A Comprehensive Study on Different Machine Learning Techniques for Rainfall Forecasting

1st Koushik Roy
MSCSE
United International University
kroy2230054@mscse.uiu.ac.bd

***Abstract*—Predicting rainfall is one of the difficult and unpredictable jobs that has a major influence on human society. Accurate and timely prediction of future rainfall will surely aid in minimizing human and financial loss. By extracting and combining hidden information from the linear and non-linear patterns of historical meteorological data, machine learning techniques can utilize computational approaches to predict rainfall. Although there are a variety of techniques and approaches for predicting rain, there is still a scarcity of reliable findings. This study includes a series of experiments that include the use of machine learning techniques to construct models that predict whether or not it will rain tomorrow based on meteorological data in major Australian cities. The findings provide a comparison of several evaluation criteria for these machine learning approaches, as well as their effectiveness in predicting rainfall using weather data.**

***Index Terms*—Rainfall Prediction, Classification, Australia, Machine Learning, Weather Forecast**

I. Introduction

Rainfall prediction is a challenging task yet crucial for an agriculture-based country. A country's economy significantly depends on the proper operation of the farming sector, and farming is dependent on rainfall [1]. Thus, the governments, industries, risk management institutions, and the scientific communities have shown major concern in figuring out a more accurate and reliable prediction model for rainfall [2]. Rainfall is a climatic element that has an impact on a variety of human activities, including agriculture, building, power generation, forestry, and tourism [3]. Rainfall forecasting is critical in this regard since it is the variable that has the strongest link to natural disasters including landslides, flooding, mass movements, and avalanches. These occurrences have had a long-term impact on society [4]. As a result, having a good method for predicting rainfall enables appropriate agencies to take preventative and mitigation actions against these natural occurrences [5].

We employed a variety of machine learning techniques and models to produce accurate and timely predictions in order to resolve the ambiguity associated with rainfall prediction. This study aims to cover the entire machine learning life cycle, from data preparation through model implementation and evaluation. Imputing missing values, feature transformation, encoding categorical features, feature scaling, and feature selection are all phases in the data preprocessing process. Models used to serve our purpose includes Fully Connected Dense Neural Net, Logistic Regression, Decision Tree, K-Nearest Neighbor, etc. Accuracy, Precision, Recall, F1-Score, and Prediction Time were used to assess the results. We used Australian weather data from several weather stations in Australia to train our classifiers for our experiments.

The following sections of this paper are organized as follows. In Section 2, we presented the relevant findings of related literature. In Section 3, we describe the dataset that has been analyzed in this work. Then the methodology of this research work has been discussed in Section 4. In Section 5, the experimental results have been discussed. Finally, In section 6, the conclusion along with future work has been presented.

II. Related Works

To avert calamities, an action must be initiated that anticipates future rainfall instability behavior. Two techniques are often used to predict rainfall; one is to examine large volumes of data acquired over time and the other is to create equations for predicting future rainfalls [1].

Haidar et al. [6] suggested a unique method for forecasting rainfall. The process was split into two halves. To reduce the collection of characteristics and examine the potential segments for rain forecasting, the biased forward classification approach is employed. In [7] researchers presented that rainfall expectation refers to the assurance of precipitation patterns for a given location. It is vital for the horticulture sector as well as other industries.

The data mining approach aids in the uncovering of hidden patterns, resulting in more accurate rainfall forecasting [8]. Data mining, in contrast to linear regression, which offers approximate values, provides generalized values. This strategy's sole flaw is that it doesn't function well with long-term forecasting [9]. Although a variety of data mining techniques are available, it is critical to select one that is appropriate for the area to which it is applied. In [10], researchers looked at several data mining approaches and came to the conclusion that k-means and decision tree algorithms outperform other weather prediction approaches.

There are three sections to the hybrid classifier such as the simulation section, training section, and testing section. Data analysis is the focus of the simulation technique. The optimal

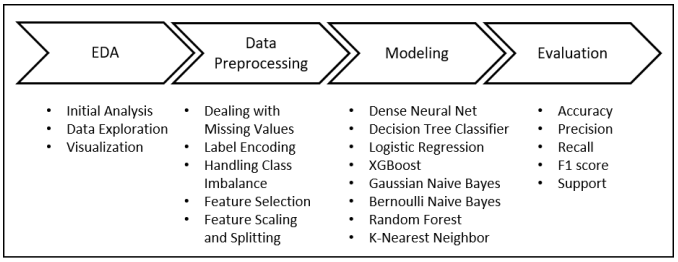


Fig. 1. Overall system flow diagram representing the components of the major processes

algorithm to employ is determined in the training stage, and the dataset is assessed to produce the anticipated outcome [11]. In Sudan, the SARIMA model has been put to the test. The most major drawback of this approach is that it is limited to a smaller area [12]. For a monthly average rain forecast of 21 Assam stations in India, the Bayesian network technique was used [13].

III. Dataset Description

The dataset that has been analyzed in this study has been originated from daily weather observations in several locations across Australia. It contains about 10 years of observational data. The dataset has been hosted in Kaggle [14]. Observations were drawn from numerous weather stations such as the Climate Data Online [15], Canberra- Australian Capital Territory [16], Daily Weather Observations [17]. Definitions adapted from here [18]. There are 23 features and around 142k instances in the data set. The features include observational weather and atmospheric data in various points of time in the 24 hour time duration for each day, which is represented in each row. The target variable to predict is the rain tomorrow.

IV. Methodology

To attain some useful insights from the Australian weather dataset, we are going to follow the entire life cycle of a data science project. It starts with loading the dataset and exploratory data analysis(EDA), followed by data preprocessing, modeling, and evaluation. Each stage consists of multiple steps and finishing all of these steps will provide us with crucial information such as how capable each of the models in generalizing the features of the dataset and making an accurate prediction of the likelihood of rainfall tomorrow. The overall system components are presented in Fig. 1. The following subsections will describe these four major processes in a detailed manner.

A. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is the critical process of doing exploratory investigations on data utilizing summary statistics and graphical representations in order to find patterns, uncover anomalies, test hypotheses, and validate assumptions. EDA is useful for machine learning challenges because it ensures that future findings will be accurate, appropriately understood, and applicable to the business settings. Only when raw data has been verified and examined for abnormalities, guaranteeing that the data set was obtained without mistakes, can such a degree of confidence be attained. EDA also aids in the discovery of ideas that would otherwise go unnoticed or uninvestigated by corporate stakeholders and researchers.

The approach taken for EDA can be grouped into the following categories:

- Checking for missing values
- Dealing with class imbalance in the target variable
- Exploring the correlation matrix

The dataset was processed using the Pandas package in Python, which provides several useful functions for the initial analysis and dataset health check. The info function of the data frame object has been called to check the initial useful information about the dataset. Then the null values of each column have been counted and printed. From the initial look, there have been multiple features with a significant amount of missing values can be found. These data features have been filled with imputer functions in the preprocessing stage. A list of missing values before and after the imputation has been presented in Table I.

Then describe function was applied in the input data frame to get a broader level understanding of each of the columns in the dataset. In this regard, the class imbalance in the target variable, rain tomorrow, is of significant consideration.

Finally, the correlation between each pair of columns has been calculated. For our prediction purpose, we will only be interested in the correlation of the target feature, rain tomorrow, with the other features. These correlation value has been sorted and the top 10 values have been presented in the bar chart in Fig. 2. In this case, the absolute values of the correlation have been used. The sign of the correlation between a pair of variables can either be positive or negative. Here, positive refers to the case where increasing the value of one variable will result in the increase of another variable and vice versa. By taking the absolute values of the correlation, we ignore the sign and consider the relative association. These sorted correlations have been used to create a heatmap of the corresponding pair of variables, which is presented in Fig. 3.

B. Data Preprocessing

Data preprocessing is the process of transforming unstructured data into well-formed data sets that may be used by data mining techniques. When it comes to raw data, it is common for it to be incomplete and structured incorrectly. Data validation and data imputation are both parts of the preprocessing process. The purpose of data validation is to determine if the data is full and correct. The objective of data imputation is to manually or automatically repair mistakes and fill in missing numbers. Machine learning (ML) processes require data preparation to guarantee that large datasets are organized in such a way that the data they contain can be read and digested by learning algorithms.

1) Missing Values: We discovered a lot of instances with null values as part of our EDA phase. As a result, this is a critical step to do. To deal with the missing data, we will use

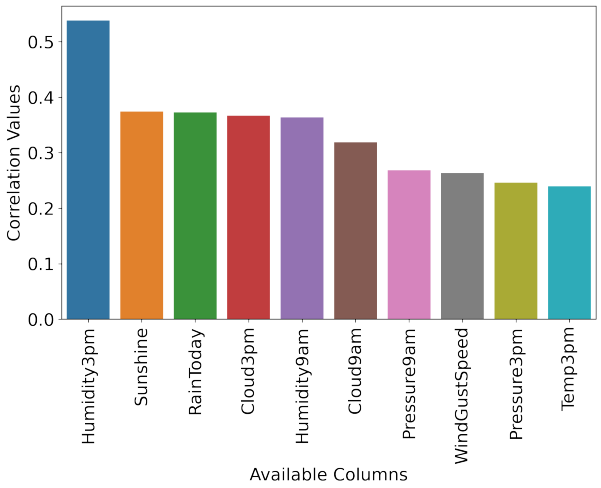


Fig. 2. Bar chart of the top 10 correlation values with Target Column(RainTomorrow) in Descending Order

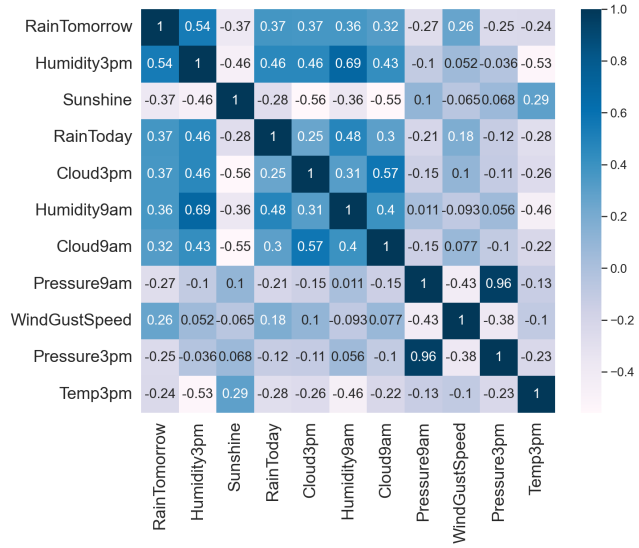


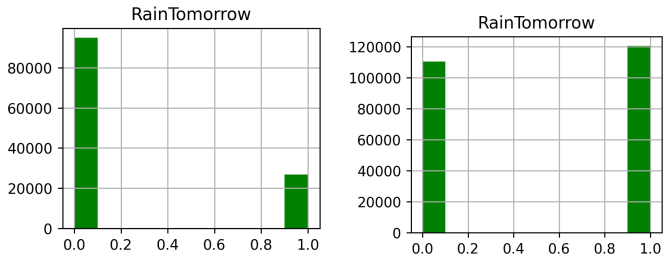
Fig. 3. Heatmap of correlation of the top 10 features

Simple Imputer from Scikit learn. The mean or the median values will replace the missing values for different features based on their individual characteristics. A comparison of missing values before and after the imputation process has been presented in the Table I. After dealing with most of the missing values the remaining values will be dropped.

2) Label Encoding: We generally work with datasets in machine learning that include numerous labels in one or more columns. Labels in the form of words or numbers can be used. The training data is frequently labeled in words to make it comprehensible or human-readable. Label encoding is the process of converting labels to a numeric format so that machines can read them. Then, using machine learning algorithms, smarter judgments may be made about how those labels should be used. It is a key preprocessing step for the structured dataset in supervised learning. In our dataset, the

TABLE I
NUMBER OF MISSING VALUES BEFORE AND AFTER THE IMPUTATION PROCESS

Feature	Before	After	Feature	Before	After
Location	0	0	MinTemp	1485	1485
MaxTemp	1261	1261	Rainfall	3261	3261
Evaporation	62790	0	Sunshine	69835	0
WindGustDir	10326	10326	WindGustSpeed	10263	10263
WindDir9am	10566	10566	WindDir3pm	4228	4228
WindSpeed9am	1767	1767	WindSpeed3pm	3062	3062
Humidity9am	2654	2654	Humidity3pm	4507	4507
Pressure9am	15065	0	Pressure3pm	15028	0
Cloud9am	55888	0	Cloud3pm	59358	0
Temp9am	1767	1767	Temp3pm	3609	3609
RainToday	3261	3261	RainTomorrow	3267	3267



(a) Before Oversampling Applied (b) After Oversampling Applied

Fig. 4. Dealing with class imbalance using oversampling method

features with object datatype have been label encoded using a label encoder from Scikit learn.

3) Handling Class Imbalance: In our EDA step, we learned that our dataset is highly imbalanced. Having imbalanced data is not particularly useful for machine learning algorithms as it might introduce biasing issues. Oversampling was performed to generate synthetic similar instances of the minority class thus minimizing the overall class imbalance. A histogram of the target variable before and after performing oversampling is presented in Fig. 4.

4) Feature Selection: Feature selection refers to selecting the features that contribute more towards predicting the target variable. Irrelevant features can result in inaccurate prediction and make the model learn based on not strongly correlated features with the output. We experimented with univariate selection and correlation matrix with heatmap and both provide us with similar output. The SelectKBest class in the Scikit learn package may be used with a variety of statistical tests to pick a particular number of features. We chose 15 of the top features from our data set using the chi-squared statistical test for non-negative characteristics.

5) Feature Scaling and Splitting: When the range and magnitude of the features of a dataset are highly varying, it is common to perform feature scaling. Since most machine learning algorithms use Euclidian distance, the varying magnitude of different features can restrict the performance of the model. To minimize this issue, we need to transform all the features into the same level of magnitude. This is referred to

as feature scaling. The min-max scalar from Scikit learn was used to carry out feature scaling and bring all the features in the range 0 to 1.

Then we need to split the dataset into two parts namely training set and testing set. In this case, 75% of data was used for training and the other 25% of data was reserved for testing.

C. Modeling

Our problem is essentially a binary classification problem. Consequently, we need to build a classification model that can predict the binary class output, which is to predict whether there will be rain tomorrow or not. Different types of models were used for rainfall prediction, each belonging to the different model family. All these different models were implemented using Scikit learn and TensorFlow. The following models have been used to build the rainfall prediction mechanism.

1) Dense Neural Net: The term Dense Neural Net (DNN) refers to a network layer in which all neurons are completely connected (dense). Each neuron in a layer receives input from every neuron in the layer before it. As a result, they are extremely intertwined. A highly connected layer provides learning capabilities based on all conceivable combinations of the properties of the preceding layer.

2) Decision Tree Classifier: Because Decision Trees have a natural if-then-else structure, they are simple to include into a programming framework. They're also useful for categorization problems involving several evaluations of characteristics or attributes to arrive at a final category. The input and output variables might be categorical or continuous. The Decision Tree's characteristics make it a good fit for our issues because our target variable is a binary categorical variable.

3) Logistic Regression: Given a collection of independent variables, logistic regression is a classification technique used to predict a binary result. Dummy variables are used to represent binary or categorical outcomes. When the outcome variable is categorical, we may conceive of logistic regression as a particular instance of linear regression with the log of changes as the dependent variable. In basic terms, it fits data to a logit function to forecast the probability of an event occurring. As a result, because ours is a binary classification problem, Logistic Regression is a better fit.

4) XGBoost: eXtreme Gradient Boosting is what XGBoost stands for. It is a high-speed and high-performance implementation of gradient boosted decision trees. The implementation of the method was created with the goal of maximizing speed and performance.

5) Gaussian Naive Bayes: The Bayes Theorem is used to build the Naive Bayes classifiers, which are a set of classification algorithms based on the Bayes Theorem. It's a group of algorithms that all work on the same premise: each pair of classified features is independent of the others. In Gaussian Naive Bayes, the continuous values associated with each feature are anticipated to be distributed according to a Gaussian distribution.

6) Bernoulli Naive Bayes: Naive Bayes classifiers have shown to be effective in a wide range of real-world applications, including document classification and spam filtering. They just need a little amount of training data to predict the needed parameters. Naive Bayes classifiers may be very fast when compared to more sophisticated methods. Because the class conditional feature distributions are separated, each one-dimensional distribution may be estimated individually. This, in turn, assists in the reduction of dimensionality-related problems.

7) Random Forest: A random forest is a machine learning approach for solving classification and regression tasks. It makes use of ensemble learning, which is a technique for solving difficult problems by combining many classifiers.

8) K-Nearest Neighbor: K-Nearest Neighbor(KNN) is a slow and non-parametric learning method. The dataset determines the model structure. It does not require any training data points. In the testing phase, all training data was used. With a smaller number of features, KNN performs better than with a high number of features. The number of neighbors(K) in KNN is a hyperparameter that must be chosen during model construction. There is no ideal value for K. Each dataset has its own set of specifications. We experimented with different values for K and recorded its impact on the model's overall accuracy and prediction time.

D. Evaluation

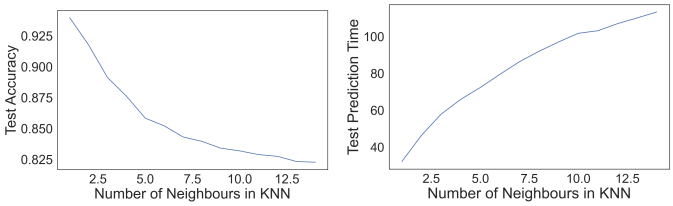
1) Accuracy: Accuracy is defined as the number of correct predictions divided by the total number of input samples. If each class has an equal number of samples, it performs well. With the use of the upsampling technique, we were able to eliminate a class imbalance in our initial dataset.

2) Precision: Precision is a statistic that measures how many correct positive forecasts have been made. As a result, precision estimates the accuracy of the minority class. The ratio of properly predicted positive instances divided by the total number of positive examples anticipated is used to compute it.

3) Recall: The recall of a classifier, which is a measure of its completeness, measures its ability to correctly discover all positive instances. The ratio of true positives to the sum of true positives and false negatives for each class is used to compute it.

4) F1 score: With 1.0 being the greatest and 0.0 being the lowest, the F1 score is a weighted harmonic mean of accuracy and recall. Because precision and recall are factored into F1 ratings, they are on average lower than accuracy assessments. We use the weighted average of F1 rather than global accuracy to compare classifier models.

5) Support: Support is the number of real instances of the class in the provided dataset. Imbalanced support in the training data might indicate that the classifier's reported scores have fundamental problems, needing stratified sampling or rebalancing. Support is not model-dependent; instead, it diagnoses the assessment process.



(a) Accuracy vs Neighbors(k) (b) Prediction time vs Neighbors(k)

Fig. 5. K-Nearest Neighbor model accuracy and prediction time vs Neighbors(k)

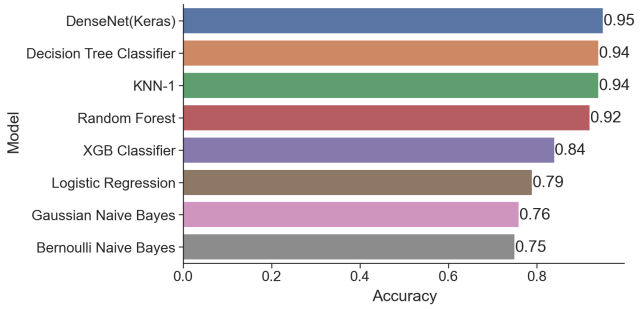


Fig. 6. All the experimented models' accuracy in descending order

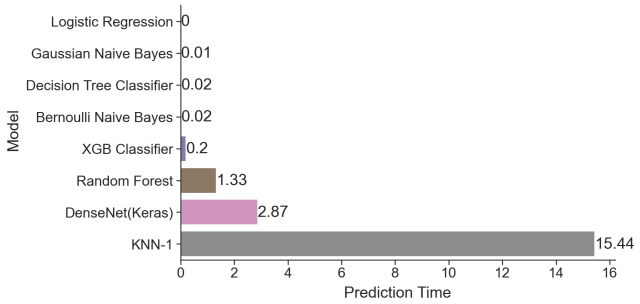


Fig. 7. All the experimented models' prediction times(in seconds) in ascending order

V. Experimental Results and Discussion

In the modeling subsection, we described the 8 models we experimented with. We collected the necessary values of the evaluation parameters in order to compare these models on the merit of their accuracy, prediction time, precision, etc. The overall results are presented in Table II. From the table, we can see that Logistic Regression has got a lead in terms of prediction time(1.8ms) with not so good accuracy of 79%. When it comes to accuracy, DenseNet(Keras) and KNN-1 have the highest value, which is 95% and 94%. Precision for class 0(No Rain), Decision Tree Classifier and KNN-1 got the lead with value 97%. For class 1(Rain) of precision, DenseNet(Keras) takes the first spot with the value of 96%. It continued to be the best in terms of Recall(class 0) with 96% and F1 Score(both classes) with the value of 95%. Finally, in Recall(class 1), the best value is held by Decision Tree Classifier and KNN-1 in 97%.

In the case of K-Nearest Neighbor(KNN), we experimented with different values of the number of neighbors(K) and checked the impact on the accuracy and prediction time. The recorded value of the accuracy and prediction time has been plotted against the number of neighbors(K), the range to which is between 0 and 15 exclusive. The plot of accuracy vs K is shown in Fig. 5a and the plot of prediction time vs K has been presented in Fig. 5b. From these two figures, it is clear that increasing the value of K has severely decreased the accuracy (from 94% to 82%) and also increased the prediction time by a significant margin(15 seconds to 120 seconds). The best KNN model, where the accuracy is the highest and prediction time is the lowest occurs when K=1.

One of the most important evaluation criteria is the accuracy of the model. So, we sorted all of our models based on accuracy and made a bar chart shown in Fig. 6. In the figure, it is clear that all of our models performed fairly well and there is not much of a performance gap between our highest performing model and the lowest. DenseNet(Keras) and KNN-1 have taken the lead with 94% accuracy and the least accurate model was Bernoulli Naive Bayes with 75% accuracy.

The prediction time is a fairly accurate estimator for the computational complexity of a model. Thus we also take into account the time each of our models took to predict the full test set. Here the prediction time(in seconds) has been sorted in ascending order and made into a bar chart shown in Fig. 7. We can see from the figure that most of our models are very efficient when it comes to the complexity of the model. Logistic Regression, Gaussian Naive Bayes, Decision Tree, Bernoulli Naive Bayes, XGB classifier barely took any time, ranges between 1.8 ms to 200 ms. The worst-performing model is the KNN, which is not a good choice if the dataset is large in quantity. The KNN model performs worse with each additional training data or the increase in the number of neighbors(K). Any of the other models can be used in places where the robustness and computational complexity is an important factor.

We experimented with 8 models and when it comes to the best overall model we need to choose the model which optimizes both the accuracy and the prediction time. So, we plotted the models in a scatter plot, shown in Fig. 8, where the x-axis represents the accuracy whereas the y-axis represents the Prediction Time. The models reside in the right half of the graph represent higher accuracy, and the model resides in the bottom portion of the graph represents lower prediction time. A few of the models fit our criteria to become efficient enough to provide good accuracy in low prediction time. These models are Random Forest, Decision Tree Classifier, and DenseNet(Keras).

VI. Conclusion and Future Work

We investigated and implemented numerous preprocessing procedures in this study, and discovered how they affected the overall performance of our classifiers. We also experimented with a variety of classification models and collected the necessary evaluation metrics to figure out the strengths and

TABLE II
EVALUATION MATRICES OF ALL THE MODELS

Model	Pred. Time	Accuracy	Precision	Recall	F1score
DenseNet(Keras)	2.873	0.95	0.93	0.96	0.95
Decision Tree Classifier	0.022	0.93	0.97	0.89	0.93
Logistic Regression	0.001	0.79	0.78	0.79	0.79
XGB Classifier	0.201	0.79	0.79	0.78	0.79
Gaussian Naive Bayes	0.012	0.76	0.74	0.77	0.75
Bernoulli Naive Bayes	0.022	0.75	0.74	0.74	0.74
Random Forest	1.332	0.91	0.92	0.90	0.91
KNN-1	15.436	0.94	0.97	0.90	0.94

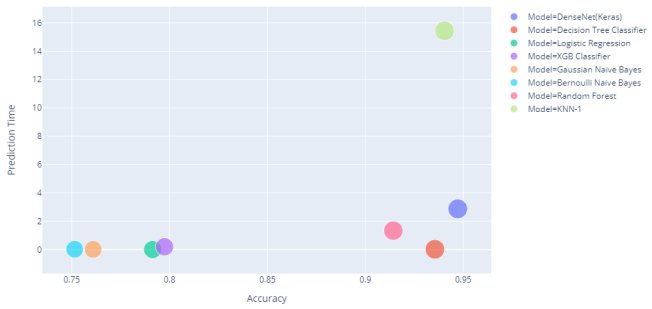


Fig. 8. Prediction time vs accuracy visualized using a scatter plot

weaknesses of each of them. We may infer that Australian weather is unpredictable, and there is not a strong relation between rainfall and location or time. We discovered certain patterns and correlations in the data, which aided in the identification of key characteristics and helped us accurately predict the next day's rainfall. For future work, time series analysis in this dataset may become fruitful in discovering patterns of climate change and potentially help combat global warming.

REFERENCES

[1] M. Gowtham Sethupathi, Y. S. Ganesh, and M. M. Ali, "Efficient rainfall prediction and analysis using machine learning techniques," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 6, pp. 3467–3474, 2021.

[2] N. Oswal, "Predicting rainfall using machine learning techniques," *arXiv preprint arXiv:1910.13827*, 2019.

[3] W. H. Organization, *Guidelines for safe recreational water environments: Coastal and fresh waters*. World Health Organization, 2003, vol. 1.

[4] I. Alcántara-Ayala, "Geomorphology, natural hazards, vulnerability and prevention of natural disasters in developing countries," *Geomorphology*, vol. 47, no. 2-4, pp. 107–124, 2002.

[5] N. Nicholls, "Atmospheric and climatic hazards: Improved monitoring and prediction for disaster mitigation," *Natural Hazards*, vol. 23, no. 2, pp. 137–155, 2001.

[6] A. Haidar and B. Verma, "Monthly rainfall forecasting using one-dimensional deep convolutional neural network," *IEEE Access*, vol. 6, pp. 69 053–69 063, 2018.

[7] S. Chatterjee, B. Datta, S. Sen, N. Dey, and N. C. Debnath, "Rainfall prediction using hybrid neural network approach," in *2018 2nd International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom)*. IEEE, 2018, pp. 67–72.

[8] J. Joseph and T. Ratheesh, "Rainfall prediction using data mining techniques," *International Journal of Computer Applications*, vol. 83, no. 8, 2013.

[9] M. Navid and N. Niloy, "Multiple linear regressions for predicting rainfall for bangladesh," *Communications*, vol. 6, no. 1, p. 1, 2018.

[10] D. Chauhan and J. Thakur, "Data mining techniques for weather prediction: A review," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, no. 8, pp. 2184–2189, 2014.

[11] P. Asha, A. Jesudoss, S. P. Mary, K. S. Sandeep, and K. H. Vardhan, "An efficient hybrid machine learning classifier for rainfall prediction," in *Journal of Physics: Conference Series*, vol. 1770, no. 1. IOP Publishing, 2021, p. 012012.

[12] E. H. Etuk and T. M. Mohamed, "Time series analysis of monthly rainfall data for the gadaref rainfall station, sudan, by sarima methods," *International Journal of Scientific Research in Knowledge*, vol. 2, no. 7, p. 320, 2014.

[13] A. Sharma and M. K. Goyal, "Bayesian network model for monthly rainfall forecast," in *2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*. IEEE, 2015, pp. 241–246.

[14] "Rain in australia predict next-day rain in australia," https://www.kaggle.com/jsphyg/weather-dataset-rattle-package, accessed: 2021-07-30 [Online].

[15] "Climate data online," http://www.bom.gov.au/climate/data/, accessed: 2021-08-07 [Online].

[16] "Canberra, australian capital territory," http://www.bom.gov.au/climate/dwo/IDCJDW2801.latest.shtml, accessed: 2021-08-07 [Online].

[17] "Daily weather observations," http://www.bom.gov.au/climate/dwo/, accessed: 2021-08-07 [Online].

[18] "Notes to accompany daily weather observations," http://www.bom.gov.au/climate/dwo/IDCJDW0000.shtml, accessed: 2021-08-07 [Online].