MAUSAM, **72**, 4 (October 2021), 781-790

551.509.314 : 551.577 : 004.42

A precipitation forecasting model using machine learning on big data in clouds environment

MAHBOOB ALAM and MOHD. AMJAD

Department of Computer Engineering

Jamia Millia Islamia University, New Delhi – 110 025, India

(*Received 17 July 2020, Accepted 23 August 2021*)

e mail : mahboobru@gmail.com

सार – न्यूमेरिकल वेदर प्रेडिक्शन (NWP) लंबे समय से मौसम विज्ञानियों के लिए एक मुश्किल काम रहा है। वायुमंडलीय गतिकी मॉडल के लिए अत्यंत जटिल है, और अराजकता सिद्धांत हमें सिखाता है कि मौसम की भविष्यवाणी करने के लिए उपयोग किए जाने वाले गणितीय समीकरण प्रारंभिक स्थितियों के प्रति संवेदनशील होते हैं; यानी, थोड़ी परेशान प्रारंभिक स्थितियां बहुत अलग पूर्वानुमान दे सकती हैं। इन वर्षों में, मौसम विज्ञानियों ने वायुमंडलीय गतिकी के लिए कई अलग-अलग गणितीय मॉडल विकसित किए हैं, जिनमें से प्रत्येक ने थोड़ा अलग अनुमान और सरलीकरण किया है, और इसलिए प्रत्येक अलग-अलग पूर्वानुमान दे रहा है। यह नोट किया गया है कि प्रत्येक मॉडल में अलग-अलग स्थितियों में पूर्वानुमान लगाने की अपनी ताकत और कमजोरियां होती हैं, और इसलिए प्रदर्शन में सुधार करने के लिए, वैज्ञानिक अब विभिन्न मॉडलों से मिलकर एक पहनावा पूर्वानुमान का उपयोग करते हैं और उन मॉडलों को विभिन्न प्रारंभिक स्थितियों के साथ चलाते हैं। यह पहनावा पद्धति सांख्यिकीय पोस्ट-प्रोसेसिंग का उपयोग करती है; आमतौर पर रैखिक प्रतिगमन। हाल ही में, मशीन लर्निंग तकनीकों को NWP पर लागू किया जाने लगा है। तंत्रिका नेटवर्क, लॉजिस्टिक रिग्रेशन और जेनेटिक एल्गोरिदम के अध्ययन ने वर्षा की भविष्यवाणी के लिए मानक रैखिक प्रतिगमन में सुधार दिखाया है। गैगने एट अल ने वर्षा पूर्वानुमान में सुधार के लिए कई मशीन लर्निंग तकनीकों का उपयोग करने का प्रस्ताव रखा। उन्होंने ब्रिमन की यादृच्छिक वन तकनीक का इस्तेमाल किया, जिसे पहले मौसम विज्ञान के अन्य क्षेत्रों में लागू किया गया था। नेक्स्ट जेनरेशन वेदर रडार (NEXRAD) डेटा का उपयोग करके प्रदर्शन का सत्यापन किया गया। एक समेकित पूर्वानुमान का उपयोग करने के बजाय, यह वर्षा के पूर्वानुमान को बेहतर बनाने के लिए मशीन लर्निंग से संबंधित तकनीकों के उपयोग पर चर्चा करता है।

यह पेपर वर्षा डेटा के मानचित्रण के लिए एक दृष्टिकोण प्रस्तुत करना है। यह परियोजना एक मशीन लर्निंग पद्धति पर पहुंचने का प्रयास करती है जो इष्टतम है और वर्षा के स्तर की भविष्यवाणी करने के लिए डेटा संचालित है जो किसानों को कृषि क्षेत्र को लाभ प्रदान करने का लक्ष्य रखता है।

ABSTRACT. Numerical weather prediction (NWP) has long been a difficult task for meteorologists. Atmospheric dynamics is extremely complicated to model, and chaos theory teaches us that the mathematical equations used to predict the weather are sensitive to initial conditions; that is, slightly perturbed initial conditions could yield very different forecasts. Over the years, meteorologists have developed a number of different mathematical models for atmospheric dynamics, each making slightly different assumptions and simplifications, and hence each yielding different forecasts. It has been noted that each model has its strengths and weaknesses forecasting in different situations, and hence to improve performance, scientists now use an ensemble forecast consisting of different models and running those models with different initial conditions. This ensemble method uses statistical post-processing; usually linear regression. Recently, machine learning techniques have started to be applied to NWP. Studies of neural networks, logistic regression, and genetic algorithms have shown improvements over standard linear regression for precipitation prediction. Gagne et al proposed using multiple machine learning techniques to improve precipitation forecasting. They used Breiman's random forest technique, which had previously been applied to other areas of meteorology. Performance was verified using Next Generation Weather Radar (NEXRAD) data. Instead of using an ensemble forecast, it discusses the usage of techniques pertaining to machine learning to improve the precipitation forecast.

This paper is to present an approach for mapping of precipitation data. The project attempts to arrive at a machine learning method which is optimal and data driven for predicting precipitation levels that aids farmers thereby aiming to provide benefits to the agricultural domain.

Key words – Rainfall, Pollution, Datasets, Analysis, Big data.

1. Introduction

Precise rainfall forecasts has always been one of the most significant concerns in hydrological research since prior warning of extreme weather conditions can help to avoid accidents and damage from natural disasters with beforehand and accurate forecasts. Creating a forecasting system for precise rainfall forecasting is among the biggest challenges in front of researchers in various fields like weather data mining, environmental machine learning, operational hydrology and statistical forecasting. A major and commonly occurring challenge in these problems is to figure out the way to predict the future trends by studying the trends of past (Baldauf *et al*., 2020).

The parameters needed to forecast rainfall are very convoluted and intricate for short-term periods. Physical processes involved in rainfall usually comprises of several sub-processes. At times, using the same global model for accurately modeling the rainfall isn't possible. The concept of modular modeling tackles this problem and combining of several separate models has recently gained much attention in rain forecasting.

Firstly the various sub-processes are defined in the modular approach, and thereafter the separate models (which are also known as the local or expert models) are set up for each of them. Quite a few modular models have been proposed till date, which depend on hard or soft segmentation of training data. Soft segmentation is nothing but the overlapping of datasets the weighted average of each local model is utilized to find the overall forecast output (Bi *et al*., 2015; Cano *et al*., 2003).

Numerical weather prediction (NWP) has long been a difficult task for meteorologists. Atmospheric dynamics is extremely complicated to model, and chaos theory teaches us that the mathematical equations used to predict the weather are sensitive to initial conditions; that is, slightly perturbed initial conditions could yield very different forecasts.

Over the years, meteorologists have developed a number of different mathematical models for atmospheric dynamics, each making slightly different assumptions and simplifications, and hence each yielding different forecasts. It has been noted that each model has its strengths and weaknesses forecasting in different situations, and hence to improve performance, scientists now use an ensemble forecast consisting of different models and running those models with different initial conditions. This ensemble method uses statistical post-processing; usually linear regression. Recently, machine learning techniques have started to be applied to NWP (Mandale *et al*., 2015; Alam *et al*., 2019).

2. Literature review

2.1. *Scope of work*

Farmers, across the nation, are majorly reliant on timely precipitation for harvest and associated profits. Uncertainty clouding this phenomenon, due to the slow onset of vices like the climatic changes and global temperature surge lead to the collaboration of traditional farming methodologies and predictions of precipitation. India being an agrarian country with around 70% of its people depending upon agriculture, accounted for about 12 % of suicides by its farmers. Numerous conflicting reasons have been accorded but primarily targeting the failure in monsoon predictions. This adopted the concept of integrating the machine learning in assisting them for cultivation of their harvest and securing their lives (Bhatt *et al*., 2015; Katal *et al*., 2013).

2.2. *Possible Machine learning models*

2.2.1. *Naive Bayes Clustering*

In ML, the naive bays classifier is a group of probability based classifiers that work by putting to use the Baye's theorem with robust (naive) assumptions of lack of interdependence among features. Knife bays have been researched upon thoroughly since the 1950s. It was introduced in the early 1960s with different name in the text retrieving fraternity, and is still a widely used (foundational) technique for textual classification, the issue of identifying documents that belong to the same category or another (say, promotional, career related or politics) with frequency of words used as features. By using correct pre-processing, this is a promising method in this field with more sophisticated ways including support vector machines. Besides, this has applications in automated medical diagnosis too (Abaker *et al*., 2015; Das *et al*., 2007).

Naive Bayes is an elementary method to create classifiers: The models assigning classes labels for problem instances are depicted as feature value vectors, where a finite set is used to draw the class labels. This is not a standalone algorithm to train such a classifier, rather it is a group of algorithms working on one general principle : The value of one specific feature is assumed by the naïve classifier to exceed that of another feature, provided the class variable is free (Garcia *et al*., 2012; Alam *et al*., 2017).

What Naive Bayes classifiers do is that, every single of these characteristics is considered to be contributing independently towards the possibility, the given fruit is a pineapple, irrespective of any probable similarities among

features like shade, gradient, degree of sphericity and radius features. However with certain selected kinds of probability models, Naive Baye's classifiers also offer the capabilities to be trained effectively in a learning environment which is under supervision. In some reallife applications, estimation of parameters for the Naive Bayes model employs the technique of highest probability; *i.e.*, one could use the naïve Bayes models without having to accept the Bayesian probability or making use of any other Bayesian methods.

In short, the inexperienced Naive is a probabilistic model which is conditional: the problem example is classified, a vector is then used to represent it, which denotes say "m" features (variables that aren't dependent on each other) (Kittler *et al*., 1998; F. G. *et al*., 2010). It specifies the probabilities.

$$p(C_k|x_1,...,x_n)$$

For every possible outcome of K or Ck squares

The challenge that the above mentioned formula poses is that when the "N" which is the feature count, is large or if an attribute can take multiple values then it is not possible to base such models on probability tables. Hence we make improvements to the model so that it becomes more streamlined. Using the Bayes theorem, we can decompose the conditional probability.

$$p(C_k|x)=\frac{p(C_k)p(x|C_k)}{p(x)}$$

2.2.2. *Support vector machine*

In ML, vector machines support (SVMs supports vectored networks too) monitoring learning models with connected learning algorithms that use data frost regression and classification analysis.

With respect to one single or a couple of classes the SVM training algorithm generates such models that provide novel test cases for 1 class or another, which makes it binary linear classifier that isn't a probabilistic one. However methods such as plot scaling use SVM exits to do probabilistic classification. An SVM model is nothing but a collection of instances like points present in space, mapped such that instances of individual classes are segmented by an explicit difference which is as detailed as possible. The novel instances are then mapped to the same place and it is estimated which category they belong to. Besides demonstrating linear classifications, SVMs can effectively use non-linear classifiers, called kernel moves, to map their inputs in multi-dimensional feature locations (Coulouris *et al*., 2011; Dean *et al*., 2010).

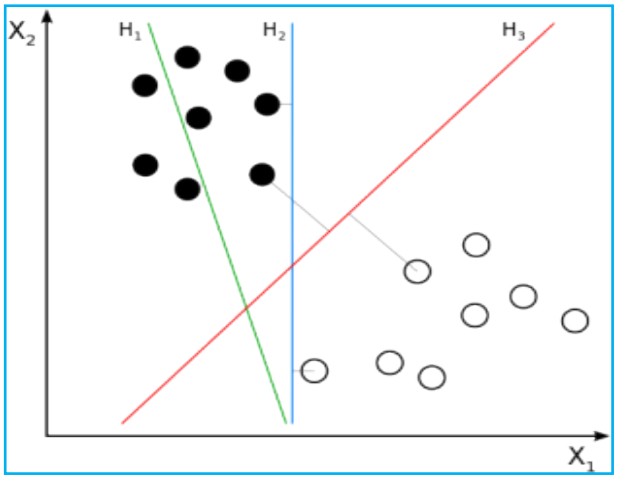

Fig. 1. Support vector machine graph plotting

Whenever data is not labeled, supervised learning ceases to be a possibility, and the requirement of an untrained learning method arises, which attempts to detect the natural clustering in datum along with the new data which these newly created clusters maps to. The Support Vector Clustering Algorithm, devised by Vladimir Vapnik and Hava Siegelmann, implements support vectors stats, generated by Supported Vector Machines algorithm, for categorizing unlisted datum along with being the most extensively utilised in the industrial uses. Is one of the cluster algorithms to be performed?

Classifying a machine is a usual job in ML. Suppose that a data point in consideration falls under 1 of every 2 classified categories, while our objective is to find out the class to which a new data point will belong to. Whereas in SVMs, a data point is considered as a vector with m-dimensions (array of m numerals), also what needs to be known is if those points can be separated in an m^{-1} dimension hyper plane. It is known as Linear Classifier. A number of hyper planes are there for data classification. The most preferred hyper plane is the one which depicts maximum margins or separations among the 2 classified groups. Therefore we attempt to pick the hyper plane in a manner to maximize the distance from the most proximate point. This sort of a hyper plane is called the maximum-margin hyper plane. Also, the linear classifier defining this plane is referred to as the maximum classifier; equivalent, indicating optimal stability (Boyd *et al*., 2011; Mahmoud *et al*., 2020).

We can see in Fig. 1 that H1 doesn't separate class; however it is done by H2, but just with a little origin. H3 separates them with the most margins.

In a more formal sense, an SVM generates a high-plane or collection of hyper planes in a very high or

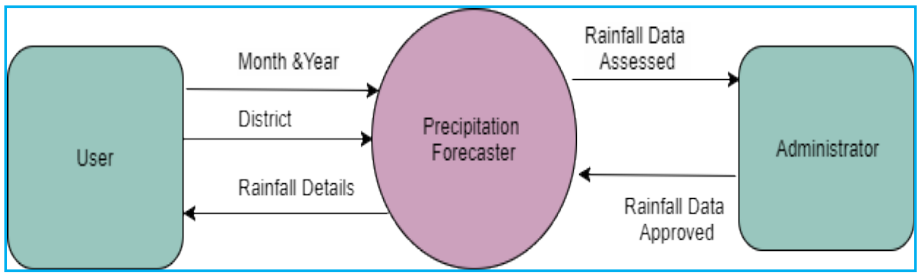

Fig. 2. DFD level-0

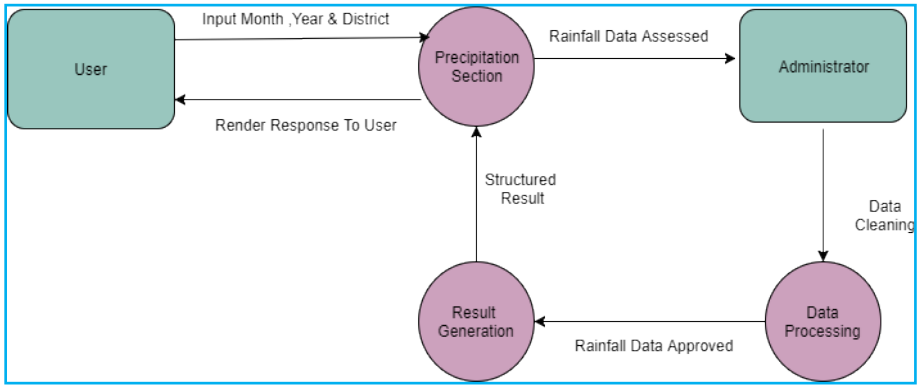

Fig. 3. Data flow diagram level 1 diagram

infinite-dimensional space that can be used to do other tasks like classification, regression, or outlier detection. Obviously, an honest distinction is obtained by the hyper plane having the biggest distance to the closest training-data point of any category (also known the functional margin), because the normally large margin is that the normalization error for that class.

The essential problem is defined in finite dimensional space, what usually happens is that sets to distinguish aren't different in a linear sense there in space. As a consequence of this, it had been suggested that the initial finite-dimensional space should be mapped to an awfully high-dimensioned space, possibly leading to the distance in this space easier to stay the computational load reasonable, the mapping utilized by Support Vector Machine criterions meant to make sure that the scalar-product are calculated with ease in terms of variables within the original space. Allowing the kernel function k (x, y) to be resolved the matter by defining as. Hyper planes in high-dimensional space are described as clusters of points whose real number is constant along the vector in space. The vectors that define the hyper plane will be chosen for linear combinations with the parameters αi of images of the feature vectors x_i occurring within the data base. The purpose x within the feature space to be mapped to the hyper plane, is calculated by the relation (Iqbal *et al*., 2020; Galar *et al*., 2012).

One thing to be observed here is that as $k(x, y)$ gets smaller as it moves farther from x,

$$\sum_i \alpha_i k(x_i, x) = \text{constant}$$

Then every term that measures the degree of proximity of the test point x to the corresponding data base point x_i. Thus, the summation of the kernels given above can provide us with the relative inflation of each test point, with the origin of the data points discriminating one or the other. Note that consequently the point set x mapped with any hyperplane can be rather complex as a result, which allows for more complicated differentiation among sets which aren't convex.

3. System methodology

In this paper, we gave the brief description about the system design and methodology of our research work. The chapter includes the complete and detailed system design and the complete system architecture. The system design includes how the flow of data will take place from one activity to the activity. The system architecture will explain the way, the user will interact with the system through the user interface [8][9]. The user will then give its review about the product through the interface and the featured based structured summary of that particular

product will be dynamically updated. The chapter also includes diagrams depicting the flow of data and the use case schematic drawing of our research work. It also includes the various algorithms that we will use in the implementation of the research (Tsai *et al*., 2016, Mayer *et al*., 2014; Kaur *et al*., 2017).

3.1. *Data Flow Diagram*

We have two level flow diagrams Fig. 2 and Fig. 3 (level-0 and level-1) in which we can see how the process will work.

3.2. *Algorithm*

Algorithm

Model Selection Using Genetic Algorithm

1. Generic algorithms are employed to select models and hyper parameters optimization.
2. Multiple copies of dataset will be created and different models are applied as input to the genetic algorithms initial population.
3. With each generation on the basis of fitness function GA will select an optimum machine learning model with optimum hyper parameters.
4. The output of genetic algorithms will be a model from the pool of algorithms like Naive Bayes Classifier, nearest neighbours, DCCA, Transferred Data Clustering, etc.

We can also select the best Parameters for each machine learning model through genetic Algorithms.

3.2.1. *Proposed model*

The layout structure is used to analysis huge datasets and the effects will assist to identify styles in area of evaluation. We can describe layout architecture in four layers as shown in Fig. 4, We are going to describe layers as given below (Englei *et al*., 2020; Fan *et al*., 2012, Lopez *et al*., 2015).

First Layer : The information layer deal with information procurement forms; it gets information from various sources which are dealt with by administrative and open structures.

Second Layer : The cloud layer stores big datasets on the cloud. In this layer all big datasets stores on cloud then work on distributed, parallel processing.

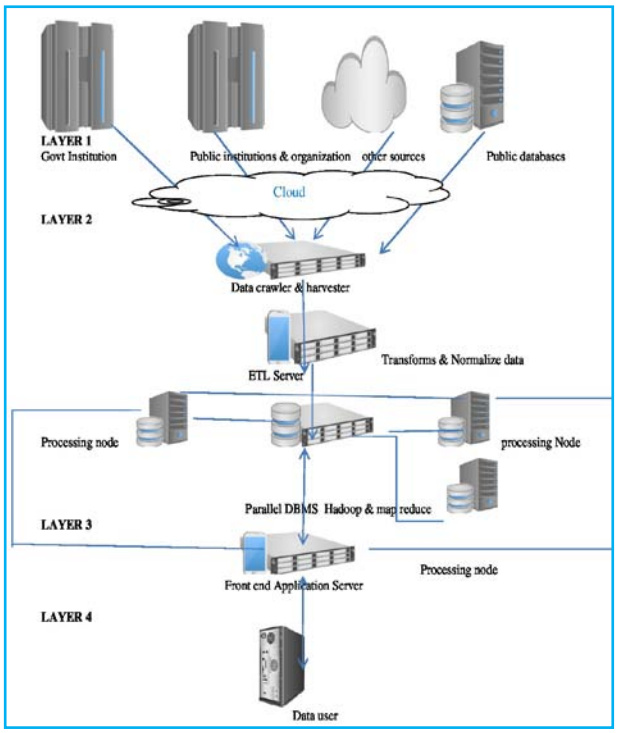

Fig. 4. Proposed architecture

Third Layer : The facts layer elements and use large datasets of monetary and economic statistics, this accretion paintings on dispensed, parallel processing.

Fourth Layer : The consumer layer is the interface for person to get entry to massive datasets and manages the query for analysis and reviews.

The ETL (extract, transform and load) server is a mediator layer placed between the second layer and third layer, get records from the information crawler and harvester detail, adjustments it in a brand new shape. The ETL regularize data, transforms it based totally on a predefined shape and rejects not needed or incompatible information.

The ETL layer inserts facts inside the parallel allotted DBMS that implements the Hadoop and MapReduce framework. The goal of the layer is to regularize statistics and convey it to a not unusual layout, asked through the parallel DBMS.

With the help of ETL system, gets consolidate, altered and loaded into the parallel DBMS, using a facts model optimized for records retrieval and evaluation. The user will process his task *via* a the front cease application server, with a purpose to translate them into queries and publish them to the parallel DBMS for dealing out

(Kollios *et al*., 2015; Fernandez *et al*., 2014; Gottlieb *et al*., 1989).

3.2.2. *Design goal*

The aim of our layout architecture is to do green and precise of weather forecasting. To obtain this, we are going to system the input records, discern out the selected features and thoroughly analysis. Therefore, the following measures are important for the processing performance of our proposed structure.

(*i*) *Analysis accuracy* : This is the main goal of our architecture design.

(*ii*) *Rate of reduction* : In this architecture, the act of characteristic engineering influences the accuracy of analysis directly.

(*iii*) *Efficiency in time* : carried out in climate forecasting, the architecture must evaluation rapid.

4. Results and analysis

The performance of PCA, SVM, Random forest, Gaussian Naïve Bayes, ANN/MLPC classifier has been evaluated on the basis of precision score as shown in Fig. 5. In information retrieval, precision is a measure of result relevancy and hence, is a useful measure for predicting the success when the classes are highly imbalanced.

The following graph depicts the results based on precision score for various models use on different set of months.

4.1. *Result for May-June-July*

We can see the results for the set of month May, June and July as given following.

4.1.1. *Artificial Neural Network (ANN)*

The result of above set of month by using ANN is shown in Fig. 6.

4.1.2. *Naïve Bayes*

The result of above set of month by using naïve bayes is shown in Fig. 7.

4.1.3. *Random forest*

The result of above set of month by using random forest is shown in Fig. 8.

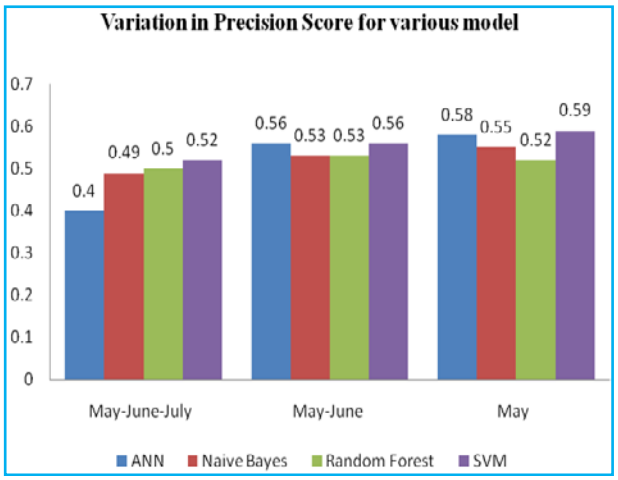

Fig. 5. Precision score for various model

Fig. 6. Result after using ANN for May-June-July month

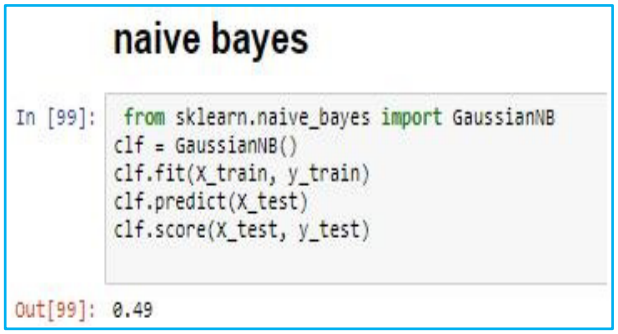

Fig. 7. Result after using Native Bayes for May-June-July month

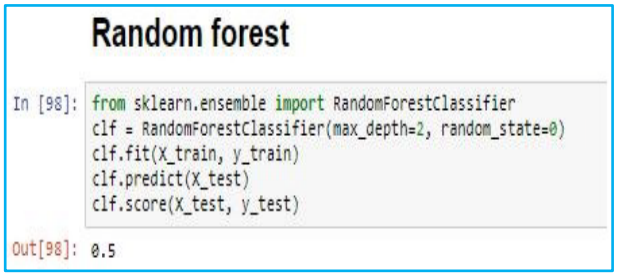

Fig. 8. Result after using Random Forest for May-June-July month

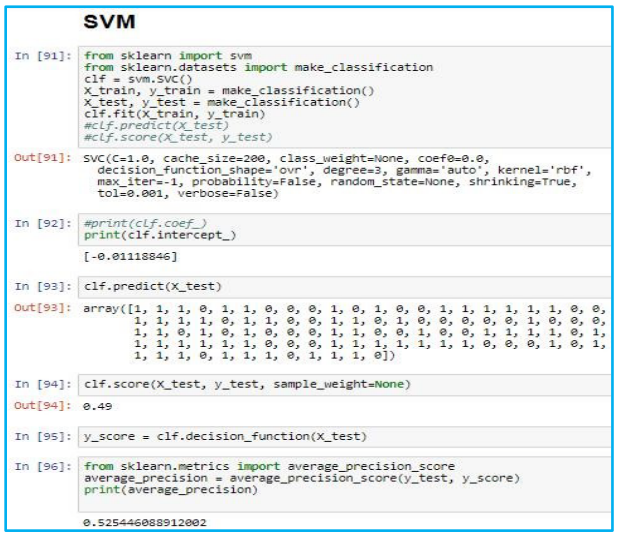

Fig. 9. Result after using SVM for May-June-July month

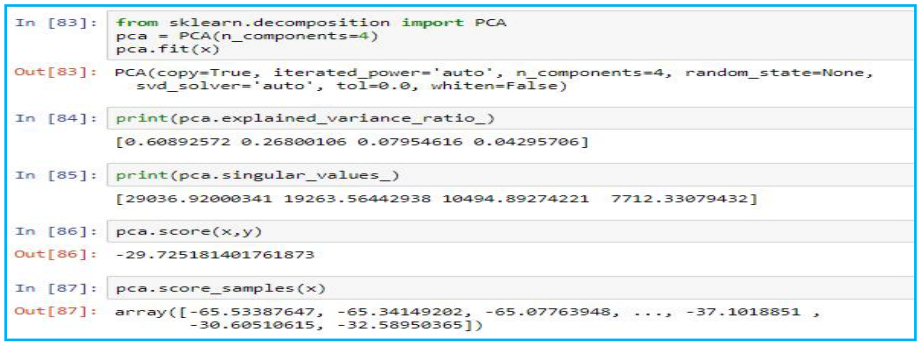

Fig. 10. Result after using PCA for May-June-July month

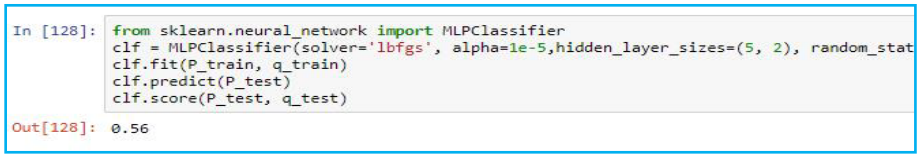

Fig. 11. Result after using ANN for May-June month

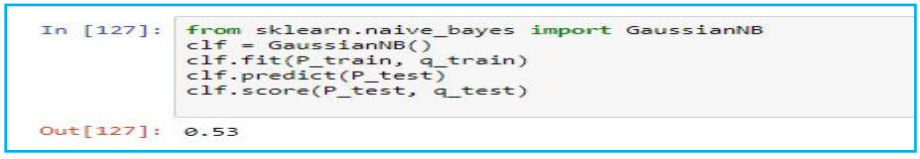

Fig. 12. Result after using Native Bayes for May-June month

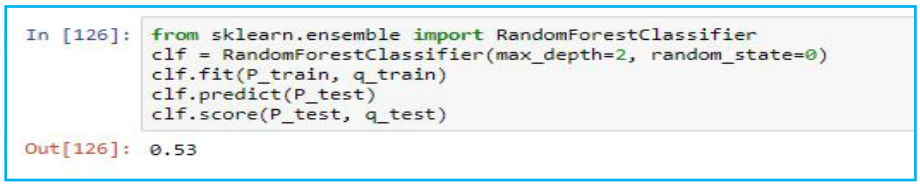

Fig. 13. Result after using Random Forest for May-June month

```
In [122]: from sklearn import svm
          from sklearn.datasets import make_classification
          clf = svm.SVC()
          P_train, q_train = make_classification()
          P_test, q_test = make_classification()
          clf.fit(P_train, q_train)
          clf.predict(P_test)
          clf.score(P_test, q_test)
Out[122]: 0.56

In [123]: q_score = clf.decision_function(P_test)

In [124]: from sklearn.metrics import average_precision_score
          average_precision = average_precision_score(q_test, q_score)
          print(average_precision)

          0.5455558109588025
```

Fig. 14. Result after using SVM for May-June month

```
In [219]: from sklearn.neural_network import MLPClassifier
          clf = MLPClassifier(solver='lbfgs', alpha=1e-5,hidden_layer_sizes=(5, 2), random_state=1)
          clf.fit(R_train, s_train)
          clf.predict(R_test)
          clf.score(R_test, s_test)
Out[219]: 0.58
```

Fig. 15. Result after using ANN for May month

```
In [218]: from sklearn.naive_bayes import GaussianNB
          clf = GaussianNB()
          clf.fit(R_train, s_train)
          clf.predict(R_test)
          clf.score(R_test, s_test)
Out[218]: 0.55
```

Fig. 16. Result after using Native Bayes for May month

```
In [217]: from sklearn.ensemble import RandomForestClassifier
          clf = RandomForestClassifier(max_depth=2, random_state=0)
          clf.fit(R_train, s_train)
          clf.predict(R_test)
          clf.score(R_test, s_test)
Out[217]: 0.52
```

Fig. 17. Result after using Random Forest for May month

4.1.4. *Support Vector Machine (SVM)*

The result of above set of month by using SVM is shown in Fig. 9.

4.1.5. Principal Component Analysis (PCA)

The result of above set of month by using PCA is shown in Fig. 10.

It is quite clear that the PCA model did not perform well and had a negative precision score. So, PCA was not used for further observations.

4.2. *Result for May-June*

We can see the results for the set of month may and june as given following.

4.2.1. *ANN*

The result of above set of month by using ANN is shown in Fig. 11.

```
In [216]: from sklearn import svm
          from sklearn.datasets import make_classification
          clf = svm.SVC()
          R_train, s_train = make_classification()
          R_test, s_test = make_classification()
          clf.fit(R_train, s_train)
          clf.predict(R_test)
          clf.score(R_test, s_test)
Out[216]: 0.59
```

Fig. 18. Result after using SVM for May month

4.2.2. *Naïve Bayes*

The result of above set of month by using native bayes is shown in Fig. 12.

4.2.3. *Random Forest*

The result of above set of month by using random forest is shown in Fig. 13.

4.2.4. *SVM*

The result of above set of month by using SVM is shown in Fig. 14.

4.3. *Result for May*

We can see the results for the month may as given following.

4.3.1. *ANN*

We can see the result for the month may by using ANN is shown in Fig. 15.

4.3.2. *Naïve Bayes*

We can see the result for the month may by using native bayes is shown in Fig. 16.

4.3.3. *Random Forest*

We can see the result for the month may by using random forest is shown in Fig. 17.

4.3.4. *SVM*

We can see the result for the month may by using SVM is shown in Fig. 18.

5. Comparison of the results

We can see in following graphs the comparison of the precision score of different methods for the different

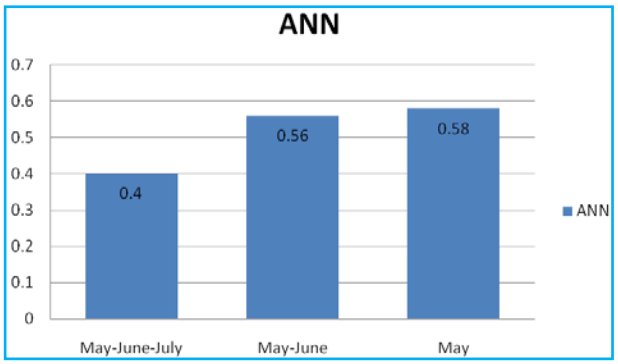

Fig. 19. Precision score for ANN

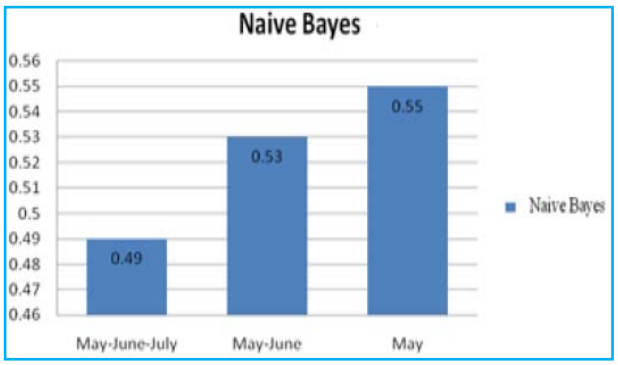

Fig. 20. Precision score for Naïve Bayes

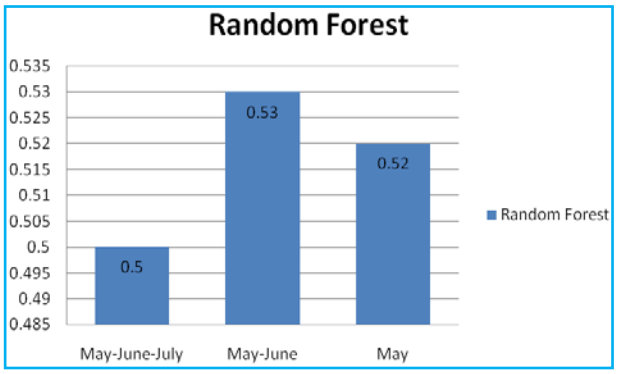

Fig. 21. Precision score for Random Forest

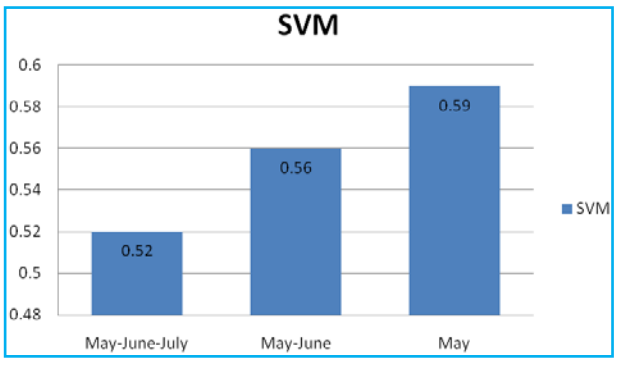

Fig. 22. Precision score for SVM

TABLE 1

Variation Precision score for SVM

Months	ANN	Naive Bayes	Random Forest	SVM
May-June-July	.40	.49	.50	.52
May-June	.56	.53	.53	.56
May	.58	.55	.52	.59

set of months. Fig. 19 shows the result by artificial neural network (ANN), Fig. 20 shows the result by naïve bayes, Fig. 21 shows the result by random forest and Fig. 22 shows the result by support vector machine (SVM).

Finding the model that can best describe the data and can best predict the rainfall is very essential. From the results, we can see according to the precision score as shown in Table 1. SVM (Support Vector Machine) classifier performed best among all the classifiers in the given performance metric-Precision score.

6. Conclusion

Forecasting of rainfall has long been a difficult task. Since, atmospheric dynamics is extremely complicated to model; several models have been used in this research work, each having its own strengths and weaknesses in forecasting for different months. For reasonable evaluation of the performance of these models -PCA, SVM, Random forest, Gaussian Naïve Bayes, ANN/MLPC classifier, they have been separately used for the purpose of comparison. This chapter summarizes the main results of the project.

Future prospects include the following things:

(*i*) Research of the climate change and its impact on health Using approaches of distributed and parallel analytics big data using cloud environment.

(*ii*) The structure optimization of the climate change and its impact on health using approaches of distributed and parallel analytics big data using cloud environment.

Acknowledgements

I want to say thank you to all reviewer of my research paper and my PhD supervisor Dr. Mohd. Amjad.

Disclaimer : The contents and views expressed in this research paper/article are the views of the authors and do not necessarily reflect the views of the organizations they belong to.

References

Abaker, I., Hashem, T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A. and Khan, S. U., 2015, "The rise of 'big data' on cloud computing : review and open research issues", *Inf. Syst*., **47**, 98-115.

Alam, M. and Amjad, M., 2018, "Segmentation and classification of Brain MR Images Using Big Data Analytics", *Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)*, 1-5, IEEE.

Alam, M. and Amjad, M., 2019, "Rainfall forecasting using parallel and distributed analytics approaches on big data clouds", *Journal of Discrete Mathematical Sciences and Cryptography*, **22**, 4, 687-695.

Alam, M. and Amjad, M., 2017, "Proposed Tools and Techniques of Parallel Analytics of Big Data in the Cloud Environment", *International journal of emerging technology and advance engineering*, **7**, 241-246.

Alam, M. and Amjad, M., 2019, "Weather forecasting using parallel and distributed analytics approaches on big data clouds", *Journal of Statistics and Management Systems*, **22**, 4, 791-799.

Baldauf, M., Garlappi and Yannelis, L., 2020, "Does climate change affect real estate prices? Only if you believe in it", *The Review of Financial Studies*, **33**, 3, 1256-1295.

Bhatt, A.,Patel, A., Chheda, H. and Gawande, K.,2015, "Amazon Review Classification and Sentiment Analysis", *International Journal of Computer Science and Information Technologies*, **6**, 6, 5107-5110.

Bi, X., Zhao, X., Wang, G., Zhang, P. and Wang, C., 2015, "Distributed extreme learning machine with kernels based on MapReduce", *Neurocomputing*, **149**, 456-463.

Cano, J. R., Herrera, F. and Lozano, M., 2003, "Using evolutionary algorithms as instance selection for data reduction: an experimental study", *IEEE Trans. Evol. Comput*.,**7**, 6, 561-575.

Coulouris, G., Dollimore, J., Kindberg, T. and Blair, G., 2011, "Distributed Systems: Concepts and Design", 5th ed. Addison-Wesley.

D. Boyd and K. Crawford, 2012, "Critical questions for big data", *Information Communication Soc.*, **15**, 5, 662-679.

Das, S. R. and Chen, M. Y., 2007, "Yahoo! for Amazon: Sentiment extraction from small talk on the web", *Management science*, **53**, 9, 1375-1388.

Dean, J. and Ghemawat, S., 2010, "MapReduce : a flexible data processing tool", *Commun. ACM*, **53**, 1, 72-77.

Engle, R., Giglio, F., Kelly, S., Lee, B. and Stroebel, H., 2020, "Hedging climate change news", *The Review of Financial Studies*, **33**, 3, 1184-1216.

Fan, W. and Bifet, A., 2012, "Analysis big data: current status, and forecast to the future", *ACM SIGKDD Explor. Newslett*., **14**, 2, 1-5.

Fernandez, A., del Rio, S., Lopez, V., Bawakid, A., del Jesus, M. J., Benitez, J. M. and Herrera, F., 2014, "Big data with cloud computing : an insight on the computing environment, MapReduce, and programming frameworks", *Wiley Interdiscip. Rev.*, **4**, 5, 380-409.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H. and Herrera, F., 2012, "A review on ensembles for the class imbalance problem : bagging-, boosting- and hybrid-based approaches", *IEEETrans. Syst. Man Cybern.*, Part C, **42**, 4, 463-484.

García, S., Derrac, J., Cano, J. R. and Herrera, F., 2012, "Prototype selection for nearest neighbor classification : taxonomy and empirical study", *IEEE Trans. Pattern Anal. Mach. Intell*., **34**, 3, 417-435.

Gottlieb, A. and Almasi, G., 1989, "Highly Parallel Computing", Benjamin-Cummings Publishing.

Iqbal, R., Doctor, F., More, B., Mahmud, S. and Yousuf, U., 2020, "Big data analytics : Computational intelligence techniques and application areas", *Technological Forecasting and Social Change*, **153**, 119253.

Katal, A.,Wazid, M. and Goudar, R., 2013, "Big data : issues, challenges, tools and good practices", *Proceedings of the International Conference on Contemporary Computing*, 404-409.

Kaur, R. and Kaur, K., 2017, "Data Mining on Customer Segmentation: A Review", *International Journal of Advanced Research in Computer Science*, **8**, 5, 201-2012.

Kittler, J., Hatef, M., Duin, R. P. W. and Matas, J., 1998, "On combining classifiers", *IEEE Trans. Pattern Anal. Mach. Intell*., **20**, 3, 226-239.

Kollios, G., Gunopulos, D., Koudas, N. and Berchtold, S., 2015, "Efficient biased sampling for approximate clustering and outlier detection in large data sets", *IEEE Trans Knowledge Data Engineering*, **15**, 5, 1170-1187.

Lopez,V., del Rio, S., Benitez, J. M. and Herrera, F., 2015, "Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce frame work for imbalanced big data", *Fuzzy Sets Syst*., **258**, 5-38.

Mahmoud, Hussam, 2020, "Barriers to gauging built environment climate vulnerability", *Nature Climate Change*, 1-4.

Mandale, M., Ashwini, B. and Jadhawar, M., 2015, "Weather Forecast Prediction: A Data Mining Application", *International Journal of Engineering Research and General Science*, **3**, 2, ISSN 2091-2730.

Mármol, F. G., Girao, J. and Perez, G. M., 2010, "TRIMS, a privacy-aware trust and reputation model for identity management systems", *Computer Networks*, **54**, 6, 2899-2912.

Mayer Schonberger, V. and Cukier, K., 2014, "Big Data : A Revolution That Will Transform How We Live, Work, And Think", Eamon Dolan/Mariner Books.

Tsai, C. F., Lin, W. C. and Ke, S. W., 2016, "Big data mining with parallel computing : A comparison of distributed and MapReduce methodologies", *Journal of Systems and Software*, **122**, 83-92.