Jestr

Journal of Engineering Science and Technology Review 15 (4) (2022) 111 - 116

JOURNAL OF
Engineering Science and Technology Review

Research Article

www.jestr.org

Ensemble-based Model for Rainfall Nowcasting using Automatic Weather Station Data

Nita H. Shah[1,*], Bipasha Paul Shukla[2] and Anupam Priamvada[1]

[1]*Department of Mathematics, Gujarat University, Ahmedabad-380009*
[2]*Atmospheric Sciences Division, Space Applications Centre, ISRO, Ahmedabad-380015 Gujarat, India*

Received 21 June 2022; Accepted 4 August 2022

Abstract

Precipitation nowcasting is important for a multitude of applications like the planning and preparation of flights in the aviation industry and also for daily routine management. Existing methods are complex, computationally extensive. Also, inherent model configurations involved in assimilation and simulation of current atmospheric state limit their application in forecasting precipitation for next two to four hours. So, in the proposed paper, the development of a data driven algorithm for precipitation nowcasting using the Automatic Weather Station (AWS) data obtained from Vikram Sarabhai Space Centre, Thiruvananthapuram has been carried out. Three different artificial intelligence (AI) models have been developed based on ensemble learning techniques because of low variance and robustness i.e., bagging and boosting, and the potential of these ensembles viz. Random Forest, Adaboost, and Xgboost have been investigated. The accuracy of Random Forest and Xgboost are comparable and slightly better than Adaboost in predicting the occurrence of rainfall events. The model can capture the instances with good accuracy of 80 % for the Random Forest and Xgboost and 65 % for the Adaboost. Sensitivity of the in-situ observations has been carried out. The results indicate that for the prediction of rainfall, time series of pressure and temperature are found to be more influential than other predictors in all the State-of-the-art, machine learning techniques. The study is important from the perspective to apply as an approach for rainfall nowcasting applications and also can be applied for the other heterogenous and dynamic systems.

Keywords: Nowcasting, Precipitation, Random Forest, Adaboost, Xgboost

1. Introduction

Rainfall estimation is important for the agriculture, aviation industry, resource-planning, water system management, and for generating alerts in case of extreme catastrophic events. There are numerous ways to predict rainfall, few of them are satellite-based data, ground-based observations, or radar data, numerical modelling is operational but accurate prediction of weather is still an open and challenging task among scientists. Herman and Schumacher, [6] atmospheric system is chaotic in nature so statistical or dynamical—will necessarily have formulaic limitations, systematic biases, and failure modes regardless of the level of care exercised during model construction. Prudden et al. [16] Numerical weather prediction models were able to capture mesoscale weather patterns but not the smaller-scale convective patterns that occur within mesoscale systems. Thus, these models had limited utility in nowcasting rainfall in the next two to four hours because of its dependence on the dynamic status of atmosphere and frequent update of these dynamic variables for simulation and assimilation. In recent years, machine learning shows great potential in a variety of domains i.e., image segmentation, voice recognition, and image recognition. So, Prudden et al. [16] Machine Learning algorithms can resolve complex non-linear hierarchies of information. Moreover, its flexibility and conceptual simplicity make it a promising approach to utilise for more diverse sources of information as well as for more complex systems.

Machine learning algorithms have been used for prediction strategies for a variety of purposes. Das et al. [4] uses random forest-based machine learning algorithm for nowcasting of convective rain with a ground-based radiometer using brightness temperatures measured at 14 frequencies (7 frequencies in 22–31 GHz band and 7 frequencies in 51–58 GHz bands) and result indicates that random forest algorithm with fixed alarm generation time of 30 min and 60 min performs quite well (probability of detection of all types of weather condition ~90%) with low false alarms. Barrera-Animas et al. [2] use Machine Learning algorithms in conjunction with time-series data for hourly rainfall estimation. Wang, et al. [22] improves the accuracy of sub seasonal forecasting of China precipitation with a machine learning approach proposed by Hwang et al. in [8] to predict the precipitation in China 2–6 weeks in advance. Wang et al. [22] used a non-linear regression model and chosen 21 meteorological elements as predictors to integrate diverse meteorological observation. Singh et al. [20] developed several hybrid forecasting models which are combinations of two feature selection techniques, Gradient boosting and Random Forest with various machine learning techniques, viz Support Vector Machine (SVM), Adaboost, Neural Network (NN) and K-Nearest Neighbour (KNN) to predict rainfall in town of carry, North Caroliana and found Adaptive boosting algorithm (Adaboost) outperforms the other algorithm based on the F-score of 0.9726 on testing data. Anwar et al., [1] build a multivariate rainfall prediction

*E-mail address: nitahshah@gmail.com
ISSN: 1791-2377
doi:10.25103/jestr.154.16

model using the Extreme Gradient Boosting (Xgboost) on historical weather data and found that the model is capable of producing accurate predictions for daily rainfall estimates. Mai [11] developed an algorithm to classify rain or shine weather forecast in precipitation nowcasting based on XGBoost. Kunjumon [9] weather predictions are needed for daily activities and it was one of the main challenging problems facing throughout the world because it consists of multidimensional and nonlinear data and surveyed the various methods and algorithms used for weather prediction in the field of data mining are supervised and unsupervised machine learning algorithms. Holmstrom et al. [7] used machine learning for weather forecasting based on a linear regression model and a variation on a functional regression model to forecast the maximum temperature and the minimum temperature for seven days by employing weather data for the past two days as predictors. Xu et al., [24] formed a data fusion method for obtaining land surface temperature with a high spatial resolution based on Random Forest. Ling et al., [10] evaluated machine learning algorithms for the prediction of regions of high Reynolds averaged Navier Stokes uncertainty and provides alternate approaches using convex combinations of regularized regression approaches and randomized sub-sampling in combination with feature selection algorithms.

Ensemble learning consists of multiple learning model to have better performance. Opitz and Maclin, [14] Previous research has shown that an ensemble is often more accurate than any of the single classifiers in the ensemble. Rokach, [19] An ensemble is largely characterized by the diversity generation mechanism and the choice of its combination procedure. Dietterich [5]; Zounemat-Kermani et al. [25] Ensemble learning can overcome shortcomings related to single models, including bias, variance, computational problems representation problems and overfitting. So, the choice of algorithms based on real-life problems is still an open and active field of research.

Raschka, [18] A good choice of model is a model with accurate predictions along with it generalizes well to unseen data. Parmezan et al. [15] poor model selection results into reduced performance and increase uncertainty given the error accumulation and these problems become more visible as the prediction horizon grows, in most of the real applications, it is the only strategy feasible to predict large horizons.

In this paper, we explore the application of machine learning algorithms viz. Random Forest, Adaboost, and Xgboost to generate more accurate weather forecasts for the next 4 hours. The scope of this paper was restricted to forecasting the precipitation for the next four hours, given the time series of humidity, temperature, atmospheric pressure, sunshine, and rainfall data for the past four hours, and also analyses the sensitivity of predictors on different ensemble-based machine learning algorithms i.e., Random Forest, Adaboost and Xgboost. These data have been taken from Automatic Weather Station data. The rest of the paper is organized as follows. Section 2 explains the study region and data. Section 3 introduces the overall methodology. Section 4 provides the results using different metrics. Section 5 gives the conclusion of the study.

2. Study Region and Data

The study area chosen is Vikram Sarabhai Space Centre (VSSC) Thiruvananthapuram, India which is demonstrated in Figure 1, which is located in extreme southwest India, is in the state of Kerala. It is surrounded by western ghats in its east and the Arabian Sea in the west. Its geographical location on the map is 8.5421° North and 76.8627° East. The maximum temperature of this region in the years 2008 and 2009 is 21.8° Celsius and the minimum temperature is 13.66° Celsius. Wind speed varies from 0 to 5.512 m/s. The atmospheric pressure of this region lies from 925 to 1015.57 hectopascal. The humidity of this region is found to be as high as 99.5 % in June-July and drops to 16 %. The study uses the data from July 2008 and July 2009.

3. Methodology

Our proposed article is experimented with Random Forest, Adaboost, and Xgboost algorithms to predict the rainfall events in the next few hours by taking the previous 4 hours' predictors such as humidity, pressure, temperature, wind speed, sunshine, and rainfall. The algorithms used are developed under the framework of Scikit-Learn: Machine Learning in Python developed by Pedregosa [17]. Python software language used in this study is developed by VanRossum [21]. The algorithms used are given in https://colab.research.google.com/drive/1DUxbpvD1VFzwnloGezSxzo2V79WDenk?authuser=4#scrollTo=_dyUFgifioG1. Training test data is 80 % and testing and validation set of data is 20 %. In the Random Forest classifier, the pruning of trees is done by using parameters “ccp_alpha” which controls the size of trees (Breiman et.al., 1984). These are the hyperparameters used in Random Forest i.e., bootstrap': True, 'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': 'auto', 'max_leaf_nodes': None, 'max_samples': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'n_estimators': 10, 'n_jobs': None, 'oob_score': False, 'random_state': 0, 'verbose': 0, 'warm_start': False.). These are the hyperparameters used in Adaboost i.e., algorithm': 'SAMME.R', 'base_estimator': None, 'learning_rate': 1.0, 'n_estimators': 10, 'random_state': 0. These are the hyperparameters used in XGBoost i.e., 'base_score': 0.5, 'booster': 'gbtree', 'colsample_bylevel': 1, 'colsample_bynode': 1, 'colsample_bytree': 1, 'eta': 0.2, 'gamma': 0, 'learning_rate': 0.1, 'max_delta_step': 0, 'max_depth': 3, 'min_child_weight': 1, 'missing': None, 'n_estimators': 10, 'n_jobs': 1, 'nthread': None, 'objective': 'binary:logistic', 'random_state': 0, 'reg_alpha': 0, 'reg_lambda': 1, 'scale_pos_weight': 1, 'seed': 123, 'silent': None, 'subsample': 0.4, 'verbosity': 1.

The experimentation setup of the methodology is explained in Fig.2. Data is pre-processed and features are extracted based on variance reduction, to know the important predictors. data is oversampled to minimize the imbalance of class.

3.1. Random Forest (RF) Algorithm:

Random Forest is a begging technique, and it is based on multiple decision trees formed on the bootstrap sample. In the process of constructing a decision tree, the first step is the selection of the root node as proposed by Breiman, L. [3]. Wang et al. [23] The basis for selecting a root node is in such a way that samples contained in one split node belong to the same category as much as possible.

3.2. Gini Index

Mao and Sorteberg [12] For classification, a commonly used cost function is the Gini impurity which measures the degree

of homogeneity of the groups created by each split. Gini impurity is given by

$$\text{GiniImpurity} = H(Q_i) = \sum_j (1 - p_{ij}) p_{ij}. \quad (1)$$

where p_{ij} be the predicted probability for the region. The target classification outcome is the binary proportion of class 0 or 1 observation in node Q_i, i is a terminal node, and k is the binary observations in node. n_i is the total number of data in node i belongs to class k.

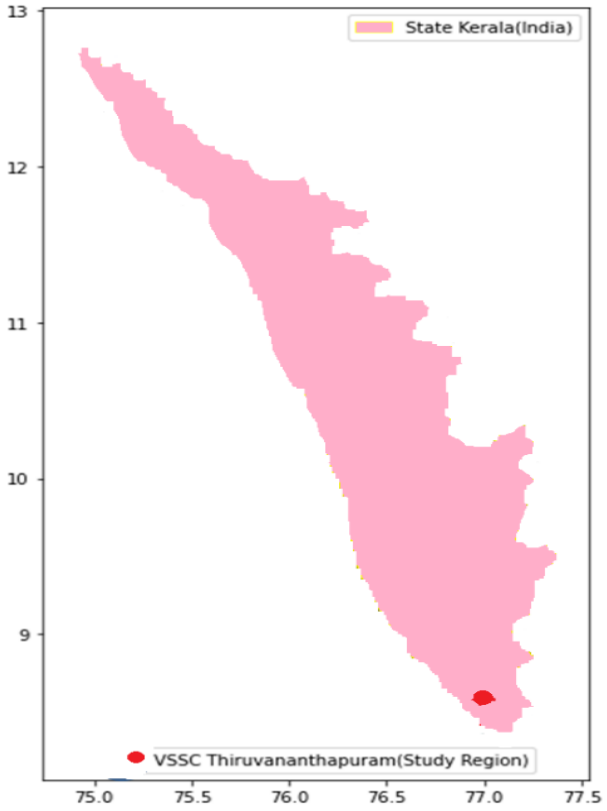


Fig. 1 Study Region.

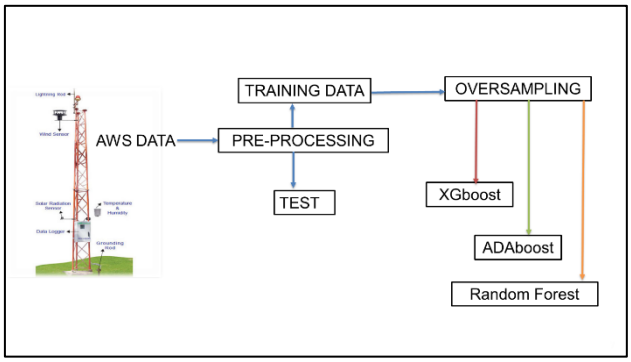


Fig.2. Scheme of the Proposed Methodology.

$$p_{ij=\frac{1}{n_i}} \sum_{y \varepsilon Q_i} i(y = k) \quad (2)$$

Mitchell et al. [13] In the decision tree, construction of trees as it reaches nodes from the root in a greedy manner. Information gain is the measure of the value of the split. Information gain describes a change in Gini Impurity from the previous step to a new step. Information gain is given by

$$\text{InformationGain}(Q_i, t_{left}, t_{right}) = Q_i - \left(\frac{n_{left}}{n_{total}}\right) H(Q_i(left)) - \left(\frac{n_{right}}{n_{total}}\right) H(Q_i(right)) \quad (3)$$

where n_{left},n_{right} and n_{total} be the total amount of data in the respective sets.

3.3 Adaboost Algorithm

Adaboost classifier is the ensemble learning technique based on boosting technique. The algorithm gives the weightage to the misclassified events in every step and improves the misclassification error. The predictions from all of them are then combined through a majority vote to produce the final prediction. Node is generated, based on Gini impurity. In this way, the decision tree is grown, and leaf and branch nodes are selected based on information gained.

3.4 Xgboost Algorithm

Xgboost is boosting algorithm based on gradient descent, it has a decision tree as a base estimator, then the model performance is boosted by using gradient descent on a differentiable loss function that measures the performance of the model on the training set iteratively. It is regularised gradient descent algorithm which has a regularisation parameter. The objective function has a loss function f_k. Taylor expansion of the function to the second-order allows used to calculate loss functions:

$$\text{Objective} = \sum_i l(x_i, y_i) + \sum_k \Omega f_k \quad (4)$$

3.5 Model Performance

To evaluate the model performance, accuracy is not a sufficient metric as data is highly imbalanced. So, probability of detection (POD) is also used along with accuracy. True Positive (TP) is the number of events correctly classified for rain and True Negative (TN) refers the which refer to correctly predicted classes of no rain events. False Positive (FP) is those events where actually no rain occurred but is predicted as rain. False-Negative (FN) corresponds to those events where there is no rain but is predicted as rain. Accuracy evaluates the percentage of correctly identified events from the total events.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} 100 \quad (5)$$

Probability of detection (POD), which identifies the percentage of events correctly.

$$\text{Probability of detection} = \frac{TP}{TP+FN} 100 \quad (6)$$

4. Results and Discussions

4.1 Sensitivity Analysis

The sensitivity analysis is done by extracting features based on variance reduction, calculated with Gini Impurity for each algorithm i.e., Random Forest, Adaboost, and Xgboost. These predictors are arranged in the sequence of decreasing order of variance with different lead times. Fig. 3(a) demonstrates the most seven important features for lead-time 1-hour. According to Fig. 3(a), The model based on Random Forest is most sensitive to temperature at the current state for predicting rainfall for the next 1-hour. The model based on Adaboost is most sensitive to pressure at the current state for

predicting rainfall for the next 1-hour. The model based on Xgboost is most sensitive to rainfall at the current state for predicting rainfall for the next 1-hour.

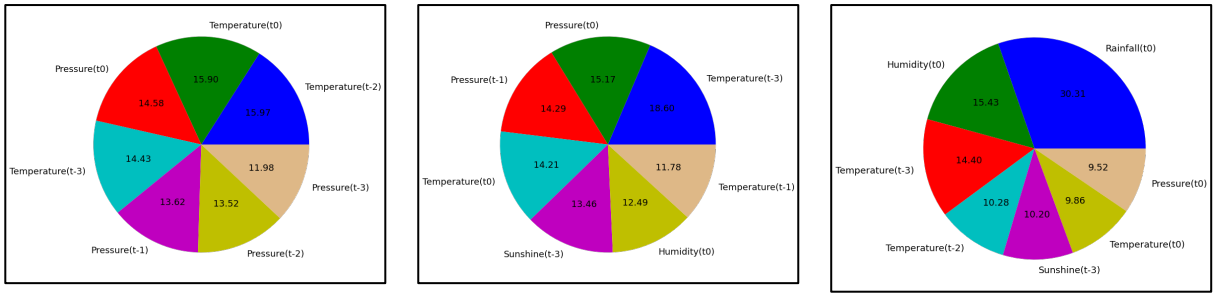
Fig. 3(b) demonstrates the most seven important features for lead-time 2 hours. According to Fig. 3(b), the Random Forest model for a lead hour 2 is most sensitive to temperature at t-2-time steps. Adaboost and Xgboost model for a lead hour 2 is most sensitive to temperature at t-3-time steps.

Fig. 3(c) demonstrates the most seven important features for lead-time 3 hours. According to Fig. 3(c), the Random Forest model for a lead hour 3, is most sensitive to temperature at t-2-time steps. Adaboost model for a lead hour 3 is most sensitive to humidity at t-2-time steps. Xgboost model for a lead hour 3 is most sensitive to temperature at the current state.

Fig. 3(d) suggests that the most important seven features for lead-time 4 hours. According to Fig. 3(d), the most influential feature which impacts the rainfall for a lead time of 4 hours is the temperature at $(t-2)$ - time steps for the Random Forest-based model. Adaboost-based model is most sensitive to pressure at $(t-2)$ - time-steps and the Xgboost-based model is sensitive to temperature at $(t-1)$ - time steps.

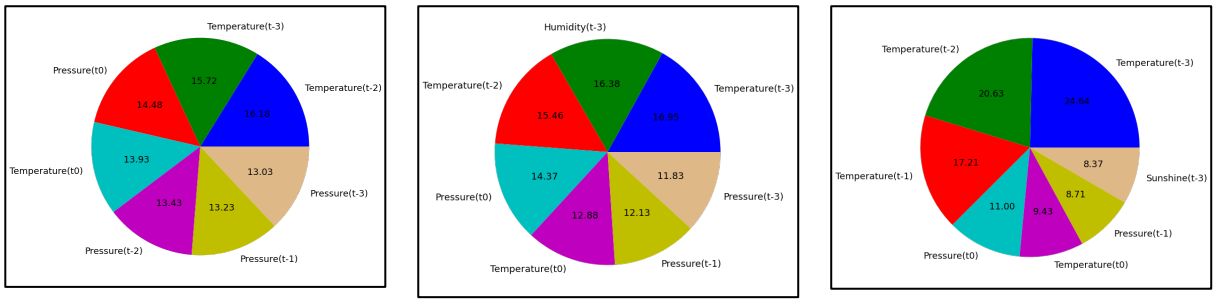
4.2. Model Performance

The accuracy talks about the percentage of correctly predicted events (rain or no rain) among all the cases. The model can capture the instances with good accuracy of 80 % for the Random Forest and Xgboost. The model can capture the instances with good accuracy of 65 % for the Adaboost as demonstrated in Fig.4(a).



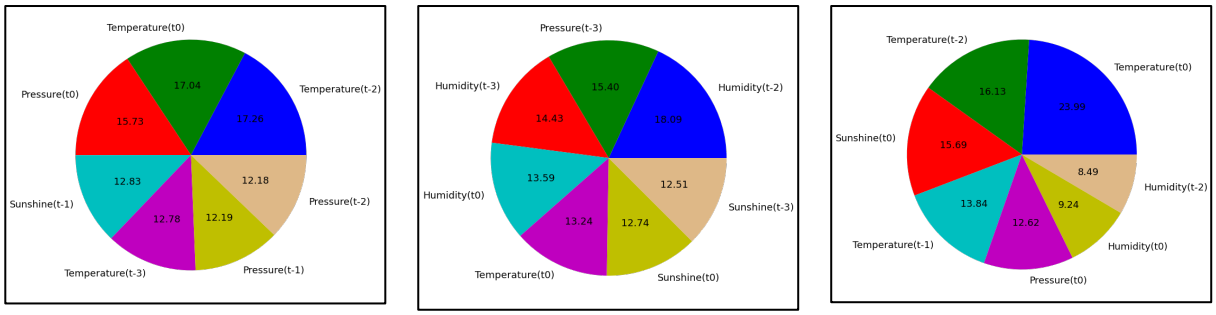
Random Forest **Adaboost** **Xgboost**

(a)



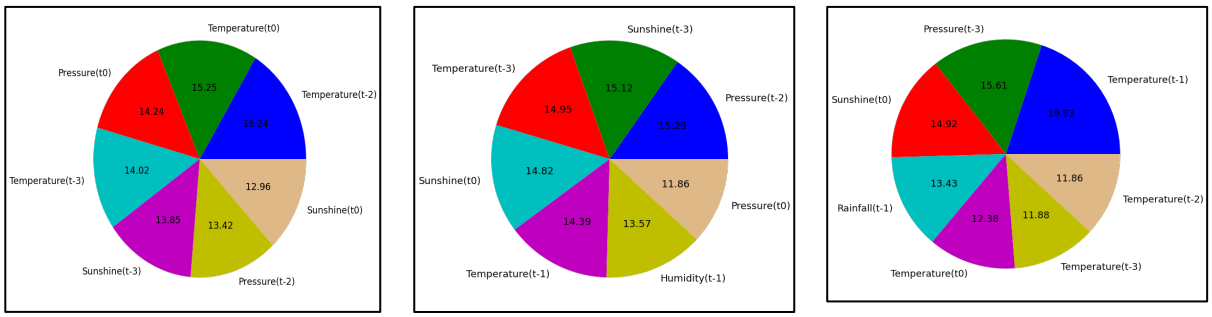
Random Forest **Adaboost** **Xgboost**

(b)



Random Forest **Adaboost** **Xgboost**

(c)



(d)

Fig. 3. (a) Influential Feature (1-hour lead- time), (b) Influential Feature (2-hour lead- time), (c) Influential Feature (3-hour lead- time), (d) Influential Feature (4-hour lead- time).

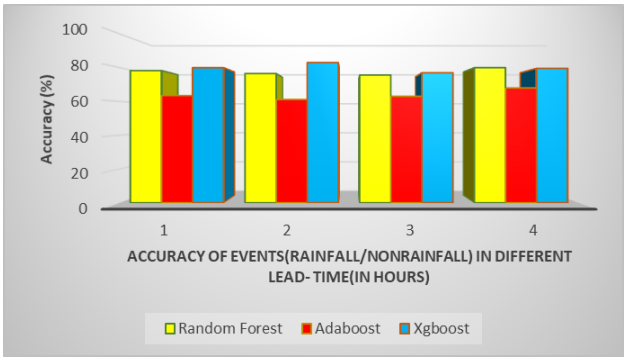


Fig.4. (a) Comparison of Model Performance Based on Accuracy.

To get the better performance of Random Forest, Adaboost and Xgboost classifiers is tested based on the metrics such as accuracy and probability of detection.

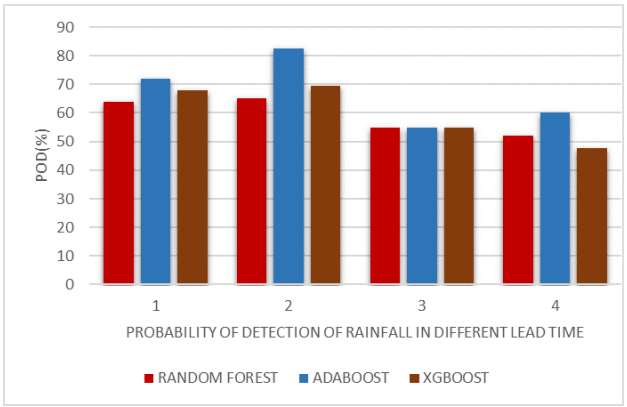


Fig.4. (b) Comparison of Model Performance Based on POD.

The probability of detection of rainfall for the Random Forest model is 65 % average for all lead hours. For the Xgboost model is 60.25 %, the average probability of detection is 60 % and for Adaboost is 66.75 % as demonstrated in Fig.4(b). For lead time 1 to 2 hours, Adaboost is better than Xgboost and Random Forest. For a lead time of 3 hours, all the model shows the same probability of detection. For lead time 4-hour, Adaboost performance surpasses the other two models.

5. Conclusions

In this paper, the potential of machine learning approaches based on Random Forest, Adaboost, and Xgboost have been investigated for sensitivity towards nowcasting of precipitation for the next 4 hours. The results indicate that for the prediction of rainfall, time series of pressure and temperature are found to be more influential in all the approaches than the other predictors. The accuracy of Random Forest and Xgboost are comparable and slightly better compared to Adaboost. In future, the purposed algorithm will be tested on data from different AWS to have a robust prediction mechanism.

Acknowledgments

The authors are indebted to reviewers for their constructive suggestions. The authors acknowledge Shri Nilesh M. Desai, Director, Space Applications Centre (SAC), Ahmedabad ISRO for his kind support. The AWS data for the study was downloaded from the website www.mosdac.gov.in

References

1. Anwar, M. T., Winarno, E., Hadikurniawati, W., & Novita, M. (2021, April). Rainfall prediction using Extreme Gradient Boosting. In *Journal of Physics: Conference Series* (Vol. 1869, No. 1, p. 012078). IOP Publishing.
2. Barrera-Animas, A. Y., Oyedele, L. O., Bilal, M., Akinosho, T. D., Delgado, J. M. D., & Akanbi, L. A. (2022). Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. *Machine Learning with Applications*, *7*, 100204.
3. Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.
4. Das, S., Chakraborty, R., & Maitra, A. (2017). A random forest algorithm for nowcasting of intense precipitation events. *Advances in Space Research*, *60*(6), 1271-1282.
5. Dietterich, T. G. (2002). Ensemble learning. *The handbook of brain theory and neural networks*, *2*(1), 110-125.
6. Herman, G. R., & Schumacher, R. S. (2018). "Dendrology" in numerical weather prediction: What random forests and logistic regression tell us about forecasting extreme precipitation. *Monthly Weather Review*, *146*(6), 1785-1812.

7. Holmstrom, M., Liu, D., & Vo, C. (2016). Machine learning applied to weather forecasting. *Meteorol. Appl*, *10*, 1-5.
8. Hwang, J., Orenstein, P., Cohen, J., Pfeiffer, K., & Mackey, L. (2019, July). Improving subseasonal forecasting in the western US with machine learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2325-2335).
9. Kunjumon, C., Nair, S. S., Suresh, P., & Preetha, S. L. (2018, March). Survey on weather forecasting using data mining. In *2018 Conference on Emerging Devices and Smart Systems (ICEDSS)* (pp. 262-264). IEEE.
10. Ling, J., & Templeton, J. (2015). Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty. *Physics of Fluids, 27*(8), 085103.
11. Mai, X., Zhong, H., & Li, L. (2020, August). Research on rain or shine weather forecast in precipitation nowcasting based on XGBoost. In *The International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery* (pp. 1313-1319). Springer, Cham.
12. Mao, Y., & Sorteberg, A. (2020). Improving radar-based precipitation nowcasts with machine learning using an approach based on random forest. *Weather and Forecasting*, *35*(6), 2461-2478.
13. Mitchell, R., & Frank, E. (2017). Accelerating the XGBoost algorithm using GPU computing. *PeerJ Computer Science*, *3*, e127.
14. Opitz, D., & Maclin, R. (1999). L2-Boosting algorithm applied to high-dimensional problems in genomic selection. *Journal of Artificial Intelligence Research*, *11*, 169-198.
15. Parmezan, A. R. S., Souza, V. M., & Batista, G. E. (2019). Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information sciences*, *484*, 302-337.
16. Prudden, R., Adams, S., Kangin, D., Robinson, N., Ravuri, S., Mohamed, S., & Arribas, A. (2020). A review of radar-based nowcasting of precipitation and applicable machine learning techniques. *arXiv preprint arXiv:2005.04988*.
17. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.
18. Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*.
19. Rokach, L. (2010). Ensemble-based classifiers. *Artificial intelligence review*, *33*(1), 1-39. doi:10.1007/s10462-009-9124-7.
20. Singh, G., & Kumar, D. (2019, January). Hybrid prediction models for rainfall forecasting. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 392-396). IEEE.
21. VanRossum, G. (1995). Python reference manual. *Department of Computer Science* [CS], (R 9525).
22. Wang, C., Jia, Z., Yin, Z., Liu, F., Lu, G., & Zheng, J. (2021). Improving the accuracy of subseasonal forecasting of China precipitation with a machine learning approach. *Frontiers in Earth Science*, *9*, 659310.
23. Wang, C., Wang, P., Wang, D., Hou, J., & Xue, B. (2020). Nowcasting multicell short-term intense precipitation using graph models and random forests. *Monthly Weather Review*, *148*(11), 4453-4466.
24. Xu, S., Cheng, J., & Zhang, Q. (2021). A Random Forest-Based Data Fusion Method for Obtaining All-Weather Land Surface Temperature with High Spatial Resolution. *Remote Sensing*, *13*(11), 2211.
25. Zounemat-Kermani, M., Batelaan, O., Fadaee, M., & Hinkelmann, R. (2021). Ensemble machine learning paradigms in hydrology: A review. *Journal of Hydrology*, *598*, 126266.