Application and Analysis of Machine Learning Based Rainfall Prediction

Hongyi Duan [1,a]
[1]Xi'an Jiaotong University
Faculty of Electronic and Information Engineering
Xi'an, China
[a]dann_hiroki_china@ieee.org

Yuchen Li [1,b]
[1]Xi'an Jiaotong University
Faculty of Electronic and Information Engineering
Xi'an, China
[b]yuchenli@ieee.org

Qingyang Li [1,c]
[1]Xi'an Jiaotong University
Faculty of Electronic and Information Engineering
Xi'an, China
[c]likon2101@ieee.org

Yiyi Wang [1,d]
[1]Xi'an Jiaotong University
Faculty of Electronic and Information Engineering
Xi'an, China
[d]1025658586@qq.com

Yuming Xie [2,e*]
[2]National University of Defense Technology
Faculty of Electronic Science
Changsha, China
Corresponding author: [e*]1392640393@qq.com

Haohui Peng [3,f]
[3]Nanjing University
Astronomy and Space Science School
Nanjing, China
[f]1352894469@qq.com

Abstract—Rainfall predicting is closely related to people's lives, and improving the accuracy of rainfall prediction is of great importance for people's life and scientific researches. In this paper we attempt to use several machine learning algorithms to analyse and predict rainfall from over 140,000 data recorded at 49 weather stations in Australia. The specific research in this paper is as follows.

(1) Explain the limitations of using traditional empirical/physical statistical methods in rainfall predicting through reviewing, and explain the advantages and remaining limitations of current machine learning algorithms applied to rainfall predicting over traditional methods.

(2) Different methods are applied to pre-process the data set for both typed and continuous variables to ensure the integrity of the information and to avoid the loss of excessive information in the data.

(3)The K-nearest neighbour algorithm, random forest algorithm and support vector machine algorithm are used to model the data for prediction. By comparing the prediction results of each model, summarize the advantages and disadvantages of each model.

The prediction results show that the prediction accuracy of the above three algorithms are 82.4464%, 85.6416% and 81.3915% respectively, and the F1-score values are 52.2388%, 61.8736% and 65.1689% respectively. Results of each algorithm are similar in accuracy, and all shows good performance in both accuracy and F1-score.

Keywords-Rainfall prediction; K-nearest neighbours; Random forests; Support vector machines; Machine learning.

I. Background and significance of this Research

Rainfall prediction plays an important role in many aspects of our society, especially in agricultural and industrial production.

Currently the main methods of rainfall predicting are empirical and physical statistical methods. The empirical statistical method is to collate and analyse the data and draw empirical conclusions from a large amount of data. However, when discerning similar weather, a certain amount of subjective decisions of the parties involved are intermingled, which may also result in significant errors in the predict results. Therefore, purely empirical statistical methods cannot be used as the main means of weather prediction[1]. The physical-statistical approach refers to the prediction of rainfall by means of relevant interaction patterns using a variety of climate dynamics. This method is often combined with empirical statistical methods and involves an exhaustive analysis of the data. This method is logical and predictions are more accurate, but the results are not as good for areas with complex climates[2].

Currently, machine learning has a wide range of applications, and the application of machine learning methods to weather predicting has been explored by a large number of scholars in this area, using the powerful non-linear learning capabilities and methods of machine learning to accurately predict seasonal-scale rainfall, which performs much better than traditional predicting results. There are many machine learning algorithms, including Plain Bayesian, Random Forest, and Artificial Neural Network (ANN). In this paper will attempt to use three machine learning algorithms to predict rainfall. KNN (K-Nearest Neighbor), Random Forest and Support Vector Machine (SVM).

979-8-3503-0245-5/23/$31.00 ©2023 IEEE

Our data is chosen from weather stations in different regions of Australia. As a result we visualise the experimental data, compare the accuracy rates derived from various algorithms, and analyse them to illustrate the relationship between rainfall and influencing factors, and finally select the more accurate rainfall prediction models. Our research will help to ensure agricultural and industrial production and provide a more theoretical basis for disaster prevention.

II. Basic Theory and Related Techniques

A. Nearest Neighbour Algorithm

1) Introduction of K-Nearest Neighbor algorithm

The K-Nearest Neighbor (KNN) algorithm is a relative simple algorithms in data classification , originally proposed by Cover and Hart in 1968. According to KNN, the category of the test data is inferred from the neighbours that are close to the training data. The training data is represented as a vector and the KNN classifies the data by measuring the distance between the different vector values.

2) Implementation of the K-nearest neighbour algorithm

If we have the input data

$$T=(x_1,y_1),(x_2,y_2)......(x_n,y_n)$$

where $x_i \in x \subseteq R^n$ is the feature vector of the instance, $y_i \in Y$ is the category of the instance and $i=1,2,...,N$. The feature vector is specified as X , and the category to which the instance belongs is specified as y.We establish the model using existing meteorological data to predict the tomorrow rainfall probability by the distance between features. The basic steps of the algorithm are:

(1) Input the training set data X and labels y , and input the test data.

(2) Calculate the distance between the test data and each training data, the Euclidean distance is calculated by the formula.

$$d(x,y)=\sqrt{\sum_{k=1}^{n}(x_k-y_k)^2}$$

(3) Select K numbers of nearest points according to the increasing distance.

(4) Determine the frequency of occurrence of the category in which the first K points are located, and return the category with the highest frequency of occurrence among the points as the predicted classification of the test data[3].

B Random Forest Algorithm

1) Introduction to Random Forests

Random Forest belongs to Ensemble Learning. The basic idea of Ensemble Learning is to develop multiple estimators through training, and when predicting, the final output depend on a combiner which take the results of the multiple estimators into account. figure 1 depicts the basic flow of Ensemble Learning. The advantage of integration learning is that it improves the generality and robustness of a single estimator and provides better prediction performance than a single estimator. Another feature of integrated learning is that it can be easily parallelized.

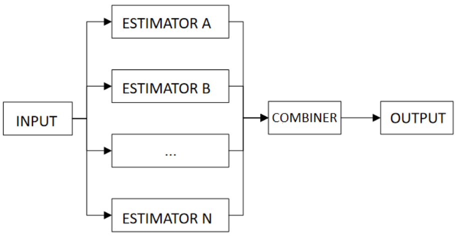

Figure 1. Basic flow of integrated learning

In a word, a random forest is made up of several CART (Classification And Regression Tree). For each decision tree, the training set they use is sampled with a put-back from the total training set, which means that some samples from the total training set may appear multiple times or none in the training set of a tree. When training the nodes of each tree, the features selected from all in a certain proportion are used randomly. Assuming that the total number of features is M , and the ratio can be $\sqrt{M}$, $\sqrt{M}/2$ and $2\sqrt{M}$.

2) Implementation of random forest

A random forest is a forest with a number of decision trees, and each of which is unrelated to others. After creating the forest, each decision tree in the forest is judged separately when a new input sample enters. The outcome of the judgement is determined by a vote of these decision trees. As shown in figure 2 in the decision tree generation process, different decision trees are obtained by randomly selecting samples from the training set and randomly selecting features for node splitting, resulting in decision trees with different learning profiles, which are eventually combined to become a global strong learner by combining these decision trees[4-5].

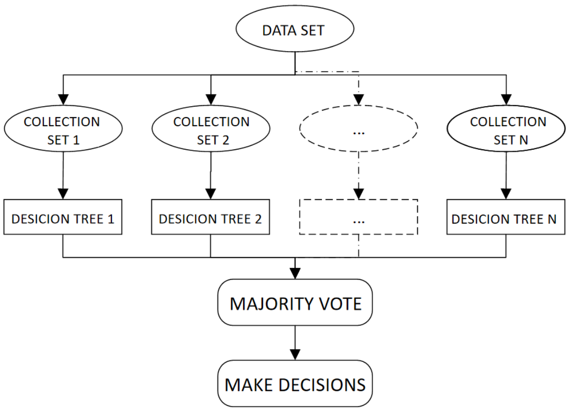

Figure 2. Schematic diagram of the random forest generation process

a) The training process of the random forest is as follows.

(1)Given a training set S , a test set T , and a feature dimension F . Determine the parameters: the number of CART t , the depth of each tree d , the number of features used at each node f . The termination conditions: the minimum number of samples at the nodes s and the minimum information gain at the nodes m ;

For number $1-t$ of tree , $i=1-t$.

(2) Extract training sets $S(i)$ of the same size of S and put them back after used, then select samples as the root nodes randomly, and start training from the root node.

(3) If the termination condition is reached on the current node, set the current node as a leaf node. For classification, the predicted output of the leaf node is the class with the highest number of samples in the current node's sample set $c(j)$. For regression, the predicted output is the average value of the sample values in the current node's sample set. After that, continue training other nodes. If the current node does not meet the termination condition, randomly select f feature from the F feature set without replacement. Using such f feature, find the one-dimensional feature k and the threshold th that produce the best classification effect. Samples with dimension k feature values below the threshold th are divided into the left node, and the rest are divided into the right node. Continue training other nodes.

(4)Repeat (2)(3) until all nodes have been trained or marked as leaf nodes.

(5)Repeat (2), (3), (4) until all CART have been trained.

b) The prediction process of the random forest is as follows.

For number $1-t$ of tree , $i=1-t$.

(1) Start from the root of the current tree, determine whether to enter the left node ($< th$) or the right node ($\geq th$) according to the threshold of the current node,and continue until reaching a leaf node, then output the prediction data.

(2) Repeat (1) until all trees have output their predictions. For classification, the output is the class with the highest total predicted probability among all trees, which is the sum of probabilities for each c(j) of p. For regression, the output is the average of the outputs of all trees[6].

Random Forest has good performance in both regression and classification. The data set used in the study contained a large number of features, and it was initially unknown which features had a greater impact on the classification results in the rainfall prediction model. By constructing multiple decision trees, the final result is determined by voting based on the results obtained from each decision tree. The "random" process in Random Forest effectively reduces the impact of each feature on the classification result.

The use of Random Forest in this study for rainfall prediction is considered due to its unique advantages: (1) Random Forest can handle high-dimensional data, which means that conditionality reduction is not required for the source data; (2) it can evaluate the importance of each feature in a classification problem (Feature Importance), without the need for subjective determination of the weight values of each feature.

C Support vector machine algorithms

1) Introduction and main ideas of Support Vector Machines

Support Vector Machine (SVM) is a machine learning algorithm that emerged in the 1990s based on the theory of statistical learning. It has attracted a lot of attention due to its excellent generalization performance in a variety of classification problems. By learning from input data, the SVM algorithm can screen out data that has a significant impact on the classification results, and use these data as support vectors to maximize the classification margin between different categories, thereby achieving data classification with high accuracy and good generalization ability. SVM is currently one of the most advanced classification techniques[7], and benchmark studies have shown that SVM performs better in classification than other techniques [8]. Many experiments have also demonstrated that SVM can achieve satisfactory classification accuracy with a limited amount of training data, so it is widely used in classification and prediction.

The main idea of SVM is to map the input data to a high-dimensional space and establish an optimal decision hyperplane in this space, such that the distance between the two closest samples from different classes on either side of the hyperplane is maximized. This provides better generalization ability for classification problems, with the two sides of the hyperplane representing different categories.

SVM is a powerful classification and regression technique with the following characteristics: (1) it can maximize the accuracy of model predictions without over fitting the training data; (2) it is particularly suitable for analyzing data with a large number of predictive variables; (3) it can be applied to high-dimensional sample data. Although mapping low-dimensional samples to high-dimensional space may lead to dimension explosion, this problem can be cleverly solved by using an inner product kernel function.

2) Implementation of Support Vector Machines

Support vector machine is a powerful classification and regression technique. Suppose that there are N training samples, each can be expressesd as $(x_i, y_i)(i=1,2,...N)$, among them the x_i is used to express the feature vector of $(x_{1i}, x_{2i}, ..., x_{ni})^T$, each sample contains n features in total. In the figure below, when $n=2$, we set the x_1 as X-axis, x_2 as Y-axis, use color to indicate y .

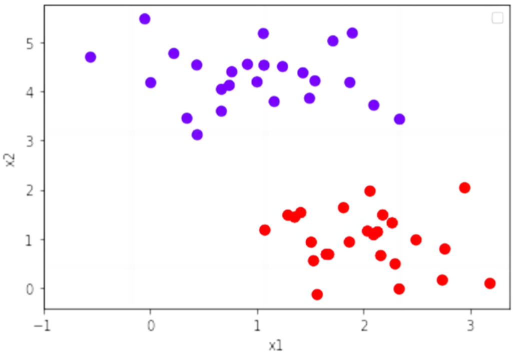

Figure 3. Distribution of the sample data points

Set all purple labels to 1, the red labels to-1, and the decision boundary is set in a two-dimensional plane to:

$$x_2 = ax_1 + b$$

The above formula can be:

$$0 = \omega^T x + b$$

Where ω is the parameter vector, x is the feature vector and b is the intercept. Thus the above formula can be expressed as the decision boundary in this training set. So it can be concluded that SVM is the solution of parameter vector ω and intercept b .

Set any two points on the decision boundary x_a , x_b .Then the two-point expression is obtained:

$$\omega^T x_a + b = 0$$

$$\omega^T x_b + b = 0$$

Reduce these two formulas:

$$\omega^T * (x_a - x_b) = 0$$

The transpose of one column vector multiplied by another column vector yields the dot product of two vectors, that is $\omega \cdot \left(x_a - x_b\right)$, since the two points(x_a , x_b) lie on the same line. The parameter vector $\omega \cdot \left(x_a - x_b\right) = 0$ can be used to conclude that the direction ω of the decision boundary is perpendicular to it.

Considering an arbitrary point x_p above the decision boundary $p > 0$, we can obtain:

$$\omega \cdot x_p + b = p$$

Similarly, considering a point below the decision boundary r , $r < 0$, we have:

$$\omega \cdot x_r + b = r$$

Before continuing, let us establish that we define the points above the decision boundary as +1 and below as -1. Thus, we can rearrange the previous equations as follows:

$$y = \begin{cases} 1 & \omega \cdot x_t + b > 0 \\ -1 & \omega \cdot x_t + b < 0 \end{cases}$$

The two sides of the decision boundary need two hyperplanes, which represent two parallel lines in the two-dimensional space. The dashed line shown in figure 4 represents the hyperplane. The distance between the two is d that is, the margin.

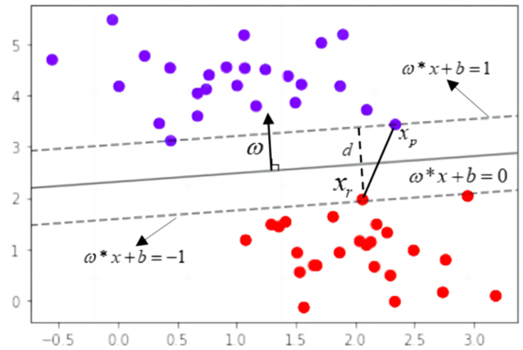

Figure 4. Illustration of the decision boundary and hyperplane of the support vector machine

Based on figure above, we can deduce that:

$$\begin{cases} \omega \cdot x_p + b = 1 \\ \omega \cdot x_r + b = -1 \end{cases}$$

Subtracting both equations yields:

$$\omega \cdot \left(x_p - x_r\right) = 2$$

By referring to above figure again , we can see that $\left(x_p - x_r\right)$ expression represents the line connecting the two points, and the margin d is parallel to ω . Thus, we can write:

That is,

$$\frac{\omega \cdot (x_p - x_r)}{\|\omega\|} = \frac{2}{\|\omega\|} \quad or \ d = \frac{2}{\|\omega\|}$$

The goal of the support vector machine is to maximize the margin, which implies minimizing. We can transform this into the following equation for minimizing:

$$f(\omega) = \frac{\|\omega\|^2}{2}$$

For any sample, we can express the decision function as:

$$\begin{cases} \omega \cdot x_i + b \geq 1 & y_i = 1 \\ \omega \cdot x_i + b \leq 1 & y_i = -1 \end{cases}$$

After integrating, we can obtain:

$$y_i(\omega\cdot x_i+b)\geq 1, i=1,2,...N$$

Finally, we arrive at the basic form of the support vector machine equation:

$$\left.\begin{array}{r}\min\limits_{\omega,b}\dfrac{\|\omega\|^2}{2}\\ subject\ to\ y_i(\omega\cdot x_i+b)\geq 1, i=1,2,...N.\end{array}\right\}$$

In cases where the data is not linearly separable, we can introduce the soft margin property to obtain the final decision function:

$$f(x_{test})=sign(\omega\cdot x_{test}+b)=sign(\sum_{i=1}^{N}\alpha_i y_i x_i\cdot x_{test}+b)$$

x_{test}

is any test sample, and the sign function $sign(h)$ returns +1 when $h>0$ and return -1 when $h<0$.

III. Research Ideas and Data Processing

A Research's Ma Idea

In this paper, we will use KNN, Random Forest, and Support Vector Machine algorithms to construct models and make predictions based on the information contained in the existing data set, and obtain research results. The research approach used for rainfall forecasting is shown in the figure 5.

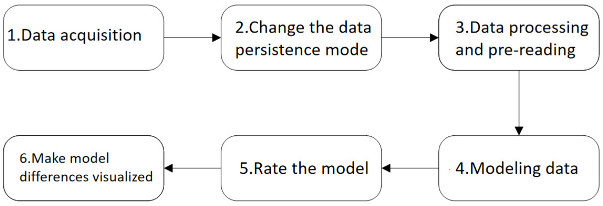
1.Data acquisition
2.Change the data persistence mode
3.Data processing and pre-reading
4.Modeling data
5.Rate the model
6.Make model differences visualized

Figure 5. Basic flowchart of research ideas

(1) Change the data persistence method. Since the data exists in a CSV file, the data persistence method needs to be changed, and the data is stored by using the MySQL database. When analyzing data through Python, it is necessary to master Python's database operation technology proficiently.

(2) Preprocess the data. "Data and features determine the upper limit of machine learning, and models and algorithms only approximate this limit." This sentence basically answers the necessity of doing data preprocessing from a macro perspective. This study will also carry out meticulous preprocessing operations on the source data, including handling of missing values, outlier values, and feature engineering. In the step of feature engineering, the study will delete irrelevant features for the rainfall problem, and focus on handling difficult features in the data set.

(3) Unified training set and test set. The data set will be divided into a training set and a test set in a certain proportion (7:3). The split data set will be uniformly used in the KNN, Random Forest, and Support Vector Machine algorithm models in this study.

(4) Model the data for prediction. Understand and master the principles of various machine learning algorithms, use Python language and related machine learning algorithm libraries to construct models for prediction, record the experimental results of KNN, Random Forest, and Support Vector Machine predictions, and use the comprehensive evaluation index (F1-Measure) value as the main evaluation index, combined with accuracy (Accuracy) to evaluate the advantages and disadvantages of each model.

(5) Data visualization. Use Python visualization libraries (such as Matplotlib) to visualize the prediction results from multiple angles and dimensions. Pay key attention to visualizing the differences in prediction results of various algorithms.

(6) Draw conclusions. Draw experimental conclusions based on the results, and provide relevant decision-making recommendations for the demand.

B Data sources

The data used in this study was selected from the meteorological data set recorded by 49 weather stations across Australia, provided by the Kaggle website. The data set covers a period from December 2008 to June 2017, totaling 145,460 records and spanning various regions in Australia. It contains specific elements, such as temperature, air pressure, precipitation, and relative humidity, as shown in Table 1. Due to space limitations, the data is displayed after being transposed from the original data.

TABLE 1. Part of Data in the Data Set

	52344	52345	52346	52347	52348
Date(2013)	10-31	11-01	11-02	11-03	11-04
Location	MountGinini				
MinTemp	3.7	4.5	8.0	8.7	-4.8
MaxTemp	16.9	17.6	18.9	15.6	11.7
Rainfall	0.0	0.0	0.0	0.0	0.0
WindGustDir	SW	WSW	W	W	SSW
WindGustSpeed	33.0	33.0	46.0	83.0	43.0
Humidity9am	52.0	51.0	38.0	49.0	56.0
Humidity3pm	39.0	39.0	40.0	40.0	32.0
Pressure9am	NaN	NaN	NaN	NaN	NaN
Pressure3pm	NaN	NaN	NaN	NaN	NaN
Cloud9am	NaN	NaN	NaN	NaN	NaN
Cloud3pm	NaN	NaN	NaN	NaN	NaN
Temp9am	9.3	10.6	12.1	11.7	1.6
Temp3pm	15.0	17.1	13.4	12.0	10.8
RainToday	No	No	No	No	No
RainTomorrow	No	No	No	No	No

C Data pre-processing

1) The processing of omitted and null values

We identify the feature data and tag data after reading the data from the database. The "RainTomorrow" field name is cited as the tag data in this instance. Following table shows that the label data consist of 3247 pieces of data with null values, which is 2.24% of the total source data samples. As the experimental data must be valid and reliable and the data do not make up a significant amount, the label data's null values are eliminated.

TABLE 2. Total number and proportion of each type of label data

Label Categories	Total Number	Proportions
No	110316	75.84%
Yes	31877	21.92%
None	3247	2.24%

2) The processing of continuous variables

In processing continuous variables, the primary objective is to transform the data into a dimensionless format. This can be achieved through two primary methods: standardization (also known as Z-score normalization) and normalization (also known as Min-Max scaling). For this study, standardization was utilized to process continuous variables. The process of standardization involves scaling the training data by a specific ratio when the sample data is dispersed or the sample scale is non-uniform. The standardization formula is as follows:

$$x^* = \frac{x-\mu}{\sigma}$$

After centering and scaling continuous variables, the resulting data distribution conforms to a standard normal distribution with a mean of 0 and a variance of 1.

D Evaluation metrics for models

Evaluation metrics for classification models include various indicators such as accuracy, precision, recall, and F1-score. Despite any prediction made by a model with regards to the occurrence of rain on the following day being classified as "No," the accuracy of the model on the test set sample is still determined to be 75.84%. As a result, using accuracy as a sole evaluation metric may result in arbitrary model predictions. In this study, F1-score will be used as the primary evaluation metric due to the severe data imbalance.

The F1-score's mathematical formula is shown as follows:

$$F_1 - sore = \frac{2 * precision * recall}{precision + recall}$$

$precision$ represents accuracy rate, $recall$ represents recall rate.

By utilizing comprehensive evaluation metrics and combining precision, it is possible to better assess the generalization ability of the three classification models used in this study.

IV. Predictive Outcomes and Analyses Based on Machine Learning Algorithms.

A Rainfall prediction result based on the KNN algorithm

Python was used for coding and data analysis due to its rich library support and high development efficiency. The KNN algorithm was used to build and analyze the model with the Sklearn library, and the accuracy of the model without parameter tuning was found to be 82.4464%, and the Comprehensive Evaluation Metrics (F1-score) was 52.2388%.

The experimental outcomes following the optimization of parameter K are presented in Table 3.

TABLE 3. The accuracy & F1-score values corresponding to K

K	Training set fit rate (%)	Accuracy (%)	F1-score(%)
1	99.9989	78.7941	51.0868
2	88.3006	80.9836	40.0266
3	89.4127	81.5533	52.3985
4	86.3987	81.9143	44.7389
5	87.1532	82.4464	52.2388
6	85.5096	82.2870	46.0054
7	86.1526	82.7371	51.6861
8	84.9902	82.5121	46.6838
9	85.4644	82.8028	51.1324
10	84.7008	82.5589	46.5747
11	82.8051	82.8051	50.2611

The obtained results were visualized through plotting, as illustrated in figure 6.

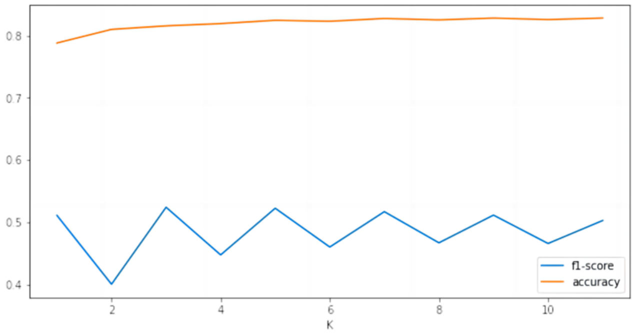

Figure 6. The accuracy and F1-score corresponding to various K

Based on the combined observations from Table 3 and figure 6, it can be deduced that when K is set to 3, the F1-score reaches its maximum value of 52.3985%. Additionally, it can be observed that the accuracy does not vary significantly with changes in K, and there is no evidence of overfitting in the training set. Therefore, it can be inferred that the optimal value of K for the KNN model is 3.

B Rainfall prediction outcomes based on the random forest

By employing a random forest model for prediction and drawing on theoretical knowledge, it can be inferred that the random forest model has a considerable number of parameters for tuning. By default, the accuracy of the random forest model prediction is 85.571% with an F1-score of 50.2611%. The default configuration for the model involves constructing 100 trees. A learning curve, depicted in figure 7, indicates that the highest value for n_estimates appears to be between 165 and 175.

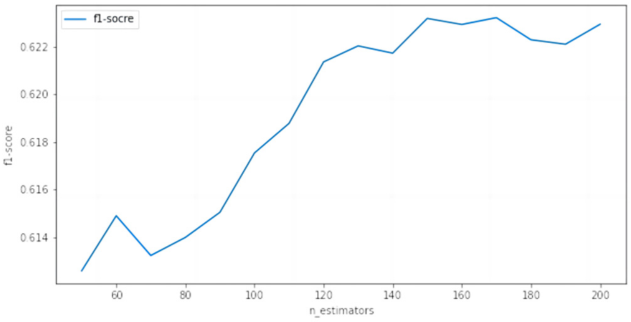

Figure 7. The number of constructed trees corresponds to the F1-score line graph

Based on the analysis of the experimental results presented in Table 4., it was determined that the optimal value for n_estimates is 170. Compared to the F1-score obtained during parameter tuning, there was a significant improvement in the data. Therefore, it is possible to continue adjusting other parameters based on this outcome. Additionally, the model was found to be in a severe overfitting state without pruning. However, setting a maximum depth for the trees resulted in a significant reduction in overfitting.

TABLE 4. The accuracy and F1-score corresponding to n_estimates

n_estimates	Training set fit rate (%)	Accuracy(%)	F1-score(%)
165	99.9989	85.7705	62.4799
166	99.9989	85.7682	62.3597
167	99.9989	85.7541	62.4296
168	99.9989	85.7471	62.2922
169	99.9989	85.7916	62.5332
170	99.9989	85.7682	62.3223
171	99.9989	85.7705	62.4474
172	99.9989	85.7330	62.2128
173	99.9989	85.7752	62.4644
174	99.9989	85.7518	62.2765

TABLE 5. The accuracy and F1-score corresponding to the max_depth parameter

max_depth	Training set fit rate (%)	Accuracy(%)	F1-score(%)
8	85.3348	84.4249	55.4034
9	85.9386	84.6570	56.6929
10	86.8036	84.7906	57.3886
11	87.8716	84.9852	58.5141
12	89.0882	85.1071	59.1315
13	90.4194	85.1657	59.4878
14	91.9033	85.3017	60.1246
15	93.1812	85.4915	60.9255
16	94.3346	85.4540	60.8492
17	95.4046	85.5830	61.4032
18	96.3510	85.6416	61.5963
19	97.1778	85.6416	61.8736
20	97.8731	85.6369	61.8896

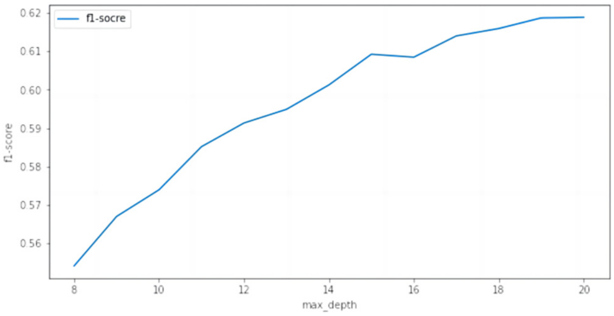

Figure 8. The line graph of the relationship between max_depth and f1-score

The present analysis integrates Table 5 and figure 8 to demonstrate that the overfitting phenomenon becomes more pronounced as the number of tree layers increases. However, the accuracy and F1-score values of the data in the current study increase proportionally. Notably, the visual inspection of figure 8 reveals that the accuracy and F1-score values increase at a considerably decelerated rate when the maximum number of layers in the tree reaches 20.

In summary, the model is no longer able to be improved by pruning the trees through parameter adjustment, and the best results were found to be 61.8896% F1-score and 85.7283% accuracy when using the random forest for rainfall predicting at a maximum depth of 31 layers. However, the corresponding data is very heavily overfitted. If the model is tested with more new data, the theoretically better results are obtained by pruning the model, so this study determines that the theoretically best model is obtained when 169 trees are created and the maximum depth is set to 19 layers, i.e. the final result is an F1-score of 61.8736% and an accuracy of 85.6416%.

C Rainfall prediction outcomes based on the support vector machine

Based on the rainfall prediction using support vector machines (SVMs), the default parameters for model creation yield an accuracy of 84.6382% and an F1-score of 57.2788%. This study takes this outcome as a benchmark for parameter tuning. The first step is to determine the kernel function that should be adopted for prediction analysis. Table 6 presents the variation of F1-score values under different kernel functions. When using the polynomial kernel, the experiment sets the degree parameter to 2, and the results obtained using other polynomial kernels are not reported in the table. When the degree is set to 1, the experiment's accuracy is 84.2820%, and the F1-score value is 57.2251%. This finding suggests that the data may exhibit a tendency toward linear distribution. Moreover, the analysis reveals that the performance of the Hyperbolic Tangent Kernel function is relatively poor compared to that of other kernel functions. The Gaussian Radial Basis Function exhibits the best performance.

TABLE 6. The accuracy and F1-score corresponding to various kernel functions

Kernel functions	Training set fit rate (%)	Accuracy (%)	F1-score (%)
Linear Kernel	84.3030	84.2937	57.2322
Polynomial Kernel	84.8666	84.6781	57.2251
Gaussian Radial Basis Function Kernel	84.2939	85.2032	60.4114
Hyperbolic Tangent Kernel	63.0863	63.0714	17.6486

This study aims to obtain optimal performance by conducting a detailed parameter tuning of the Gaussian radial basis function as a kernel function

TABLE 7. The accuracy and F1-score corresponding to various gamma values of the Gaussian radial basis function kernel.

Gamma	Training set fit rate (%)	Accuracy (%)	F1-score(%)	F1-score relative improvement percentage
0.0100	85.6955	85.0251	58.5947	-3.01%
0.0144	86.1084	85.1610	59.3239	-1.80%
0.0188	86.5374	85.1891	59.6912	-1.19%
0.0233	86.9835	85.2478	60.0976	-0.52%
0.0278	87.4396	85.2665	60.3044	-0.18%
0.0322	87.9007	85.2946	60.5299	0.20%
0.0367	88.3277	85.2970	60.5733	0.27%
0.0411	88.7245	85.2759	60.5737	0.27%
0.0455	89.1334	85.2009	60.4026	-0.01%
0.0500	89.5212	85.1774	60.3150	-0.16%

Table 7 presents that the application of SVM in rainfall prediction can lead to an enhancement in accuracy by 0.0727% and a relative increase in F1-score by 0.27% upon adjusting the gamma parameter, albeit with a marginal increase in magnitude. Consequently, further parameter tuning is recommended based on the gamma parameter obtained.

In conclusion, our findings suggest that parameter tuning of the support vector machine (SVM) can significantly improve its F1 score, albeit with a slight decrease in accuracy. Notably, as the F1-score improves, the model exhibits better performance in predicting the minority class compared to the model without parameter tuning. Specifically, the SVM model achieved an F1-score of 65.1689% and an accuracy of 81.3915%.

D Analysis and Comparison

Based on the predictive results of the three models, which underwent modeling and parameter tuning, it is apparent from figure 9 that there is no significant difference in F1-score performance among them. Nevertheless, the K-nearest neighbors (KNN) model demonstrates a relatively inferior F1 score compared to the other models. The differences in accuracy of the three algorithms are evenly not very substantial.

Upon comparing the predictive performance of random forest and support vector machine (SVM), it is observed that random forest exhibits the highest accuracy and is the most effective in predicting rainfall accuracy, while SVM achieves the highest F1-score and demonstrates superior performance in predicting the minority class compared to other models.

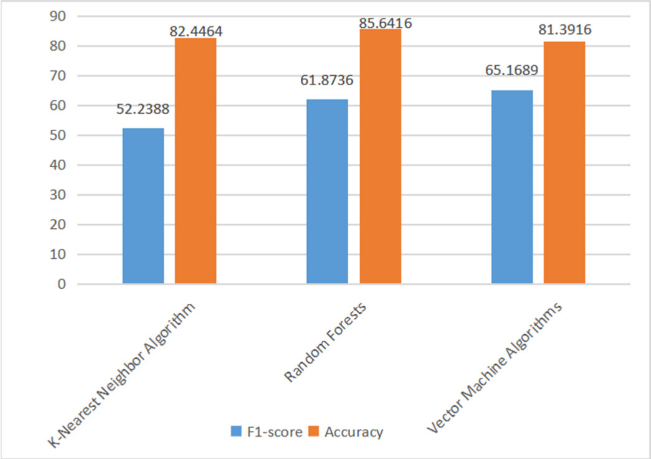

Figure 9. The accuracy and F1-score histograms for each model

To further elaborate, although SVM does not show higher predictive accuracy than the random forest on the test set due to the severe imbalance in the distribution of training samples. Hence, when confronted with the necessity of predicting the possibility of short-term rainfall, it is preferable to employ a support vector machine as the model of choice for forecasting purposes.

V. Summary

This paper focuses on the prediction of rainfall using K-nearest neighbors, random forest, and support vector machine models(SVM) built on over 140,000 records from 49 meteorological stations in Australia. The study presents the following main points:

(1) Based on a review of relevant literature, this study provides a summary of domestic and international approaches to rainfall prediction. Researchers worldwide have explored various methods for predicting rainfall, including traditional empirical and physical statistical methods, as well as machine learning methods. A comprehensive review of the literature suggests that the use of machine learning algorithms can significantly improve the accuracy of rainfall prediction. This study presents significant implications for practical applications.

(2) Preprocessing of the data was done in this work, and three different methods were developed and tested on a uniform training and testing dataset.

(3) Prediction analysis is performed after building K-nearest neighbors, random forest, and support vector machine (SVM) models.

The results indicate that support vector machines achieved the best F1-score of 65.1639%, while random forest had the highest accuracy of 85.6416%. The K-nearest neighbors algorithm had the fastest training and computation speed. In contrast, the support vector machine required the longest training time, and its computation time doubled when dealing with larger datasets. Overall, the random forest algorithm performed better in terms of speed, accuracy, and F1 score, while the support vector machine excelled in predicting minority classes.

REFERENCES

[1] Liu K, Pan J, Zhang RG. 2018. Research progress of summer rainfall prediction methods in China [J]. People's Yellow River, 40(01):18-22.

[2] Li ZY, 2021,Climate rainfall prediction model based on machine learning[D]. Chengdu University of Technology.

[3] Liu Z. 2010.Application of K-nearest neighbor algorithm in automatic text classification [J]. Journal of SuZhou Vocational University, 21(02):58-60.

[4] Guyon I, Weston J, Barnhill S, et al. 2002. Gene Selection for Cancer Classification using Support Vector Machines[J]. Machine Learning, 46(1-3):389-422.

[5] Hsu C W, Lin C J.2002. A comparison of methods for multiclass support vector machines[J]. IEEE transactions on Neural Networks, 13(2): 415-425.

[6] Cao ZF. 2014. Research on optimization of random forest algorithm[D]. Capital University of Economics and Business.

[7] Wutao Li, Zhigang Huang, Rongling Lang, Honglei Qin, Kai Zhou, Yongbin Cao. 2016.A real-time GNSS interference monitoring technique based on dual support vector machine method[J]. Sensors, 16(3).

[8] Sonnenschein A.,Fishman P.M. .1992. Radiation detection of spread spectrum signals in noise with uncertain power[J]. IEEE Transactions on Aerospace and Electronic Systems, 28(3).