Short-Term Rainfall Prediction Using Supervised Machine Learning

Nusrat Jahan Prottasha[1,*], Anik Tahabilder[2], Md Kowsher[3], Md Shanon Mia[1], Khadiza Tul Kobra[1]

[1]Department of Computer Science, Daffodil International University, Dhaka, Bangladesh

[2]Department of Computer Science, Wayne State University, Detroit, Michigan, USA

[3]Department of Computer Science, Stevens Institute of Technology, Hoboken, New Jersey, USA

Received 30 August 2021; received in revised form 16 May 2022; accepted 02 June 2022

DOI: https://doi.org/10.46604/aiti.2023.8364

Abstract

Floods and rain significantly impact the economy of many agricultural countries in the world. Early prediction of rain and floods can dramatically help prevent natural disaster damage. This paper presents a machine learning and data-driven method that can accurately predict short-term rainfall. Various machine learning classification algorithms have been implemented on an Australian weather dataset to train and develop an accurate and reliable model. To choose the best suitable prediction model, diverse machine learning algorithms have been applied for classification as well. Eventually, the performance of the models has been compared based on standard performance measurement metrics. The finding shows that the hist gradient boosting classifier has given the highest accuracy of 91%, with a good F1 value and receiver operating characteristic, the area under the curve score.

Keywords: rain prediction, machine learning, supervised classification, agriculture resource, crops yield

1. Introduction

Agriculture plays a vital role in the development of many developing countries [1]. IoT-based smart agriculture model is being implemented worldwide to increase crop yields. The use of intelligent tools in farming can increase the production of crops and also minimize the damage due to disasters. The economy of South Asian countries, including Bangladesh, India, China, and Pakistan, depends more on agriculture. But there are always some natural disasters, including rain and floods, that create huge demolition of crops and property.

Therefore, a good rain prediction model is necessary to forecast the rain to reduce the risk to life and also to maintain the agriculture farms in a better way. In addition, a rain prediction model helps farmers take early flood measurements and properly manage water resources.

Observing the significance of rain prediction, researchers have developed a lot of devices to predict rainfall, but none of them is worth noting in terms of short-term rain prediction. Hence, it has not been adopted eventually by the end-level user to forecast the rain situation. However, machine learning techniques can make a more accurate prediction because of their underlying technology. Researchers have implemented neural networks (NN) in rainfall prediction and showed that the NN-based model usually exceeds the performance of the numerical weather prediction model.

This study aims to develop a short-term rain prediction model that can effectively and accurately predict rainfall. In this proposed work, several relevant machine learning models have been used to predict rainfall, and finally, a performance comparison has been made to determine the best suitable model.

* Corresponding author. E-mail address: jahannusratprotta@gmail.com
Tel.: +8801778-111461

In this project, the twenty-nine most optimistic classifiers have been used from eleven different categories. All these models have been trained and tested with a relevant rainfall dataset to implement this prediction model. The data was collected from a popular and recognized public repository and split into training, validation, and testing data. Since the raw data came from natural weather resources, it has been preprocessed before going to the training phase. A few preprocessing techniques have been implemented to prepare the raw data, such as missing value check, feature selection, features scaling, dimension reduction, etc. After analyzing all the models and comparing them, it was found that the hist gradient boosting classifier (HGBC) has shown the highest accuracy of 91%. A lot of other models have shown the second-highest accuracy of 90%. The contribution of this paper can be summarized below:

(1) A pipeline for estimating rain prediction has been developed.

(2) Diverse types of classifiers have been used to ensure the best model that suits diverse types of data.

(3) A comparison among all trained models has been described to measure the comparative performance.

The rest of the sections of this paper is organized as follows. Section 2 explains the related work of various classification techniques for rainfall prediction. Section 3 describes the major technology components used, including the dataset, preprocessing, and algorithm. Section 4 describes the methodology that has been used to solve the proposed problem. Later on, Section 5 contains the experiments and the results. This article is wrapped up in section 6 by discussing the conclusion and future works.

2. Related Work

A country's agriculture largely depends on rain, and there is a lot of research on forecasting rain. All the earlier methods of rain forecasting are mainly statistical and numerical analysis based [1]. Also, some methods predict the rain by analyzing radar images. Denœux and Rizand [2] have proposed a model that performs deep learning-based analysis on radar images to predict rainfall.

However, with the recent advancement of machine learning, many new machine learning-based models have been proposed for rainfall prediction. Researchers like Shah et al. [3] have developed a simple polynomial regression-based model to predict the rain to benefit agricultural products. Asha et al. [4] have proposed a hybrid machine learning classification model for predicting rainfall, and it has shown better performance than the ordinary ml-based model. Sakthivel and Thailambal [5] have also demonstrated such a hybrid approach for rain prediction, predicting continuous long period rainfall. Naidu et al. [6] presented the changes in rainfall patterns in numerous agro-climatic zones using machine learning approaches. Besides, Dinh et al. [7] have used a support vector machine (SVM)-based method to measure the rain forecast and the soil erosion due to the rain.

On the other hand, Abdel-Kader et al. [8] showed a vigorous hybrid technique by particle swarm optimization (PSO) and multi-layer perceptron (MLP) for the prediction of rainfall. Also, Samsiahsani et al. [9] evaluated many machine learning classifiers based on Malaysian data for rainfall prediction. Similar models have been developed to predict the flood forecast due to heavy rainfall and ocean waves. Luk et al. [10] mentioned data scarcity as a limitation in modeling such a predictive model. Abbot and Marohasy [11] have shown the application of NN in rainfall prediction based on a dataset from Queensland, Australia.

Among the short-term prediction models, a model by Shah et al. [12], is a good invention for predicting very short-term rainfall. On the other hand, some researchers focused on heavy rainfall only. Research by Sangiorgioet al. [13] has made improvements in determining heavy rainfall based on water vapor measurement using a NN-based model. Han et al. [14] have mentioned the limitations of such a predictive model for forecasting rainfall and flood by determining the major uncertainties.

Unlike those works, in this project, several rain-forecasting models for upcoming rain prediction have been developed using twenty-nine different machine learning classifiers. Additionally, their performances have also been compared, and hence the best machine learning model for rain prediction has been determined.

3. Dataset and Algorithm Description

The structural dataset and algorithms are a machine learning model's two most essential parts. This section will provide a brief description of the dataset and algorithm. The features of the dataset will be discussed in more details form. Then the data preprocessing steps will also be explained in the subsequent section. Finally, all the algorithms that have been used to make the models will be described in brief.

3.1. Dataset

To implement the proposed model, the rain prediction database from Kaggle has been used. The dataset [15] comprises the precipitation estimation from the years 1901 to 2015 for each state of Australia. Each observation contains 19 qualities (person-months, annual, and combinations of 3 continuous months) for subdivisions. The rain prediction unit mentioned in the dataset is measured in millimeters (mm). Table 1 summarizes the dataset and its features. The feature name and the description of the feature are shown in the left column and the right-side column, respectively. The dataset is robust and has an adequate number of observations and features, which ensures the quality of the data and guarantees a model with good accuracy if modeled properly.

Table 1 The description of the dataset

Feature	Details description of the feature
Location	This is the name of the location
MaxTemp	Max temperature recorded in degrees centigrade
MinTemp	Min temperature recorded in degrees centigrade
WindGustSpeed	The speed of wind gust in km per hour
WindGustDir	The direction of the strongest wind gust
WindSpeed9am	Wind speed averaged over 10 minutes before 9 a.m.
WindDir9am	The direction of the wind gust at 9 a.m.
WindSpeed3pm	Averaged wind speed over 10 min before 3 p.m.
WindDir3pm	The direction of the wind gust at 3 p.m.
Humidity9am	Relative humidity at 9 a.m. measured in percentage
Humidity3pm	Relative humidity at 3 p.m. measured in percentage
Temp9am	The temperature at 9 a.m. measured in degrees Celsius
Temp3pm	The temperature at 3 p.m. measured in degrees Celsius
Pressure9am	Atmospheric pressure at 9 a.m. measured in hPa
Pressure3pm	Atmospheric pressure at 3 p.m. measured in hPa
Rainfall	The rainfall recorded for the day (mm)
RainToday	If precipitation in the 24 h to 9 a.m. exceeds 1 mm it will be 1, otherwise 0 (mm)
Rain tomorrow	If it will rain or not on the next day, it is output as binary target variable

3.2. Pre-processing

Data preprocessing is a crucial step that helps improve the quality of data to advance the extraction of important bits of knowledge from the data. It refers to preparing the raw data in a format that will be neat, clean, and understandable to the machine learning model.

The flowchart of data preprocessing steps is illustrated in Fig. 1. in the subsequent section. After the data is collected, it goes through a cleaning process, and then the missing values are handled logically. Then the data encoding is performed, and important features get selected. Eventually feature scaling process is used to bring all the features on a similar scale, and 10-fold cross-validation (CV) is implemented. Each stage of data preprocessing is described elaborately below.

Data cleaning is the method of preparing data for analysis by removing or adjusting incorrect, fragmented, unimportant, duplicated, or improperly organized data. It includes further tuning of data by fixing spelling, and syntax errors, and adjusting mistakes, such as dealing with empty data, invalid values, and recognizing duplicate data points. There were some missing, some duplicates, some invalid, and some incomplete values in this dataset. The following actions have been taken to correct those data.

(1) A part of the data points is repeated in row and column segments. Hence, all the duplicate data was removed, and only a single instance was kept.

(2) A few rows and columns were almost empty. That corresponding row or column has been removed from the dataset.

(3) Some instances were shown to have some invalid values. That instance was also deleted to make the dataset solid and readable.

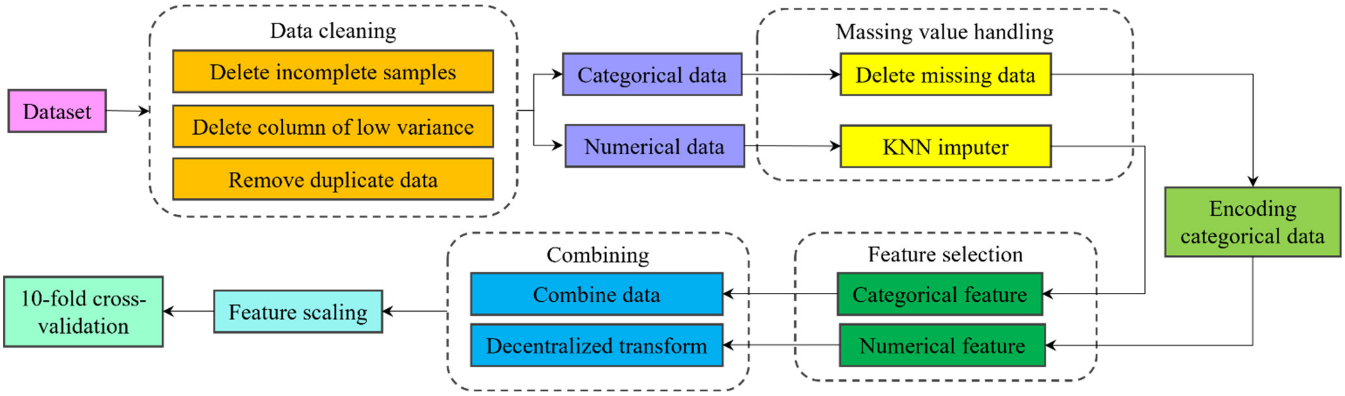

Fig. 1 The entire data preprocessing procedure

Missing values are a common occurrence in the dataset. Therefore, they need to be handled to prepare the dataset properly. If the data cannot pass the statistical test, then it is removed from the dataset. Besides, most predictive algorithms can't handle any missing value. Hence, this issue must be solved before the data is fed into the model. In most machine learning work, people utilize various techniques such as mean, median, and mode methods to handle missing values. But the most appropriate method for managing missing data is to remove the full row for categorical features and replace the missing data with the nearest neighbors for numerical data. The k-nearest neighbors (KNN)-based imputation method has been implemented in this work for a more accurate missing value imputation. The not-a-number (NaN) data has been replaced by getting the nearest value by considering the three neighbors.

Categorical data could be a subjective feature whose values are taken by label encoding or one-hot representation. It implies that categorical data must be encoded into numbers before it is fed to the model. In the dataset, there are six categorical factors “Location,” “WindGustDir,” “WindDir9am,” “WindDir3pm,” “RainToday,” and “Rain tomorrow.” One hot encoding is one of the most popular methods for encoding categorical variables, which has been used in this project. It is one of the widespread approaches, and it works well unless the categorical variables count is too high. It makes a new binary column for each category, demonstrating the presence of each possible value from the categorical data.

Feature selection is the strategy to figure out highly related input variables when creating a predictive model. Reducing the number of input variables is desirable to reduce the computational cost in modeling and increase the model’s performance. The dataset contains 21 components, and the p-value has been tested to check the probability of the null hypothesis. Features with a p-value less than 0.05 have been discarded. After checking multicollinearity, a vital separation is maintained from those components which appear redundant and don't back the p-value assumption. Besides, to handle the numerical feature, the Pearson correlation coefficient has been used, which is characterized within Condition-1, and for categorical features, ANOVA-F has been used, which is described as:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{1}$$

After performing the feature selection, eighteen features were kept, including Location, MinTemp, MaxTemp, Rainfall, WindGustDir, WindGustSpeed, WindDir9am, WindSpeed9am, WindSpeed3pm, Humidity9am, Humidity3pm, Pressure9am, Pressure3pm, Temp9am, Temp3pm, RainToday, and Rain tomorrow. The numerical data are mostly skewed or nonstandard deviation in data analysis due to outliers, multi distributions, exceptionally exponential distributions, and more. This issue was solved by changing the numeric value into a categorical feature. To implement this method, a discretization process that changes over the numerical value into different distribution work has been applied as shown below:

$$F = \frac{n\sum_{i=1}^{n}\left(\overline{x_K} - \overline{x_G}\right)^2 / (K-1)}{\sqrt{\sum_{i=1}^{n}(x_K - x_G)^2} / (N-K)} \tag{2}$$

Feature scaling is one of the significant procedures required to standardize the independent features of the working dataset. There are different strategies for feature scaling such as min-max scaling, variance scaling, standardization, mean normalization, unit vectors, etc. In this work, min-max scaling has been used as a feature scaling method, and the exchange range has been set between 0 and 1. Mathematical formulla of the min-max scaling has been described as:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3}$$

CV is a resampling method to train, test, and validate the model using different data in each iteration. There are lots of CV methods to perform this operation. This paper uses a 10-fold CV technique where the whole dataset is separated into ten folds. Each section is used either for training, validation, or test set. After data preprocessing, there were 142193 rows as samples and 17 columns as features.

3.3. Algorithms

A few of the most suitable machine learning and deep learning-based models have been implemented to build the best model. Some of them are neighbor relationship-based, some are naive Bayes (NB) theory-based, some are based on SVM, and so on. Table 2 shows a summary of all the algorithms that have been used in this research. For the KNN, five neighbors have been considered to find the similarity. For MLP, the learning rate is 0.001 with two hidden layers of 56 units. For radius learning, the radius value was set to 0.1. The Gini index has been used to split decisions between the decision tree and the random forest tree.

Table 2 The summary of the algorithm used

Methodology	Algorithms	Methodology	Algorithms
Neighbors classifier	k-nearest neighbors	Discriminant analysis classifier	Linear discriminant analysis
	Radius neighbors classifier		Quadratic discriminant analysis

Table 2 The summary of the algorithm used (continued)

Methodology	Algorithms	Methodology	Algorithms
Neighbors classifier	Nearest centroid	Support vector machine classifiers	Linear SVC
Ensemble classifiers	AdaBoost classifier		Linear SVC
	Bagging classifier		Nu SVC
	Gradient boosting classifier	Linear model	Stochastic gradient descent classifier
	Hist gradient boosting classifier		Ridge classifier
	Random forest classifier		Ridge classifier CV
Naive Bayes classifiers	Bernoulli NB		Passive aggressive classifier
	Multinomial NB		Logistic regression CV
	Categorical NB		Logistic regression
	Complement NB		Perceptron
	Gaussian NB		Impact learning [16]
Semi-supervised classifiers	Label propagation	Gaussian process classifiers	Radial basis function
	Label spreading	Neural network	MLP classifier

4. Methodology

To complete the whole workflow, a total of four steps have been executed: data collection and preprocessing, training model using supervised learning methods, testing, and performance analysis. A popular and acceptable dataset from the Kaggle platform [11] has been adopted for this project. This dataset was split into three parts: the training part, the validation part, and the testing part. After gathering all raw data, the dataset goes through data preprocessing steps, which have been used to make the dataset outliers-free and more solid. These data preprocessing steps also help in increasing the performance of the models [17]. As a result, diverse preprocessing methods such as cleaning data, missing value checks, handling the categorical data, feature selection, and feature scaling have been applied. The machine learning models are made by training with the data preprocessed earlier. The training and testing methods of all those models are different. From all the training methods, 29 classifiers have been used so that the performance can be compared and the best suitable model can be selected. Most of the models listed in the table showed good performance, but some didn’t fit very well. The complete methodology of this proposed model is shown in Fig. 2.

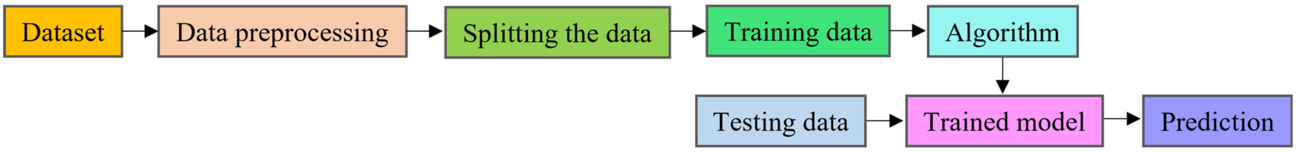

Fig. 2 The overview of the methodology of the proposed work

5. Experiments and Results

The model was built and then trained with the preprocessed dataset. In this section, the performances of all the algorithms have been compared. Besides, various experimental parameter has been tuned for performance analysis and evaluation. In addition, the experimental setup to accomplish the entire task has also been described. For this model, 11 statistical performance metrics have been considered for performance analysis and comparison.

The whole work has been completed in Google colab, and python has been provided as a simulation environment by Google. A machine learning framework named sci-kit learns and deep learning framework Keras have also been used to implement the classification algorithm. In addition, the Matplotlib library for data visualization, graphical representation, and data analysis has been used in this project.

Table 3 The summary of the algorithms that have been used in this research

Model	Accuracy	F1 score	Rs	PS	FBS	HL	JS	MC	AUC	BAC	CKS
Neighbors classifier											
KNC	0.868	0.752	0.672	0.749	0.809	0.202	0.604	0.468	0.749	0.689	0.411
NC	0.813	0.719	0.663	0.661	0.749	0.257	0.561	0.378	0.74	0.68	0.338
RNC	0.803	0.708	0.654	0.648	0.736	0.267	0.55	0.356	0.731	0.671	0.316
Ensemble classifiers											
ADC	0.893	0.802	0.72	0.791	0.856	0.177	0.656	0.559	0.797	0.737	0.507
BC	0.894	0.802	0.72	0.793	0.857	0.176	0.657	0.561	0.797	0.737	0.508
GBC	0.904	0.818	0.734	0.812	0.875	0.166	0.675	0.594	0.811	0.751	0.54
HGBC	0.91	0.833	0.751	0.817	0.885	0.16	0.692	0.619	0.828	0.768	0.569
RFC	0.907	0.823	0.737	0.819	0.881	0.163	0.68	0.604	0.814	0.754	0.549
Naive Bayes classifiers											
MNB	0.811	0.718	0.662	0.659	0.747	0.259	0.56	0.376	0.739	0.679	0.335
CoNB	0.733	0.686	0.679	0.63	0.712	0.337	0.512	0.36	0.756	0.696	0.297
CNB	0.731	0.686	0.683	0.632	0.713	0.339	0.512	0.365	0.76	0.7	0.299
GNB	0.695	0.66	0.669	0.618	0.692	0.375	0.482	0.336	0.746	0.686	0.263
CC	0.899	0.813	0.731	0.799	0.866	0.171	0.668	0.58	0.808	0.748	0.529
Semi-supervised classifiers											
LP	0.884	0.787	0.712	0.758	0.832	0.186	0.64	0.521	0.789	0.729	0.476
LS	0.902	0.816	0.733	0.808	0.872	0.168	0.673	0.589	0.81	0.75	0.536
Discriminant analysis classifier											
LDA	0.901	0.817	0.736	0.802	0.869	0.169	0.673	0.588	0.813	0.753	0.537
QDA	0.708	0.656	0.641	0.603	0.684	0.362	0.483	0.294	0.718	0.658	0.237
SVM classifiers											
LSVC	0.902	0.812	0.727	0.811	0.872	0.168	0.669	0.586	0.804	0.744	0.529
NuSVC	0.899	0.809	0.726	0.802	0.865	0.171	0.665	0.576	0.803	0.743	0.522
SGDC	0.894	0.795	0.709	0.8	0.857	0.176	0.65	0.555	0.786	0.726	0.496
RdC	0.9	0.802	0.714	0.813	0.867	0.17	0.658	0.572	0.791	0.731	0.511
RdCV	0.9	0.802	0.714	0.813	0.867	0.17	0.659	0.572	0.791	0.731	0.511
PAC	0.85	0.623	0.568	0.767	0.709	0.22	0.506	0.322	0.645	0.585	0.202
LRCV	0.9	0.815	0.733	0.801	0.868	0.17	0.671	0.584	0.81	0.75	0.533
LR	0.901	0.815	0.733	0.802	0.869	0.169	0.672	0.585	0.81	0.75	0.534
Pr	0.804	0.716	0.666	0.653	0.742	0.266	0.556	0.373	0.743	0.683	0.332
IL	0.902	0.813	0.727	0.811	0.872	0.168	0.669	0.586	0.804	0.744	0.53
Gaussian process classifiers											
GPC	0.903	0.818	0.736	0.807	0.873	0.167	0.675	0.592	0.813	0.753	0.54
Neural network classifier											
MLPC	0.906	0.833	0.757	0.805	0.879	0.164	0.691	0.613	0.834	0.774	0.568

Here, the twenty-nine most suitable machine learning models have been used to predict the rainfall possibility. A total of 11 statistical measurements have been considered and listed in Table 3. The table shows that the HGBC has predicted the best accuracy of 0.91, and the F1 score is 0.833. The random forest tree classifier has obtained the best accuracy of 0.907 with an F1 score is 0.823 from the branch of ensemble classifiers. The MLPC has acquired good accuracy, which is 0.906, along with an F1 score of 0.833 from the section of the NN classifier.

Moreover, from the section on the neighbor's classifier, it can be noticed that the KNC has shown the best accuracy of 0.868, and the F1 score is 0.752. Also, from the section on NB algorithms, CC has shown the best accuracy of 0.899 and its F1 score is 0.813 among all NB classifiers. After that, LS also has placed the best accuracy of 0.902, and its F1 score is 0.816 from the branch of semi-supervised classifiers. Besides, it can also be seen from the discriminant analysis section that the linear discriminant analysis has proved the best position of accuracy, 0.901, with an F1 score of 0.817.

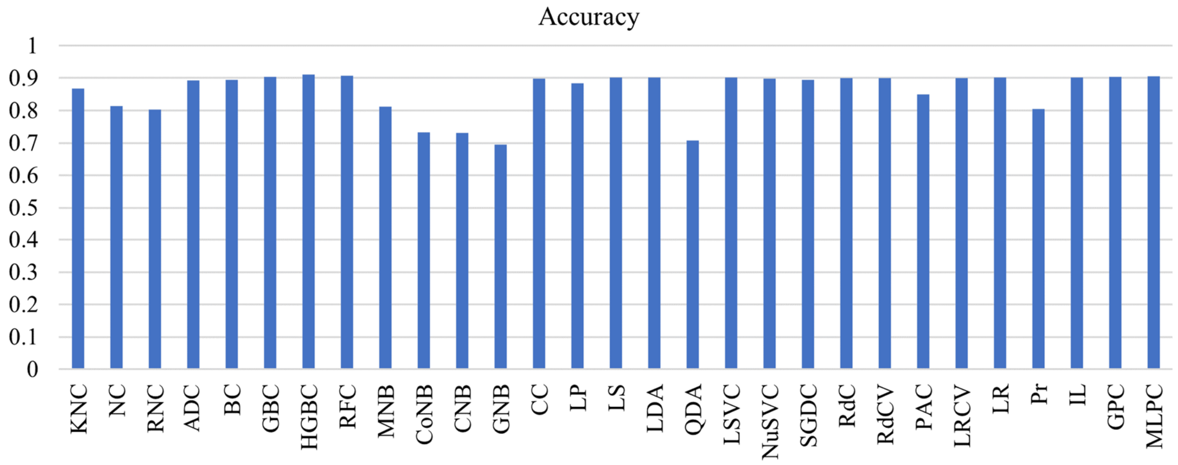

Fig. 3 Comparison of accuracy among the models

Next, LSVC also has the best accuracy of 0.902, and its F1 score is 0.812 from the branch of SVM classifiers. Finally, GPC has figured out the best accuracy result is 0.903 with an F1 score of 0.818 from the section of Gaussian process classifiers. All-inclusive, by analyzing all the sections of algorithms for the rain prediction model, it came out that the HGBC is the winner with an accuracy of 0.91, and the F1 score is 0.833. Fig. 3 shows the accuracy comparison among all the models. A ROC curve is a graph showing the performance of a classification model at all classification thresholds. Most of the algorithms show a high chance that the classifier will be able to distinguish the positive class values from the negative class values.

6. Conclusion and Future Work

In this work, a machine learning model has been presented that can determine whether it will rain or not on the very next day. Real data from Australia has been adopted from the Kaggle platform and implemented in the model. This data-driven model is more accurate than any other statistical or numerical-based model. The primary purpose here is to find the best classifiers for predicting rainfall. For this reason, various machine learning classifiers have been implemented. Eventually, the most significant performance metrics have been compared, including accuracy, F1 scores, ROC, AUC, and HGBC have shown the best accuracy with a good score. All the models used here have been trained based on the dataset of a particular zone. Therefore, rain prediction accuracy may vary depending on the dataset characteristics. Training this model with a dataset that has a sample collected from diverse places can make this model more general and suitable for all locations in this world.

This proposed model will be able to forecast the rain for the short-term, specifically for the next day. A new model can be built that will predict the rain for the long term in the future. Combining this two may build a complete solution for rain prediction. Moreover, this machine learning-based model may not be easily useable for general people. Therefore, mobile and computer apps can be built so those general people can easily use them. A deep learning and NN model approach can be used to improve the result. Undoubtedly, there is a plan to evaluate the other country's data for forecasting the rain using this model. Thus, this model is expected to be a universal and easy rain prediction tool for ordinary people.

Nomenclature

HGBC	Hist gradient boosting classifier	BC	Bagging classifier
ROC	Receiver operating characteristic	GBC	Gradient boosting classifier
AUC	Area under the curve	RFC	Random forest classifier
NN	Neural networks	MNB	Multinomial naive Bayes
SVM	Support vector machine	CoNB	Complement naive Bayes
PSO	Particle swarm optimization	CNB	Categorical naive Bayes classifier
MLP	Multi-layer perceptron	GNB	Gaussian naive Bayes
mm	Measured in millimeters	CC	Calibration classifier
CV	Cross-validation	LP	Label propagation
KNN	K-nearest neighbors	LS	Label spreading
NaN	Not-a-number	LDA	Linear discriminant analysis
NB	Naive Bayes	QDA	Quadratic discriminant analysis
Rs	Recall score	LSVC	Linear support vector classifier
PS	Precision score	NuSVC	Nu support vector classification
FBS	F-beta score	SGDC	Stochastic gradient descent classifier
HL	Hamming loss	RdC	Ridge classifier
JS	Jaccard score	RdCV	Ridge classifier CV
MC	Matthew's correlation	PAC	Passive aggressive classifier
BAC	Balanced accuracy	LRCV	Logistic regression CV
CKS	Cohen's kappa	LR	Logistic regression
KNC	K neighbors' classifier	Pr	Perceptron classifier
NC	Nearest centroid	IL	Impact learning
RNC	Radius neighbor's classifier	GPC	Gaussian process classifier
ADC	AdaBoost classifier	MLPC	Multi-layer perceptron classifier

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] K. T. Sohn, J. H. Lee, and S. H. Lee, "Statistical Prediction of Heavy Rain in South Korea," Advances in Atmospheric Sciences, vol. 22, no. 5, pp. 703-710, 2005.

[2] T. Denœux and P. Rizand, "Analysis of Radar Images for Rainfall Forecasting Using Neural Networks," Neural Computing and Applications, vol. 3, no. 1, pp. 50-61, March 1995.

[3] B. K. Shah, S. Thapa, R. S. Diyali, S. Hk, and S. Maharjan, "Rain Prediction Using Polynomial Regression for the Field of Agriculture Prediction for Karnatakka," International Journal of Advances in Engineering and Management, vol. 2, no. 3, pp. 62-66, March 2020.

[4] P. Asha, A. Jesudoss, S. Prince Mary, K. V. Sai Sandeep, and K. Harsha Vardha, "An Efficient Hybrid Machine Learning Classifier for Rainfall Prediction," Journal of Physics: Conference Series, vol. 1770, no. 1, article no. 012012, March 2021.

[5] S. Sakthivel and G. Thailambal, "Effective Procedure to Predict Rainfall Conditions Using Hybrid Machine Learning Strategies," Turkish Journal of Computer and Mathematics Education, vol. 12, no. 6, pp. 209-216, April 2021.

[6] D. Naidu, B. Majhi, and S. K. Chandniha, “Development of Rainfall Prediction Models Using Machine Learning Approaches for Different Agro-Climatic Zones,” Handbook of Research on Automated Feature Engineering and Advanced Applications in Data Science, IGI Global, 2021.

[7] T. V. Dinh, H. Nguyen, X. L. Tran, and N. D. Hoang, “Predicting Rainfall-Induced Soil Erosion Based on a Hybridization of Adaptive Differential Evolution and Support Vector Machine Classification,” Mathematical Problems in Engineering, vol. 2021, article no. 6647829, 2021.

[8] H. Abdel-Kader, M. Abd-El Salam, and M. Mohamed, “Hybrid Machine Learning Model for Rainfall Forecasting,” Journal of Intelligent Systems and Internet of Things, vol. 1, no. 1, pp. 5-12, 2021.

[9] N. Samsiahsani, I. Shlash, M. Hassan, A. Hadi, and M. Aliff, “Enhancing Malaysia Rainfall Prediction Using Classification Techniques,” Journal of Applied Environmental and Biological Sciences, vol. 7, no. 2S, pp. 20-29, April 2017.

[10] K. C. Luk, J. E. Ball, and A. Sharma, “An Application of Artificial Neural Networks for Rainfall Forecasting,” Mathematical and Computer Modeling, vol. 33, no. 6-7, pp. 683-693, March 2001.

[11] J. Abbot and J. Marohasy, “Application of Artificial Neural Networks to Rainfall Forecasting in Queensland, Australia,” Advances in Atmospheric Sciences, vol. 29, no. 4, pp. 717-730, June 2012.

[12] C. Shah, C. Hendahewa, and R. Gonzalez-Ibanez, “Rain or Shine? Forecasting Search Process Performance in Exploratory Search Tasks,” Journal of the Association for Information Science and Technology, vol. 67, no. 7, pp. 1607-1623, July 2016.

[13] M. Sangiorgio, S. Barindelli, R. Biondi, E. Solazzo, E. Realini, G. Venuti, et al., “Improved Extreme Rainfall Events Forecasting Using Neural Networks and Water Vapor Measures,” 6th International Conference on Time Series and Forecasting, pp. 820-826, September 2019.

[14] D. Han, T. Kwong, and S. Li, “Uncertainties in Real-Time Flood Forecasting with Neural Networks,” Hydrological Processes: An International Journal, vol. 21, no. 2, pp. 223-228, January 2007.

[15] J. Young, “Rain in Australia,” https://www.kaggle.com/jsphyg/weather-dataset-rattle-package, October 30, 2007.

[16] M. Kowsher, A. Tahabilder, and S. A. Murad, “Impact-Learning: A Robust Machine Learning Algorithm,” Proceedings of the 8th International Conference on Computer and Communications Management, pp. 9-13, July 2020.

[17] C. V. Z. Zelaya, “Towards Explaining the Effects of Data Preprocessing on Machine Learning,” IEEE 35th International Conference on Data Engineering, pp. 2086-2090, April 2019.