International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Analogousness Enhanced Rainfall Predictor using XGBoost Backbone

Govardhana Meti[1], Dr. Ravi Kumar. G. K[2]

Submitted: 10/11/2022 **Accepted:** 11/02/2023

Abstract: The forecasting of intense rainfall presents a significant challenge for the meteorological department because of the strong connection between rain and the economy as well as human lives, but extreme climate shifts have made it more complicated than ever before to estimate precipitation accurately. In a country that relies heavily on agriculture, the precision of rainfall forecasts is vital. Predicting rainfall is a common application for machine learning systems. By figuring out the hidden patterns in weather data from the past, these methods can almost accurately predict when it will rain. This study proposes a novel machine learning method called Analogousness Enhanced Rainfall Predictor using XGBoost Backbone to foretell rainfall. The proposed method uses the basis of XGBoost and tunes parameters for it to get higher accuracy for the outcomes. This study uses a large dataset of weather observations collected over ten years in various places in Australia. This model successfully deals with the issue of the data-class imbalance issue.

Keywords: *Rainfall prediction, Machine Learning, Classification, Extreme Gradient Boosting, Data Imbalance*

1. Introduction

The art of weather forecasting is a challenging but gratifying endeavor. The term "weather data" refers to a variety of different aspects of the atmosphere, including wind speed, humidity, pressure, temperature, and so on [1]. Rainfall prediction is one of these weather forecasting tasks. It is a complicated atmospheric process that is dependent on a wide variety of elements that are associated with the weather. The order to get an appropriate and pertinent forecast of rainfall can be beneficial in a variety of contexts, including but not limited to the effective management of water resources, the issuing of flood warnings before time, the management of flight operations, and the limitation of transport and construction activities [2, 3]. This is why the ability to forecast rainfall accurately has captured the interest of not only the research world but also administrations, businesses, and relevant institutions with risk management. Farmers are able to improve managing their harvests and contribute to the financial development of their nation when they have access to information about rainfall in the past. Predicting when and how much rain will fall can be difficult for climatologists due to the unpredictability of both the time and the amount of precipitation [4].

One of the most important features of present research in the field of research arena is the assessment of time series data [5]. The data that makes up this are compiled over a particular time span [6-8]. Methods such as machine learning and data mining can be applied to analyze historical time series information to uncover previously undiscovered patterns and make refined predictions on future trends [5, 8]. Supervised and unsupervised learning are the two core sorts of ML methods. The output of supervised techniques is first categorized based on the training data, which consists of data that has already been pre-classified [9-11]. Unsupervised processes, on the other hand, do not need any groundwork at all; instead of working with pre-labeled data, such approaches employ algorithms to uncover latent structure in unlabeled data. According to the most recent research, it has been found that researchers prefer integrated methods for forecasting rainfall since these methods have a high level of accuracy.

This paper proposes an efficient novel rainfall predicting model using a fusion of machine learning methods. The details on the weather in real-time were gathered from a number of sensors that are dispersed throughout distinct cities in Australia at distinct vital spots. An enhanced technique backed by the extreme gradient boosting algorithm is used here to foretell rainfall effectively. The synthetic minority oversampling technique was used to recover from the issue of the data imbalance in this study [12-14]. The K-fold cross-validation technique [15, 16] was utilized to find the appropriate values of parameters required for this experiment.

Section 2 of this paper discusses various works related to this experimental study. Section 3 presents the various methodology employed in due course of this study and the novel machine learning technique proposed by this research. Section 4 presents the results and discussions. Section 5 concludes the paper.

[1] *Computer Science & Engineering, BGS Institute of Technology, Karnataka, India*
[2] *Computer Science & Engineering, BGS College of Engineering and Technology, Karnataka, India*
[1]*govardhan.meti@gmail.com*

2. Literature Review

Over the last 20 years, the main goal of many research teams has been to improve the accuracy of weather forecasting using machine learning. Some of the studies in this area are talked about here.

Sawale *et al.* describe a technique that takes an ANN to predict the weather based on a set of data that includes heat, moisture, and wind velocity. The authors used a Back Propagation Network and a Hopfield System to make a hybrid method. The results from the BPN are sent to the hybrid system, which is in response to making the actual predictions. The proposed method can figure out the non-linear relationship between historical data on parameters like temperature, wind speed, humidity, etc., and then use that information to predict what the weather will be like [17]. In Liu *et al.*, researchers came up with a way to predict rain that used genetic algorithms to choose which features to use and Naive Bayes to predict the rain. Its suggested solution had two stages. The first phase was to predict whether or not it would rain, and the final method was to categorize the rain as light, moderate, or strong [18]. Kumar *et al.* utilized ANN to foretell the mean amount of rain in India during the monsoon season. For predicting, a set of data was used that covered eight months each year. It was assumed that it would rain a lot during the chosen months. Look-ahead Back Propagation, Stack Reoccurring, and Cascaded networks were used to look at how well they worked [19]. ABF neural network was utilized by Philip et al. for the purpose of forecasting annual rainfall in the Kerala region. According to their findings, the ABFNN performs notably better than the Fourier analysis [20].

Wu *et al.* made a forecast for the precipitation that would fall in India and China. They used a system known as the Modular ANN [21]. In the paper by Joseph J. *et al.*, the authors of the study developed a hybrid approach for rainfall prediction by integrating feature extraction techniques with forecasting models. The experiment relied on a dataset that was acquired from National Oceanic and Atmospheric Administration. This dataset covered a period of further than 50 years and included data on a variety of meteorological characteristics. On the basis of a previously established training set, a Machine Learning technique was applied to the problem of categorizing the occurrences into low, middle, and high classes [22].

Grover *et al.* proposed a Methodology for Weather Prediction, which generates forecasts by taking into account the combined influence of key climatic variables. For the purpose of the study, they collected data that were observed in sixty different locations around the United States from 2009 to 2015. The proposal includes a cutting-edge hybrid approach that is both discriminative and generative in nature. Combining a bottom-up predictor to every independent factor with a top-down deep learning model that simulates the combined empirical correlations is what the proposed architecture does. The framework also includes an information kernel, which is based on a similarity measure that is automatically learned from the data [23].

Researchers analyzed various machine learning strategies for the purpose of forecasting rainfall in Malaysia and presented their findings in Zainudin *et al.* [24]. Five different conventional machine learning strategies were included in the mining approaches. Before classification could begin, the dataset underwent pre-processing so that any gaps in its values could be filled in and any noise could be eliminated. The Random Forest algorithm fared significantly better than the others; it accurately identified a greater number of examples using a less amount of the training data. Scholars described the method of rack-mountable Support Vector Machine (SVM) in Lu *et al.*, which may be used to forecast and imitate rainfall. The method that was suggested involved a number of stages, including the production of v–SVM, the mentoring of the SVM kernel function, the sampling of SVM pairing representatives through the use of the Partial Least Square method, and the creation of a training dataset through the bagging sampling technique. The method was put to the test in Guangxi, China, where it beat competing models when it came to monthly rainfall forecast [25].

3. Methodology

The methods used in this study include exploratory data analysis, multiple data pre-processing techniques, various machine learning techniques, and scaling of weights and oversampling. The main aim of this study is to predict rainfall precisely. The Flow Diagram of the Methodology for the proposed work is presented in Fig. 1.

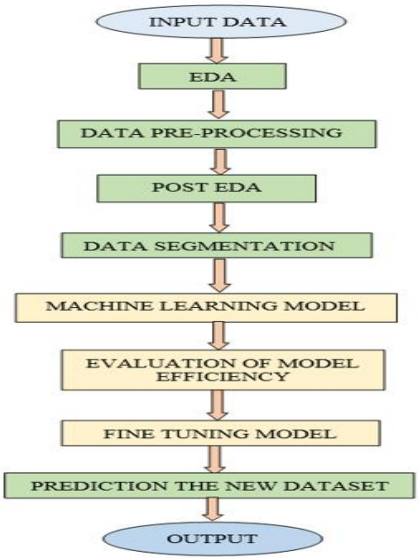

Fig. 1. The Flow Diagram of Methodology for proposed work.

3.1. Dataset

The data were compiled from a large number of meteorological stations. This dataset includes approximately ten years' worth of daily weather observations gathered from a wide variety of sites across Australia. The total number of observations presented here in this dataset is 145459. A wide variety of climatic characteristics, such as rainfall course, rainfall amount, lowermost temperature, uppermost temperature, airstream speed, humidity, etc., are included in the dataset.

3.2 Exploratory Data Analysis

Exploratory Data Analysis, popularly known as EDA, is a way to investigate the information by graphical tools. It is used to detect tendencies and patterns or to confirm expectations for the use of statistical summaries and graphical depictions. It is not easy to glance at a table of numbers or an entire spreadsheet and discern the significant aspects of the data just by looking at them. To gain insights by merely looking at the numbers might be a time-consuming, uninteresting, and overwhelming process. In order to provide assistance in dealing with this matter, EDA strategies have been developed.

3.3. Data Pre-processing and Standardization

Missing data handling is the first step, to begin with, data pre-processing. For the data column, which has a low amount of data missing, they are assigned with values using backward and forward filling methods. The data columns, which are missing more than 40% of the data, are dropped or removed from the dataset to be used in this study. After this step is done, 18 data columns are left in the dataset. In the feature creation stage, the column responsible for the date is removed, and the day, the month, and the year columns are created in place of that to make a date more efficient for the study. Now we got 22 different features in this dataset to work with. For the category type feature, the dummy columns or columns of the binary type for each category are created. The value of each row is one if the category in question can be found in that row; otherwise, the value is zero. After all, these are done, the numerical and categorical data columns are merged together.

3.4. Data segmentation

After data pre-processing comes segmentation or splitting of the dataset to create training and testing datasets from it. The data for each dataset is chosen randomly.

3.5. ML models

In this research work, various machine learning classification methods were employed on the pre-processed and segmented dataset to observe the accuracy percentage for each case. Scikit-learn classifier models were used for these prediction experiments with necessary input data pre-processing, feature extractions, and optimizers.

3.5.1. Decision Tree Classifier

Decision Tree, a supervised learning technique, is a classifier as the arrangement of a tree. It has inner nodes which represent the attributes of a database, branches on behalf of the decision directions, and all leaf nodes in lieu of the deduction of the cataloging. It is a graphical portrayal to find all of the possible responses for a query or decide based on the given circumstances. As well as quantitative data, a decision tree may include categorical data (YES/NO). In this experiment with the decision tree classifier, the maximum leaf nodes were set to be 10. The definition of the best nodes is based on the relative drop in impurity. An additional parameter that is employed here is the random state, which affects the amount of randomness that the estimator generates. At each split, the traits are always randomly rearranged in different, i.e., random order, and there may be variations in found-to-be the best split across different runs. So, it is necessary to set the random state equal to an integer in order to get an indicating behavior during the fitting process. Here, that integer was set to be 101. Figure 2 represents a flow diagram of a decision tree classifier.

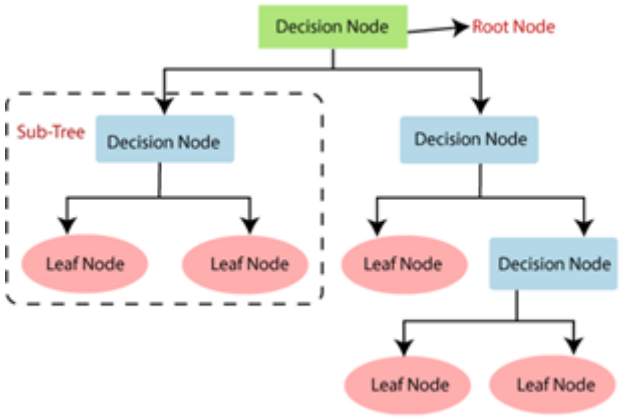

Fig. 2. A pictorial representation of a decision tree algorithm.

(Source: https://www.javatpoint.com/)

3.5.2. Random Forest Classifier

Random Forest is a classification algorithm that takes the average of a number of decision trees on different parts of a given dataset to improve the accuracy of that dataset's predictions. The random forest model, which is set on the basis of the idea of ensemble learning, does not rely on a single decision tree; rather, it considers the prognosis of each tree in the forest and determines the final output based on which tree's prediction received the majority of votes.

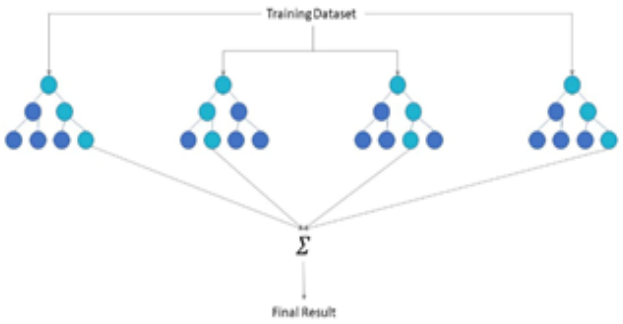

Fig. 3. A layout of the Random Forest Classifier

(Source: www.ibm.com)

Here, in this experiment, the number of trees which is represented by n-estimators was set to be 10. The greatest amount of depth that the tree can reach was set at 10. In this algorithm, n-jobs represent the number of tasks that can be executed simultaneously. Over the trees, the fitting, predicting, decision-path, and application processes these tasks are all parallelized. Here, the value of n-jobs was fixed at -1 due to parallelization of backend implementation, and the value of random-state was set as the same as the decision tree classifier. Then the training dataset was fed into this model to get the prediction for rainfall as output.

3.5.3. Extreme Gradient Boosting (XGBoost) Classifier:

Boosting is a method of ensemble modeling in which an attempt is made to construct a powerful classifier by combining a number of less effective classifiers. One of the most common types of boosting algorithms is called Gradient Boosting. For gradient boosting, every predictor is responsible for correcting the errors made by its predecessor. The implementation of gradient-boosted decision trees known as XGBoost can be found here. In this particular algorithm, judgment trees are constructed in a step-by-step fashion. This algorithm places a significant emphasis on weights. Following the process of assigning weights to each of the independent variables and then feeding that information into the decision tree that predicts results, the process is complete. The variables whose results were incorrectly anticipated by the tree have their weights increased, and these results are subsequently provided to the next decision tree. After that, all of these separate classifiers and predictors are combined to produce a robust and more accurate model. The advantage of XGBoost is that it overcomes the random fitting disadvantage (high bias and low variance), and it has multiple parameters to be tuned to build a good robust and efficient machine learning model.

In this experiment, the random seed is set at 43. ETA or learning rate identifies how soon the classifier can adapt the residual errors via making use of extra-base learners. Here, the learning rate is 0.01. The 'n_estimators' parameter represents the amount of boosting phases that must be completed, which is set here at 20. Gradient boosting is relatively resistant to over-fitting; hence a big number usually yields better results. Value for n_jobs and the maximum depth is kept as same as Random Forest Classifier.

3.54. Weighted XGBoost Classifier:

Now, if there is an imbalance in input data which is also present here in our dataset, using this data would not yield a better result. The data for 'Yes' class (1) is only 31877 and for 'No' class (0) is 110316. One of the classes has almost 3.5 times higher data than the other one, making the dataset highly imbalanced. The key concern here is that the system will be skewed toward forecasting the majority class. The algorithm will lack sufficient data to learn the underlying patterns in the minority class. So, another algorithm called Weighted XGBoost is implemented here. This includes a new parameter named scale_pos_weight, which is the ratio of the count of negative samples to the count of positive samples. This removes the issue of data imbalance. Because of this, the f1 score was the evaluation metric here. The F1 score is just the harmonic mean of the precision and recall portions of the test. Other than these, other parameters are kept as same as the standard XGBoost algorithm parameters.

3.5.5. The Novel Machine Learning Algorithm – Analogousness

Enhanced Rainfall Predictor using XGBoost Backbone:

To deal with the imbalance between data classes and to predict the rainfall estimation more accurately, we created a novel machine learning method which is called Analogousness Enhanced Rainfall Predictor, using XGBoost as the Backbone. The following Fig. 4 shows the algorithm for this study.

Step 1: Data collection and load the data from storage

Step 2: Descriptive data analysis to know insights of the data

Step 3: i) Handle missing data.
ii) Pre-process numerical and categorical columns separately.
iii) Feature creation
iv) Categorical data conversion
v) Merge numerical and categorical columns.
vi) Generate clean data.

Step 4: Label 'Yes' class as 1 and 'No' class as 0

Step 5: Data segmentation into two subsets- training, testing.

Step 6: Start with the design and development of machine learning model

Step 7: Use SMOTE method to balance the data classes

```
gendata = SMOTE()
```

Step 8: Analogousness enhanced XGB classifier model employment

```
model = xgb.XGBClassifier()

n_estimators = (40,50)
learning_rate = [0.01,0.1]
max_depth = (25,35)
scoring =["f1"]
```

Step 9: Hyper parameter optimization using XGBoost

Step 10: K-fold Cross Validation technique to find best value of parameters for XGBoost

```
kfold = StratifiedKFold(n_splits=4,
                        shuffle=True,
                        random_state=101)
```

Step 11: Evaluation of prediction model accuracy using precision, recall, F1 score

Step 12: Generate confusion matrix.

Step 13: Prediction of testing data

Step 14: Save the predictor output in a data table

Fig. 4. The algorithm for rainfall prediction.

In this algorithm, there are three main things to focus on. They are described below.

SMOTE technique to Deal with the Imbalanced Data –

As there is an imbalance between the 'Yes' and 'No' data classes, applying the model to this original dataset would have led to a biased output towards the 'No' data class as this class has more amount of data present in it. Analogousness means the condition of being in equivalence. So, as the name of the model suggests, we used a technique to balance the data. It is called Synthetic Minority Oversampling Technique (SMOTE) [26-28]. It is an oversampling method in which synthetic samples are created again for minority classes. The over-fitting problem that was caused by random oversampling can be solved with the help of this approach. It concentrates on the feature space in order to produce new examples with the assistance of approximation between the positive class instances that lie together in close proximity.

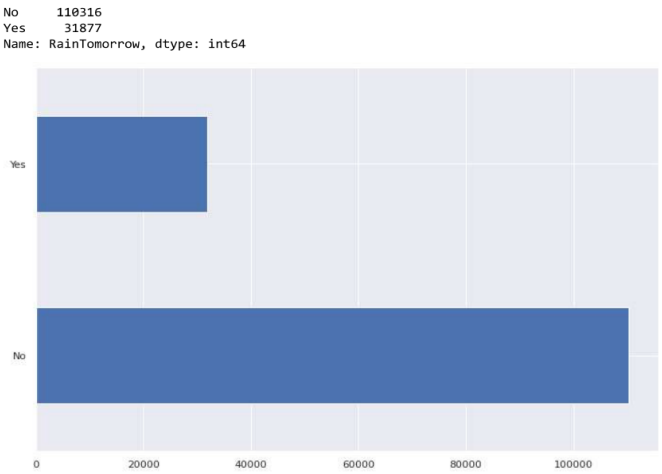

Fig. 5. The amount distribution of positive and negative classes in the original data

An object called 'gendata' was built to introduce the SMOTE algorithm through it. The data was fed into it and resampled using it. Then a library was created to generate the required balancing data. In this way, the number of 'Yes' class data was increased from 31877 to 88252, and the amount of data in the 'No' class was decreased to 88252. Thus, the analogousness for the input data was established in this experiment.

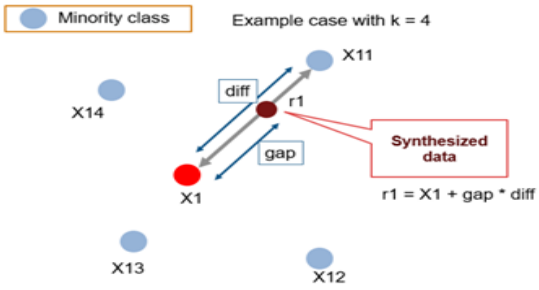

Fig. 6. Flow diagram of SMOTE technique

(Source: https://github.com/minoue-xx/Oversampling-Imbalanced-Data)

XGBoost library as the Backbone –

The parameters used in standard XGBoost are usually in integer form. Still, in this novel machine learning technique, the dynamic value was fed into these parameters to tune the features of the XGBoost model.

Model for finding best parameters of balanced XGBoost –

Fig. 7. Pictorial representation of k-fold cross-validation method.

(*Source: www.analyticsvidhya.com/blog/)*

A component was designed using K-fold Cross-Validation Technique to find the best value of the parameters used in this experiment [29]. In order to prevent overfitting, this technique is followed. In this case, all the permutations and combinations of the parameters are used to find the best value of them for the balanced XGBoost algorithm to work in an efficient way. Fig. 7 shows a pictorial diagram for this method.

4. Results and Discussions

For weighted XGBoost, there is a decrease in accuracy, but the f1 score is increased here. As the data for 'No' cases (0) is already high, so it already got the f1 score around 0.90 as a starting value. So, the main aim here is to increase the f1 score for 'Yes' cases (1). The scaling ratio that is being utilized in this scenario has the potential to encourage the system to over-correct any scaling errors that may have occurred while it was being trained. These errors may have been caused by the positive class. When it comes to making predictions about the 'Yes' class, this can, in turn, help the system achieve greater performance. If it is pressed too far, it could result in the algorithm over-fitting the 'Yes' class, which would result in either underperformance upon that 'No' class or on both classes.

The Analogousness Enhanced Rainfall Predictor using XGBoost as Backbone induces equilibrium among data classes and generates the best parameters to become an efficient model. For this novel machine learning technique, the F1 score increased from 0.85 (for weighted XGBoost case output) to 0.89 for negative class data and also increased the F1 score to 0.60 for positive class. In this experiment, more balanced data was used to train the model so that the model becomes less biased. The accuracy increased from 79.28% to 83.29%. The comparison study is presented in Table 1 below.

Table 1. The comparison of outcomes from different models.

Model	Class	Precision	Recall	F1-Score	Accuracy (%)
Decision Tree Classifier	0	0.8466	0.9538	0.8970	83.00
	1	0.7152	0.4017	0.5145	
Random Forest Classifier	0	0.8352	0.9741	0.8993	83.07
	1	0.7888	0.3344	0.4697	
XGBoost Classifier	0	0.8687	0.9489	0.9070	84.91
	1	0.7401	0.5038	0.5995	
Weighted XGBoost	0	0.9201	0.8026	0.8573	79.28
	1	0.5262	0.7589	0.6215	
Analogousness Enhanced Rainfall Predictor utilizing XGBoost Backbone	0	0.8808	0.9075	0.8939	83.29
	1	0.6423	0.5749	0.6067	

5. Conclusion and Future Scope

For the purpose of making accurate predictions of rainfall, the suggested scheme would gather feature-based meteorological data from weather sensors that are both technologically advanced and extremely sensitive. The accuracy of prediction obtained through the application of a novel supervised machine learning approach is incorporated into the proposed model. The modified technique using the XGBoost model improved the accuracy of the predictions because the data imbalance issue was balanced here. In the future, the training and testing data can be segmented in a different ratio. Other than this, here, the positive class was increased to overcome the imbalance of data issues. In future studies, the negative class can be decreased to test out the model instead of increasing the positive class.

References

[1] C. Wu and K.-W. J. E. a. o. a. i. Chau, "Prediction of rainfall time series using modular soft computingmethods," vol. 26, no. 3, pp. 997-1007, 2013.

[2] K. W. Chau and C. J. J. o. H. Wu, "A hybrid model coupled with singular spectrum analysis for daily rainfall prediction," vol. 12, no. 4, pp. 458-473, 2010.

[3] J. Wu, J. Long, and M. J. N. Liu, "Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm," vol. 148, pp. 136-142, 2015.

[4] A. Parmar, K. Mistree, and M. Sompura, "Machine learning techniques for rainfall prediction: A review," in International Conference on Innovations in information Embedded and Communication Systems, 2017, vol. 3.

[5] S. Aftab et al., "Rainfall prediction in Lahore City using data mining techniques," vol. 9, no. 4, 2018.

[6] M. A. Nayak, S. J. T. Ghosh, and a. climatology, "Prediction of extreme rainfall event using weather pattern recognition and support vector machine classifier," vol. 114, no. 3, pp. 583-603, 2013.

[7] T. Yue, S. Zhang, J. Zhang, B. Zhang, and R. J. J. o. E. M. Li, "Variation of representative rainfall time series length for rainwater harvesting modelling in different climatic zones," vol. 269, p. 110731, 2020.

[8] N. Mishra, H. K. Soni, S. Sharma, A. J. J. o. I. R. Upadhyay, and Applications, "A Comprehensive Survey of Data Mining Techniques on Time Series Data for Rainfall Prediction," vol. 11, no. 2, 2017.

[9] M. Ahmad, S. Aftab, and I. J. I. J. C. A. Ali, "Sentiment analysis of tweets using svm," vol. 177, no. 5, pp. 25-29, 2017.

[10] M. Ahmad, S. J. I. J. o. M. E. Aftab, and C. Science, "Analyzing the performance of SVM for polarity detection with different datasets," vol. 9, no. 10, p. 29, 2017.

[11] M. Ahmad, S. Aftab, I. Ali, and N. J. I. J. M. S. E. Hameed, "Hybrid tools and techniques for sentiment analysis: A review," vol. 8, no. 3, pp. 29-33, 2017.

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. J. J. o. a. i. r. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," vol. 16, pp. 321-357, 2002.

[13] T. Zhu, Y. Lin, and Y. J. P. R. Liu, "Synthetic minority oversampling technique for multiclass imbalance problems," vol. 72, pp. 327-340, 2017.

[14] S. Barua, M. Islam, and K. Murase, "A novel synthetic minority oversampling technique for imbalanced data set learning," in International Conference on Neural Information Processing, 2011, pp. 735-744: Springer.

[15] T. J. S. Fushiki and Computing, "Estimation of prediction error by using K-fold cross-validation," vol. 21, no. 2, pp. 137-146, 2011.

[16] P. Refaeilzadeh, L. Tang, and H. J. E. o. d. s. Liu, "Cross-validation," vol. 5, pp. 532-538, 2009.

[17] G. J. Sawale and S. R. J. I. J. C. S. A. Gupta, "Use of artificial neural network in data mining for weather forecasting," vol. 6, no. 2, pp. 383-387, 2013.

[18] J. N. Liu, B. N. Li, T. S. J. I. T. o. S. Dillon, Man,, and P. C. Cybernetics, "An improved naive Bayesian classifier technique coupled with a novel input solution method [rainfall prediction]," vol. 31, no. 2, pp. 249-256, 2001.

[19] K. Abhishek, A. Kumar, R. Ranjan, and S. Kumar, "A rainfall prediction model using artificial neural network," in 2012 IEEE Control and System Graduate Research Colloquium, 2012, pp. 82-87: IEEE.

[20] N. S. Philip, K. B. J. C. Joseph, and Geosciences, "A neural network tool for analyzing trends in rainfall," vol. 29, no. 2, pp. 215-223, 2003.

[21] C. Wu, K. W. Chau, and C. J. J. o. H. Fan, "Prediction of rainfall time series using modular artificial neural networks coupled with data-preprocessing techniques," vol. 389, no. 1-2, pp. 146-167, 2010.

[22] J. Joseph and T. J. I. J. o. C. A. Ratheesh, "Rainfall prediction using data mining techniques," vol. 83, no. 8, 2013.

[23] A. Grover, A. Kapoor, and E. Horvitz, "A deep hybrid model for weather forecasting," in Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, 2015, pp. 379-386.

[24] S. Zainudin, D. S. Jasim, and A. A. J. I. J. A. S. E. I. T. Bakar, "Comparative analysis of data mining techniques for Malaysian rainfall prediction," vol. 6, no. 6, pp. 1148-1153, 2016.

[25] K. Lu and L. Wang, "A novel non-linear combination model based on support vector machine for rainfall prediction," in 2011 Fourth International Joint Conference on Computational Sciences and Optimization, 2011, pp. 1343-1346: IEEE.

[26] A. Fernández, S. Garcia, F. Herrera, and N. V. J. J. o. a. i. r. Chawla, "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," vol. 61, pp. 863-905, 2018.

[27] P. Skryjomski and B. Krawczyk, "Influence of minority class instance types on SMOTE imbalanced data oversampling," in first international workshop on learning with imbalanced domains: theory and applications, 2017, pp. 7-21: PMLR.

[28] P. Jeatrakul, K. W. Wong, and C. C. Fung, "Classification of imbalanced data by combining the complementary neural network and SMOTE algorithm," in International Conference on Neural Information Processing, 2010, pp. 152-159: Springer.

[29] M. N. Triba et al., "PLS/OPLS models in metabolomics: the impact of permutation of dataset rows on the K-fold cross-validation quality parameters," vol. 11, no. 1, pp. 13-19, 2015.