process to classify the predicted class of the test data set. Fairly good performance is also shown in multiclass prediction.

• The naive Bayes classifier does go very well if the independence assumption holds as compared to other models like logistic regression and requires much lesser training data.

• This is quite and pretty efficient against categorical input variables but not so much when it comes to (numerical variable(s)). For numerical variable, we assume a normal distribution (bell curve, which is a very strong assumption).

4) AdaBoost

AdaBoost, or Adaptive boosting, is an instance of boosting algorithms, which in full is an ensemble method in machine learning. It is called adaptive because, with every instance, the weights of all the instances are updated, giving higher weightage to incorrectly classified instances. Supervised boosting is thus intended to reduce bias and variance. This constitutes a sequential generation of learners-all except for the learner most recently developed are for the purpose of developing the next one.

Essentially, making weak learners strong would be a simplified description. The AdaBoost algorithm is based on the principle of boosting with a slight difference, and this difference is explained here. First, we learn about boosting-how it works? During training, it creates n decision trees for the data-and so, once there is one decision tree/model created, it is going to favor misclassified records from the first model.

These records have to be carried to the next model as inputs and so on until you specify the number of base learners to create. Note that repeats of records are allowed in boosting techniques.

In the/model 1, boosting considers errors from subsequent models and adds them as input to model 2. This loop goes on till the aforementioned condition is satisfied. As in the previous model, n models will be built from the errors of the previous models.

And that's how boosting works. Models 1, 2, 3...N are considered individual models which can be classified as decision trees. All boosting models remain to some degree under this one principle. With this boost concept being understood, understanding AdaBoost would not be that hard, and so let us get into what it was designed for.

When random forests were chosen by the algorithm, n number of trees form a full tree, which alternates starting nodes, then multiple leafs. Some trees could be huge, while others could be dwarfed by its fellow trees, and there's no saying how deep a tree can actually go in a random forest. But however, nowadays, an algorithm in AdaBoost builds only one node with two leaves called stump.

This diagram shows the stump. It can be understood clearly that it has only one node with two leaves. Weak learners, indeed, are stumps and boosting techniques rather like these. The sequence of stumps becomes very important in AdaBoost. The error of the first stump would decide how the other stumps would be created. Let's try to understand it with some illustration. Here is a sample dataset which has only three features where the output is in categorical form. Thus, the images display how the dataset looks in reality. Since the output is in binary/categorical form, it will be a classification problem. The dataset can have any number of records in it and any number of features. For explanation purposes, let us take 5 datasets. The output is categorical such as Yes or No. All these records would be assigned some weight.

5) KNN Classifier

K-Nearest Neighbor is one of the very simple Machine Learning algorithms that come under the category of Supervised Learning.

According to the K-NN algorithm, there is a new case/data which will be assumed as similar to its available cases, thus it would be assigned to that category most related to the available categories.

K-NN retains the available information and classifies a new point based on the similarity. Then, it is used to accurately classify the new data into the most appropriate category by the help of its algorithm.

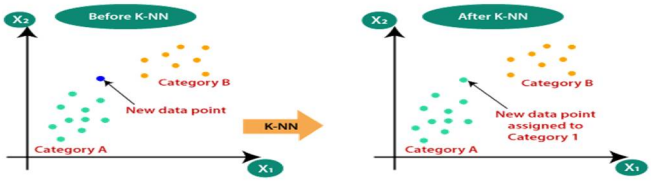
K-NN is used in Regression as well as in Classifications, although it is most popular under the Classification scenarios.

Accordingly, it is called as a non-parametric algorithm, as no assumption is made regarding the data for the underlying assumptions.

Hence, it is named as lazy learner algorithm as this does not keep an instant learning from the training dataset yet it stores the dataset, and when the classification has to be done, uses that dataset.

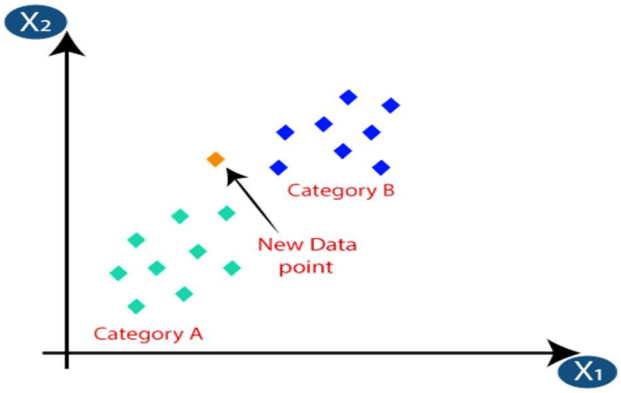
During training, the KNN algorithm stores the dataset and at a later time carries out assignment for an input to that dataset when a new sample needs a classification. For instance, while we have an image of an animal which looks similar to a cat and a dog; we want to make it clear whether it's a cat or a dog. So this identification with KNN algorithm would work since it actually takes some measure of similarity: It would slot features from the new dataset similar to features of both images, i.e. cats or dogs, and bring it to one class or the other based on the most similar features.

Well, we need K-NN algorithms because a data point like x1 must be classified into a particular category, say Categories A and B. K-NN is the algorithm that will help in that; it can classify a specific dataset easily into a particular class of values. Some reference diagram appears in the present:



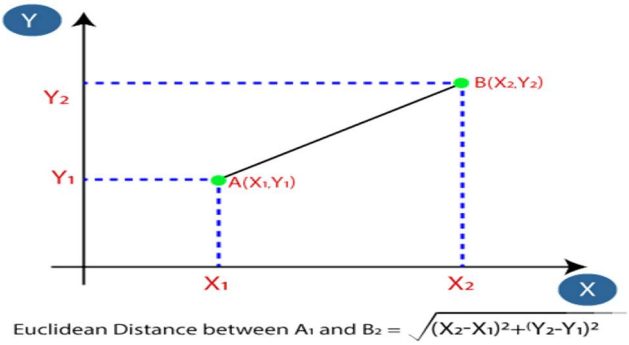
How does K-NN work?
The working of K-NN can be described in terms of the following algorithm.
Focal Steps:

- Step 1: The number K of the neighbor is selected.
- Step 2: The Euclidean distance of K numbers of neighbors is computed.
- Step 3: All K nearest neighbors will be according to the calculated Euclidean distance.
- Step 4: Among these k neighbors, count the number of data points in each category.
- Step 5: Assign new data points to that category for which the number of the neighbor is maximum.
- Step 6: Model as ready.

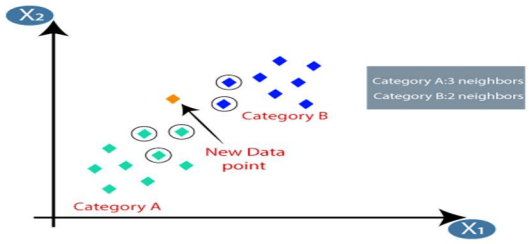


Now to the assignment of the new data point in the relevant category, look into the image below:

- Firstly, we have to choose a number of neighbors and here k=5.
- Now the Euclidean distances between the data points are calculated. Euclidean distance is the distance between two points distance as studied in Geometry as below:



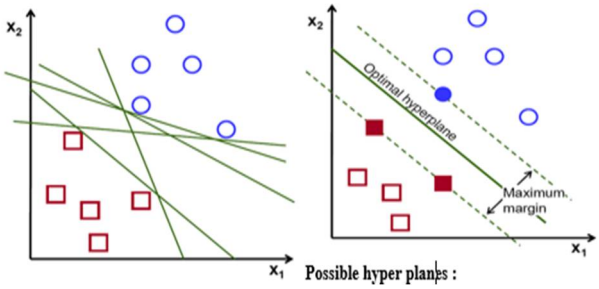
- These three nearest neighbors in category A and two nearest neighbors in category B were derived from the results of calculating the Euclidean distances.



- It shows that three nearest neighbors are of category A and therefore the new data point should belong to category A.
- How to find K in K-NN algorithm?
- Here are some tips to keep in mind while selecting value K in K-NN algorithm:
- There is no fixed way to know which is the best for 'K' so will have to run on some values and find the best suited value among them. Most preferred value of K is 5.
- Very small values of K such as K=1 or K=2, can be badly noisy and their effects are amplified by being outlier points in our model.
- Maximum size of K would in fact be good, but that might lead to some difficulties in practice.
- Merits of KNN Algorithm:
- Very simple to implement.
- Robust against noisy training data.
- It may perform even better with large training datasets.
- Demerits of KNN Algorithm:
- There is always a need to fix K, whose value is sometimes very complex.
- It shares very high computation cost because distance is to be calculated between all the data points to all training samples.

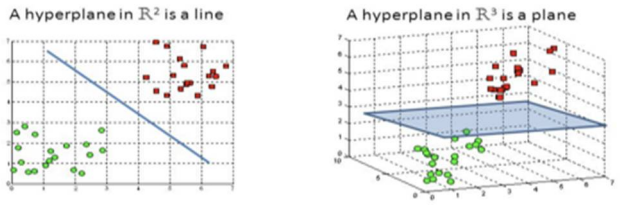
6) SVM

The SVM algorithm has a straightforward objective separates the data points well. Possible Hyperplanes:\



In other words, recognize a hyperplane with a maximum possible margin, or the farthest distance between points of both classes. Data points on the margin give some reason to be classed more confidently in the future.

Hyperplane and Support Vector

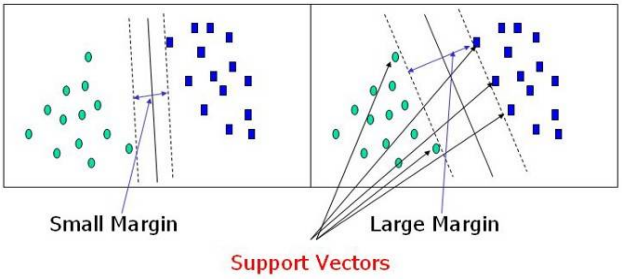


Hyper planes in 2D and 3D feature space

Hyperplanes are actually the decision boundaries that classify data points. One can have data points on either side of hyperplane belonging to different classes. The dimension of hyperplane will also depend on number of features you have. In two-dimensional feature input space, hyperplanes are lines. In an input space of three dimensions, hyperplanes can be called two-dimensional planes. After that it is hard to picture it.

Support Vectors

Support vectors refer to those data points that fall close to hyperplane and therefore, the position and orientation of hyperplane is dependent on these data points. Maximizing margin of classifier is mainly concerned with these support vectors. If all such support vectors are removed, the hyperplane will no longer be defined. In other words, these are those points which do contribute to making your SVM.



Large Margin Intuition

In logistic regression, we take the output from the linear function and squash it in the range of [0,1] with the sigmoidal activation function. When the squashed value is greater than some threshold value (which we take to be 0.5), we assign it a label 1; otherwise, we assign it a label 0. Now SVMs consider the output from the linear function greater than 1 to be one class and less than -1 to be the other class. Therefore, the threshold values changed for

Now the cost function looks like this with the regularization parameter included.

SVM Loss Function

Now, a partial derivative is taken for the loss function concerning the weights to calculate the gradient, and those gradients now update the weights.

7) Multilayer Perceptron Classifier

These are wonderful tools in deep learning, but TensorFlow, Keras, Microsoft Cognitive Toolkit (CNTK), and PyTorch top the charts with all. Most of us would not know that even more popular libraries such as scikit-learn can do a little bit of deep learning modeling.

Following are some salient features of Scikit-learn Multilayer Perceptron (MLP):

• Output layer has no activation function.
• Since the loss for regression cases is square error, the cross-entropy is used for classification cases.
• It has this ability to regression single or multiple target variables.
• The important difference between Scikit and many other popular packages such as Keras is that Scikit does not implement GPU for MLP.

We cannot tune parameters like different activation functions, weight initializers, and so on for each layer. The multilayer perceptron (MLP) is defined as a class of feedforward artificial neural networks (ANNs). The term MLP is ambiguous and sometimes loosely applies to all feedforward ANNs, while sometimes it strictly refers to networks of more than two layers of perceptrons (with threshold activation); see §Terminology. Informally, multilayer perceptrons are referred to as vanilla neural networks, especially with regard to those with just one hidden layer.

An MLP is essentially a network built with at least three layers of nodes - an input layer, a hidden layer, and an output layer. Neurons with nonlinear activation functions constitute all nodes, except input nodes. MLP is trained using backpropagation, which is a supervised learning methodMLP distinguishes itself from a linear perceptron by having multiple layers and non-linear activation. It can discriminate data that cannot be differentiated linearly.

Multilayer perceptron does not refer to a perceptron with many layers, but to many perceptrons organized in layers. It can be referred to as "multilayer perceptron network." MLP perceptrons also, in fact, are not perceptrons in the narrowest sense. The true perceptrons are formally a special case of artificial neurons that employ threshold activation functions such as the Heaviside step function. MLP perceptrons can make use of arbitrary activation functions. A real perceptron makes use of binary classification, while an MLP neuron can classify or regress arbitrarily, depending on the applied activation function.

Later, without respect to nature of the nodes/layers-the arbitrary defined artificial neurons could comprise but not be limited to perceptrons specifically-the term multilayer perceptron was applied. Such interpretation would not lead to a loosening of the "perceptron" definition to that of an artificial neuron in general.

The best MLP model serves in research as it can solve most problems stochastically, which permits an approximate solution to a very complicated problem, e.g., fitness approximation.

According to Cybenko's theorem, MLPs can make universal function approximations

V. RESULTS

The research study has primarily evaluated the performance in terms of working with different algorithms in machine learning concerning the prediction of seismic events such as earthquakes. After implementation and tuning hyperparameters of each model, they were evaluated and compared using different metrics. Thus, the accuracy comparison amongst the tested models has revealed that Random Forest proved to be the best classifier, followed by SVM and MLP classifiers. Random Forest algorithm emerged as the most reliable with prediction-based accuracy lying close to 89% for major seismological activities. The other fairly accurate classifiers were in the range of 83-85% with the exception of Naïve Bayes and KNN while Logistic Regression and AdaBoost were margin-wise in the 80-82 band. In addition, the Classification and Regression Trees (CART) performed acceptably, but in truth had very little opportunity of competing with the beasts. In this regard, these results substantiate the claim towards the machine learning model, and it is thus reasonable to justify its tuning for utmost efficiency in predicting earthquake events. Furthermore, the results show that ensemble techniques such as Random Forest outperform other algorithms in predicting seismic activity when trained using historical earthquake data.

VI. CONCLUSION

Different aspects of application machine learning techniques study for predicting future earthquakes are related to the study and application of different machine learning techniques in predicting the future earthquakes' occurrence. The specific aspect here refers to the capability to differentiate real or positive occurrences of earthquakes from fictive or negative occurrences. Algorithms implemented include Random Forest, Naïve Bayes, Logistic Regression, Multi-Layer Perceptron, AdaBoost, K-Nearest Neighbors, Support Vector Classifier, and Classification and Regression Tree. All of these algorithms have braced well to reasonable performances on some real earthquake datasets. This is given in terms of performance assessment on accuracy and other evaluation metrics by appropriate choice of hyperparameters for each one's models. Performance turned out good as some models averaged an accuracy of 0.89 for significant earthquake events. Thus, in addition, this spices it up to the usage of machine learning techniques into the problem of earthquake prediction, again with a solid feature selection and tuning of hyperparameters toward better prediction accuracy. Though further advances in the usage of real-time input data would still be beneficial for making predictions more credible. Future works may therefore focus on seeking the generalizations of the models towards the complex conceptual designs of geological and environmental determinants for their accurate earthquake predictions and effectiveness in time.

VII. FUTURE SCOPE

There is a very bright future ahead for machine learning techniques concerning earthquake prediction, and they promise to bring more accuracy and relevance into those predictions. With almost continuous improvement in the state-of-the-art high-quality real-time data, possibilities for emerging prediction outcomes are being created. Research that may be of importance in the near future would holistically integrate advanced deep learning architectures, which include Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), as they can yield finer information on patterns of seismic events. The inclusion of satellite imagery, geological surveys, and environmental data to explore the seismic space would mean stronger prediction systems. New data can lead to advances in prediction schemes using reinforcement learning. Real-time data acquisition and localized earthquake alarms will also minimize losses in terms of severity and number of affected individuals at a scaled-down level. In long-term perspectives, automation should help the whole world to monitor seismic activities and produce predictive results that could otherwise not be achieved within traditional systems.

REFERENCES

[1] S. B. Feroz, M. S. Sevas, N. Sharmin, and A. Chatterjee, Performance Analysis of Machine Learning and Deep Learning Architecture for Earthquake Magnitude Prediction,Proceedings of 2023 IEEE 9th International Women in Engineering (WIE) Conference on Electrical and Computer Engineering, WIECON-ECE 2023, pp. 70–75, 2023, doi: 10.1109/WIECON-ECE60392.2023.10456401.

[2] F. Ahmed, S. Akter, S. M. M. Rahman, J. Bin Harez, A. Mubasira, and R. Khan, "Earthquake Magnitude Prediction Using Machine Learning Techniques," 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation, IATMSI 2024, 2024, doi: 10.1109/IATMSI60426.2024.10502770.

[3] R. Mallouhy, C. A. Jaoude, C. Guyeux, and A. Makhoul, "Major earthquake event prediction using various machine learning algorithms," Management, ICT-DM 2019, Dec. 2019, doi: 10.1109/ICT-DM47966.2019.9032983.

[4] A. Gaba, A. Jana, R. Subramaniam, Y. Agrawal, and M. Meleet, "Analysis and Prediction of Earthquake Impact-a Machine Learning approach Technology for Sustainable Solution, Proceedings, Dec. 2019, doi: 10.1109/CSITSS47250.2019.9031026.

[5] B. Arunadevi, M. Madiajagan, M. I. Hussain, R. Lakshmi, M. M. Ramakrishna, and K. Sengupta Das, "Risk Prediction of Earthquakes using Machine Learning," 3rd International Conference on Electronics and Sustainable Communication Systems, ICESC 2022 - Proceedings, pp. 1589–1593, 2022, doi: 10.1109/ICESC54411.2022.9885674.

[6] P. Chomchit and P. Champrasert, "Strong-motion Earthquake Prediction Model using Convolutional Extreme Learning Machine," ITC-CSCC 2022 - 37th International Technical Conference on Circuits/Systems, Computers and Communications, pp. 756–759, 2022, doi: 10.1109/ITC-CSCC55581.2022.9894973.

[7] R. Kumar, M. Gupta, K. S. Bedi, A. Upadhyay, and A. J. Obaid, "Earthquake Prediction Using Machine Learning," Proceedings - IEEE 2023 5th International Conference on Advances in Computing, Communication Control and Networking, ICAC3N 2023, pp. 195–198, 2023, doi: 10.1109/ICAC3N60023.2023.10541482.

[8] S. Dhotre, K. Doshi, S. Satish, and K. Wagaskar, "Exploring Quantum Machine Learning (QML) for Earthquake Prediction," 2022 2nd International Conference on Intelligent Technologies, CONIT 2022, 2022, doi: 10.1109/CONIT55038.2022.9848250.

[9] S. Rath, A. B. Gurulakshmi, S. Aditya, V. R. Naik, A. Pawar, and V. Karoor, "A Comparison of Earthquake Prediction using Machine Learning Algorithms," Proceedings of 5th International Conference on IoT Based Control Networks and Intelligent Systems, ICICNIS 2024, pp. 1529–1533, 2024, doi: 10.1109/ICICNIS64247.2024.10823378

[10] M. Elbes, S. Alzu'bi, and T. Kanan, "Deep Learning-Based Earthquake Prediction Technique Using Seismic Data," 2023 International Conference on Multimedia Computing, Networking and Applications, MCNA 2023, pp. 103–108, 2023, doi: 10.1109/MCNA59361.2023.10185869.

[11] G. S. Manral and A. Chaudhary, "Prediction of Earthquake Using Machine Learning Algorithms," doi: 10.1109/ICIEM59379.2023.10166658.

[12] P. Shrote, P. Dasarwar, S. Dongre, and P. Khobragade, "Earthquake Prediction Through Machine Learning Approach," Proceedings - 4th International Conference on Technological Advancements in Computational Sciences, ICTACS 2024, pp. 542–546, 2024, doi: 10.1109/ICTACS62700.2024.10840652.

[13] R. Kail, E. Burnaev, and A. Zaytsev, "Recurrent Convolutional Neural Networks Help to Predict Location of Earthquakes," IEEE Geoscience and Remote Sensing Letters, vol. 19, 2022, doi: 10.1109/LGRS.2021.3107998.

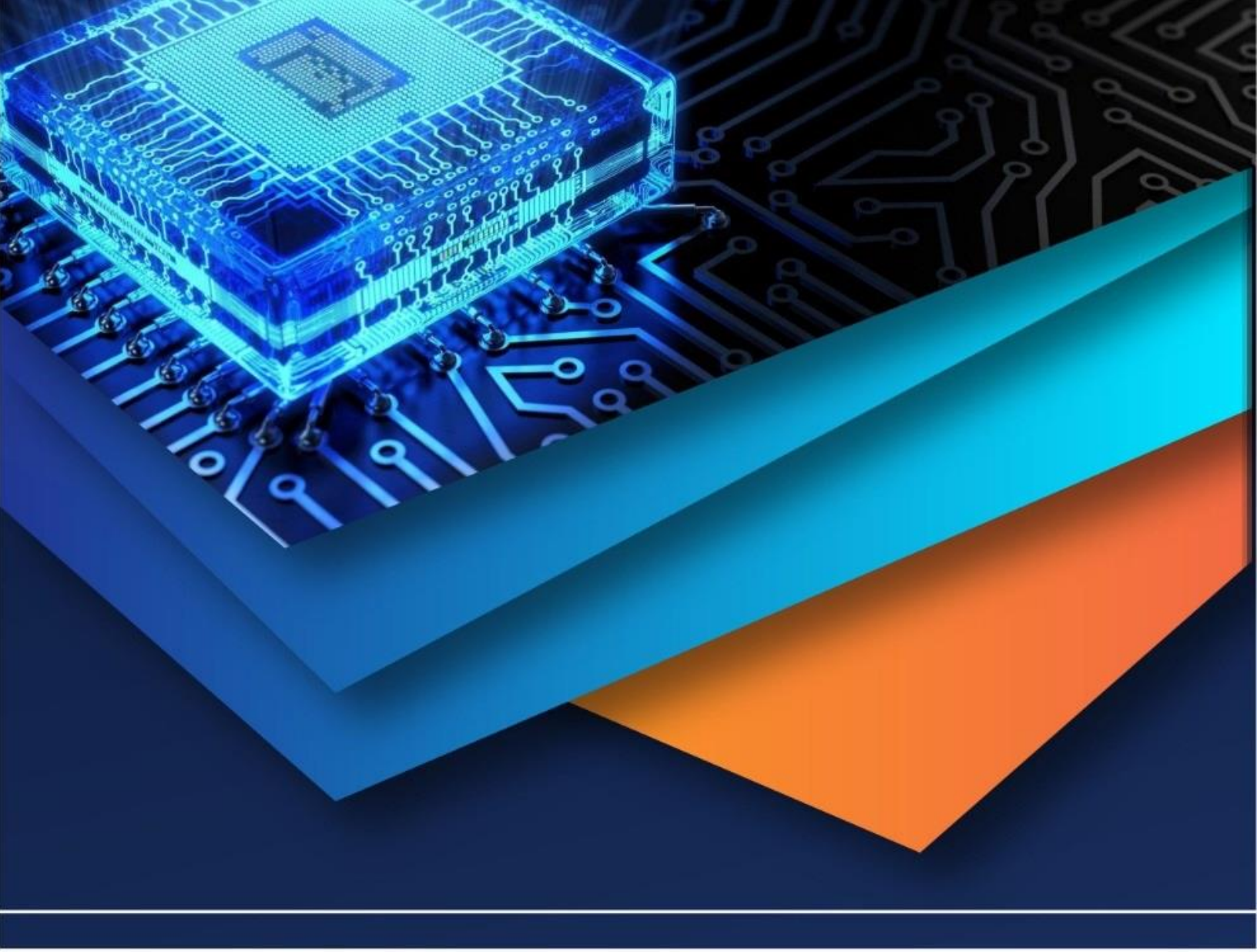
[14] M. H. Al Banna et al., "Application of Artificial Intelligence in Predicting Earthquakes: State-of-the-Art and Future Challenges," doi: 10.1109/ACCESS.2020.3029859.

[15] Y. Zhao, S. Lv, and P. Liu, "Advances in earthquake prevention and reduction based on machine learning: A scoping review (July 2024)," doi: 10.1109/ACCESS.2024.3467149.

[16] Q. Wang, Y. Guo, L. Yu, and P. Li, "Earthquake Prediction Based on Spatio-Temporal Data Mining: An LSTM Network Approach," 10.1109/TETC.2017.2699169

[17] C. Yang, K. Zhang, G. Chen, Y. Pan, L. Zhang, and L. Qu, "Application of Machine Learning to Determine Earthquake Hypocenter Location in Earthquake Early Warning," IEEE Geoscience and Remote Sensing Letters, vol. 21, pp. 1–5, 2024, doi: 10.1109/LGRS.2023.3348107

.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY