Journal of Agricultural Science and Technology B 13 (2023)
doi: 10.17265/2161-6264/2023.01.001

DAVID PUBLISHING

Spray Prediction Model for Aonla Rust Disease Using Machine Learning Techniques

Hemant Kumar Singh[1], Bhanu Pratap[1], S. K. Maheshwari[2], Ayushi Gupta[3], Anuradha Chug[3], Amit Prakash Singh[3] and Dinesh Singh[4]

1. Acharya Narendra Deva University of Agriculture and Technology, Ayodhya 224229, U.P., India

2. ICAR-Central Institute for Arid Horticulture, Bikaner 334006, Rajasthan, India

3. University School of Information, Communication & Technology, GGSIPU, New Delhi 110012, India

4. ICAR-Indian Agricultural Research Institute, New Delhi 110012, India

Abstract: Disease prediction in plants has acquired much attention in recent years. Meteorological factors such as: temperature, relative humidity, rainfall, sunshine play an important role in a plan's growth only if they are present in adequate amounts as required by the plant. On the other hand, if the factors are inadequate, they may also support the growth of a disease in the plants. The current study focuses on the Rust disease in Aonla fruits and leaves by utilizing a real time dataset of weather parameters. Fifteen different models are tested for spray prediction on conducive days. Two resampling techniques, random over sampling (ROS) and synthetic minority oversampling technique (SMOTE) have been used to balance the dataset and five different classifiers: support vector machine (SVM), logistic regression (LR), k-nearest neighbor (kNN), decision tree (DT) and random forest (RF) have been used to classify a particular day based on weather conditions as conducive or non-conducive. The classifiers are then evaluated based on four performance metrics: accuracy, precision, recall and F1-score. The results indicate that for imbalanced dataset, kNN is appropriate with high precision and recall values. Considering both balanced and imbalanced dataset models, the proposed model SMOTE-RF performs best among all models with 94.6% accuracy and can be used in a real time application for spray prediction. Hence, timely fungicide spray prediction without over spraying will help in better productivity and will prevent the yield loss due to rust disease in Aonla crop.

Key words: Aonla, Internet of Things, machine learning, plant disease, rust, spray prediction.

1. Introduction

Aonla or Amla (*Emblica officinalis* Garten), popularly known as Indian Gooseberry, is well known for its health benefits. The fruit is an outstanding source of not just vitamin C, A and E, but also iron and calcium. However, Aonla plants—both leaves and fruit, suffer from serious diseases such as rust and soft rot, leading to significant losses in yield. Rust is a severe disease caused by the fungus *Phakopsora phyllanthi* on leaves and by the teliospore *Ravenelia emblicae* on the fruit [1]. It initially appears as small brownish rusty pustules on the fruit and grows into large rings later. On leaves, it develops in the form of pinkish brown pustules. This disease also affects other fruits and crops such as apple, wheat, beans, sugarcane, etc. [2].

The climatic conditions and meteorological factors such as minimum temperature, maximum temperature, rainfall, wind speed, humidity, etc. in adequate amounts may lead to good crop productivity. Although inadequate, they may also contribute to developing dangerous diseases in plants. Due to changes in environmental conditions and increasing global warming every year, the risk of such diseases becomes even higher. As per a report, in Uttar Pradesh, India, where presently gooseberry is grown over 6,000 hectares of land, a decline in Aonla cultivation has been witnessed for almost ten years due to rotting. This has led to grief among the farming community [3].

Corresponding author: Dinesh Singh, Ph.D., research field: plant bacteriology.

Usually, manual experts are required to advise the farmers on the appropriate use of fungicides and pesticides to prevent disease growth. This procedure is costly, and the farmers have to bear huge financial losses due to the non-availability of experts. Also, the excessive use of pesticides or fungicides, even on non-conducive days environments, may lead to further deterioration of the crop. This implies the need for a system that detects the current environmental conditions as conducive or non-conducive so that the use of fungicides for crop protection can be controlled.

A spray prediction model can be implemented, which can send an alarming signal to the farmer indicating to spray the fungicide in case the environmental conditions turn out to be conducive. In this way, the growth of a disease can be prevented along with the retention of fruit quality. In this study, the Aonla Rust dataset of weather parameters collected over 16 years has been used to label the environmental conditions of a day as conducive or non-conducive for the growth of the rust disease. To overcome the problem of imbalanced data, the dataset has been resampled using the random over sampling (ROS) method and synthetic minority oversampling technique (SMOTE). Next, five different machine learning based classifiers—support vector machine (SVM), logistic regression (LR), k-nearest neighbor (kNN), decision tree (DT), and random forest (RF), have been used both on imbalanced and balanced datasets to classify the weather conditions as conducive or non-conducive for the Aonla Rust disease. The performance of the prediction models has been evaluated using four performance measures—accuracy, precision, recall and F1-score. This study will be helpful in finding the best spray prediction model for Aonla Rust dataset that can be used in the real-time application for the spray predictions. If the weather conditions are conducive for a particular day, then an appropriate amount of fungicides can be sprayed over the plant. Otherwise, the unnecessary spray of fungicides can be prevented.

Several studies have worked upon rust disease in different crops using the techniques of remote sensing, hyperspectral imagery, unmanned aerial vehicles (UAV) imagery, machine learning and deep learning. Several studies have also considered weather parameters such as temperature, sunshine, rainfall, humidity or sensor data such as soil moisture, pH, leaf wetness, for early disease predictions in different crops. In this research, only those studies are considered which have either worked on Rust disease or taken weather parameters as input. Some of such studies are listed in Table 1. Apart from these, various studies have focused on plant or leaf images and performed disease prediction using image processing, machine learning

Table 1 Past studies on plant disease prediction model developed by various workers.

Disease	Crop	Dataset type	Classifier/technique	Reference
Yellow rust	Wheat	Reflectance data	Neural networks	[4]
Orange rust	Sugarcane	Spectral vegetation in dices	Hyperspectral imagery	[5]
Powdery mildew, Leaf rust	Wheat	Multi-spectral remote sensing data	Decision tree, Normalized difference vegetation index (NDVI)	[6]
Leaf rust	Wheat	Spectral vegetation in dices	Hyperspectral imagery	[7]
Leaf rust	Wheat	Reflectance data	Partial Least Square Regression (PLSR), v-Support Vector Regression (v-SVR), Gaussian Process Regression (GPR)	[8]
Late blight	Potato	Weather/Sensor data	Support Vector Regression	[9]
Late blight	Potato	Weather/Sensor data	Artificial Neural Network	[10, 11]
Yellow rust	Wheat	UAV images	Deep Convolutional Neural Network (DCNN)	[12]
Powdery mildew	Tomato	Weather/Sensor data	Extreme Learning Machine	[13]
Rice diseases	Rice	Weather/Sensor data	Naive bayes	[14]
Powdery mildew	Tomato	Weather/Sensor data	kNN, decision tree, random forest	[15]

and deep learning techniques. Such studies are out of scope for this research work and hence are omitted.

Although much research has been done in plant disease prediction using weather data, none of the studies focused on Aonla fruit Rust disease. Also, none of the studies has focused on real-time weather dataset of Aonla Rust disease. The study aims to predict based on the weather conditions of a particular day to be conducive or non-conducive for the rust disease growth in Aonla fruit. Since the dataset is highly imbalanced, two resampling techniques, namely ROS and SMOTE have been applied to balance the data. Then five machine learning classifiers, SVM, LR, kNN, DT and RF were utilized for the binary classification task, and their results were compared using four performance metrics to decide which classification model is most appropriate for spray prediction.

The rest of the paper is structured this way: Section 2 provides the dataset details and methods used. The results are shown in Section 3 followed by a discussion in Section 4.

2. Materials and Methods

This section discusses the primary elements of this study and is divided into five subsections. The first subsection 2.1 explains the data collection procedure. The second subsection 2.2 defines the resampling techniques used. The third subsection 2.3 defines the machine learning classifiers utilized. The fourth subsection 2.4 describes the performance metrics used for evaluation. Lastly, the fifth subsection 2.5 discusses the experimental framework for this study.

2.1 Data Collection Procedure

For the real time dataset collection, the Aonla plants of variety NA-7 (Narendra Aonla-7), were planted at Main Experiment Station, Horticulture, Kumarganj, Ayodhya (U.P.) India (Latitude: 26°47′ N, Longitude: 82°12′ E, Altitude: 113 m above mean sea level) in randomized block design. A picture of the Agro-Meteorological Observatory is shown in Fig. 1. The data have been collected over 16 years from 2004-2020 (excluding 2016). The initiation of the rust disease in Aonla took place during the 36th or 37th standard meteorological week. Hence, every year, the data have been collected over 18 weeks—starting from week 35 till week 52. Six weather parameters, namely maximum temperature (°C), minimum temperature (°C), relative humidity (morning %), relative humidity (evening %), rainfall (mm) and sunshine (h/d) have been measured using different equipment which are mentioned in Table 2 and based on these parameters, the disease severity has been computed by the experts. The weekly data have been averaged, resulting in 18 samples each year—total of 288 samples over 16 years. To establish the ground truth, the samples with disease severity value 0 are considered non-conducive and those with a disease severity value greater than 0 have been considered conducive. The resultant dataset has 71 non-conducive and 217 conducive samples, which is highly imbalanced.

2.2 Resampling Techniques

The imbalance in data refers to the non-equal distribution of samples belonging to different classes or categories. In an imbalanced dataset, the class label with fewer samples is called the minority class and the one with a large number of samples is called the majority class. The data imbalance problem may lead to biasness towards the majority class samples resulting in low performance. Therefore, this study uses two known oversampling techniques—ROS and SMOTE to balance the data and are described below.

ROS balances the dataset by increasing the minority class samples to become equal to the majority class samples [16]. This technique randomly copies the minority class samples to increase the data. Hence the model is prone to overfitting due to replication.

SMOTE overcomes the overfitting problem in ROS [17]. This technique creates new synthetic samples of the minority class to balance the dataset. First, a feature vector is located, and its nearest neighbor is identified.

Fig. 1 Agro-meteorological observatory.

Table 2 Equipment for weather data collection.

Parameter	Equipment
Maximum temperature (°C)	Maximum thermometer (mercury)
Minimum temperature (°C)	Minimum thermometer (Alcohol)
Relative humidity morning (%)	Dry bulb thermometer and wet bulb thermometer
Relative humidity evening (%)	Dry bulb thermometer and wet bulb thermometer
Rainfall (mm)	Rain gauge

Then the distance between them is computed and multiplied by a random number between 0 and 1. The resulting distance is the new data point on the line segment. The process is repeated until the dataset is balanced.

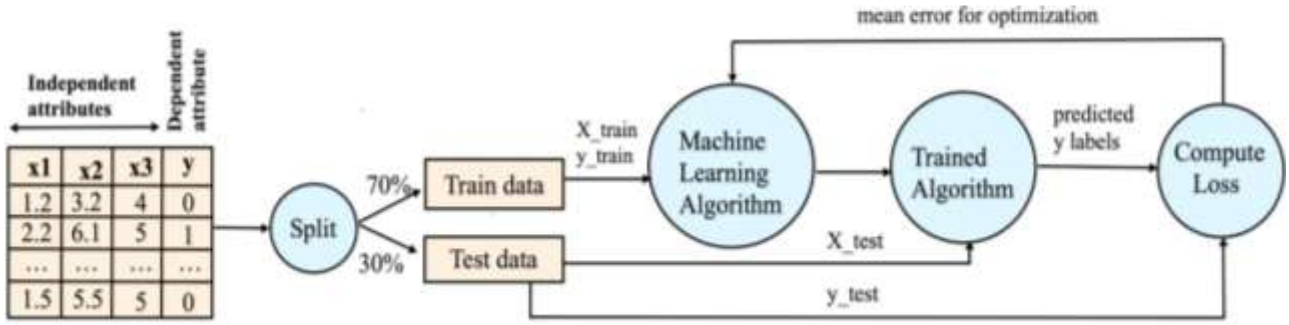
2.3 Machine Learning Classifiers

Machine learning algorithms take the independent and dependent variables as input, learn from them and improve their learning by minimising a loss function. A flow diagram representing the same is shown in Fig. 2. Five different machine learning classifiers have been used on imbalanced and balanced datasets resulting in 15 different models. The technical aspects of the classifiers are explained below.

- SVM: It is a classification or regression algorithm that finds the best hyperplane or decision boundary that separates the data [18]. There can be several valid decision boundaries, but SVM chooses the best by maximizing the margin—the distance of the hyperplane from any of the training examples. Fig. 3 shows a diagrammatic representation of the dataset having two features—relative humidity and temperature and two class labels—conducive (red points) and non-conducive (green points). The dark central line in Fig 3a. represents the optimal hyperplane, and the two parallel lines (lighter ones) represent the marginal lines. The margin width is also indicated. The four points—two green and two red, on the marginal lines are called support vector points. The idea behind seeking a large margin is that the farther the hyperplane is from the support vector points, the better the classification will be for the points lying on the two sides of the plane. That is why this classifier is also called a large margin classifier.

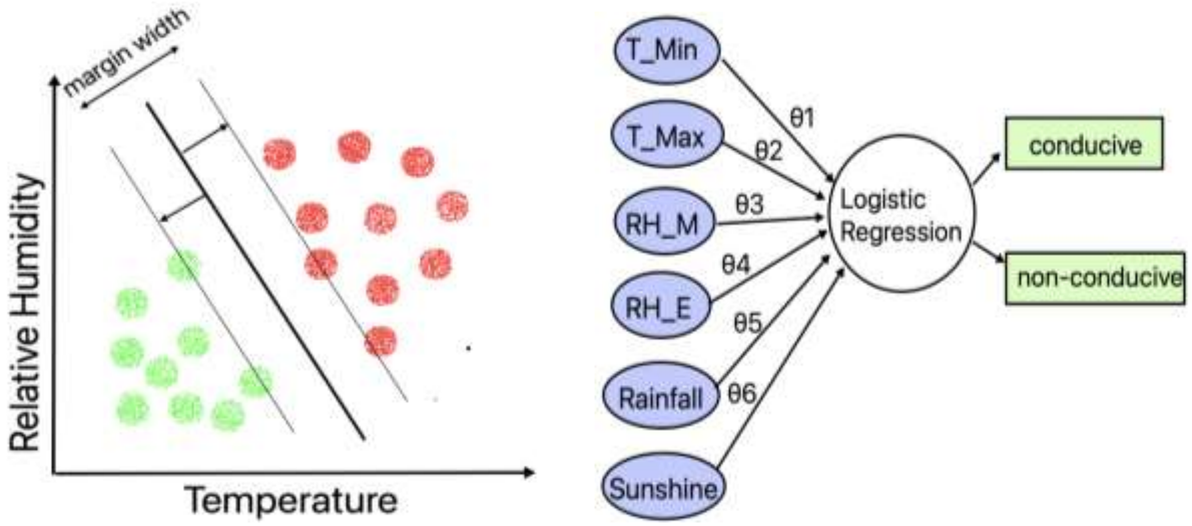
- LR: It is a classification technique initially designed for binary classification tasks but can be extended for multiclass classification problems as well [19]. Let X_{train} denotes the training set independent variables, y_{train} denotes the training set dependent variable, X_{test} denotes the test set independent variables, y_{test} denotes the test set dependent variable and y_{pred} denotes the dependent variable predictions on test set. Then the cost associated with the i^{th} test sample can be defined as,

$$cost(y_{pred}(i), y_{test}(i)) = -y_{test}(i) log(y_{pred}(i)) - (1 - y_{test}(i)) log(1 - y_{pred}(i)) \quad (1)$$



x1	x2	x3	y
1.2	3.2	4	0
2.2	6.1	5	1
...	...	...	...
1.5	5.5	5	0

Fig. 2 Model training to machine learning algorithms



(a) (b)

Fig. 3 SVM and LR.

If the actual label of the i^{th} sample matches the predicted label, i.e., if $y_{test}(i) = y_{pred}(i)$, then the cost would be 0. Otherwise, the cost would be high. Gradient descent is used for the parameter optimization and the sigmoid function is used for the predictions. Fig. 3b shows a diagrammatic representation of the algorithm where θ denotes the parameters of the algorithm.

- kNN: It is a supervised classification or regression algorithm that works on the assumption that samples similar to each other and having the same class label are close to each other [20]. The parameter k denotes the number of closest samples that will be considered for a data point or sample. The algorithm can be explained as follows:

(1) Provide the training set X_{train} and y_{train} to the classifier.

(2) Then for a test set data point i, calculate the distance of i with all the data points in the train set. The distance can be computed using Euclidean or Manhattan distance.

(3) Out of all the data points in the train set, pick k data points or neighbors closest to the data point i in terms of distance measured.

(4) Predict the label for the data point i as the mode of the labels of all the k neighbors selected.

The diagrammatic representation for binary classification with two features—temperature and relative humidity is shown in Fig. 4a. The red circles represent the conducive class, and the green circles represent the non-conducive class. A new data point—white circle, will be classified as non-conducive for k = 3 nearest neighbors since it has 2 non-conducive neighbors and 1 conducive neighbor.

- DT: This technique partitions the input space into regions which are interpretive and easy to visualize [10, 11]. They are non-parametric in nature. Depending on the data, the features are selected which will be split based on a value and the regions will be formed. An example is shown in Fig. 4b where we have three features: temperature (Temp), relative Humidity (RH) and rainfall and two classes: conducive and non-conducive. Let p_{Rk} denote the probability of a point in region R belonging to class k. Then,

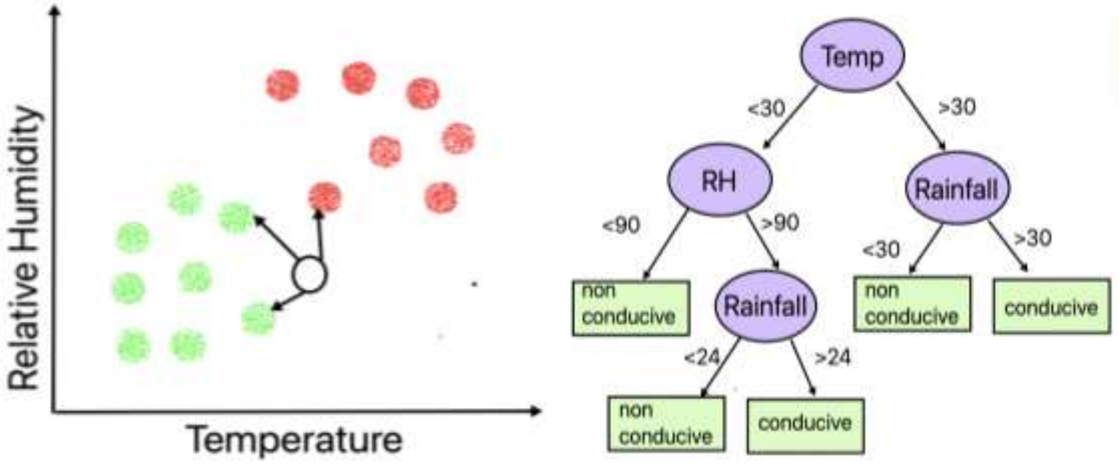
$$p_{Rk} = \frac{1}{|R|}\sum_{x_i \epsilon R} I\{y_i = k\} \tag{2}$$

where $I\{y_i = k\}$ is the identity function which is 1 if $y_i = k$, otherwise 0. Then the misclassification error ME can be written as:

$$ME = \frac{1}{|R|}\sum_{x_i \epsilon R} I\{y_i \neq k_R\} \tag{3}$$

where k_R is the class label of region R and is equal to the label k for which the probability in region R is maximum.

$$k_R = argmax_k p_{Rk} \tag{4}$$



(a) (b)

Fig. 4 kNN and DT.

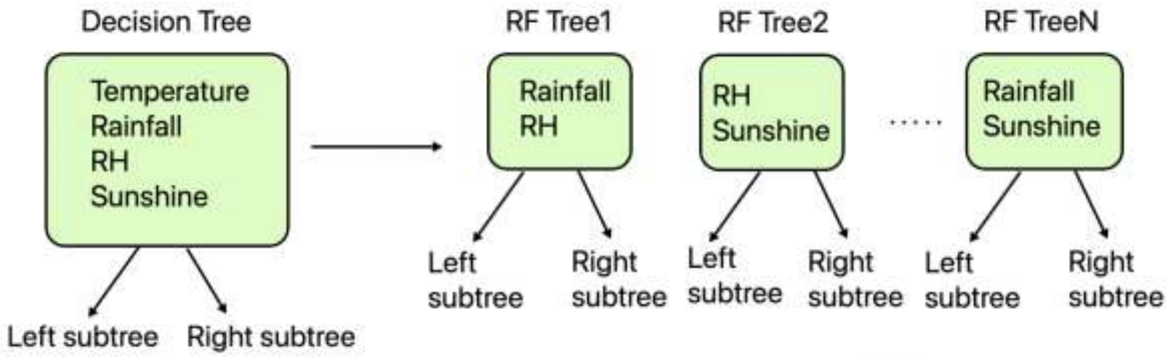


Fig. 5 Random forest.

Table 3 Performance metrics.

Performance metric	Formula
Accuracy	$\frac{TC+TN}{TC+TN+FC+FN}$
Precision	$\frac{TC}{TC+FC}$
Recall	$\frac{TC}{TC+FN}$
F1-score	$2*\frac{Precision*Recall}{Precision+Recall}$

Table 4 Dataset transformation after resampling.

Class label	Data points before resampling	Data points after ROS	Data points after SMOTE
Non-conducive	71	217	217
Conducive	217	217	217

- RF: This technique follows the approach of bagging with Decision Trees [21]. Different samples are taken from the dataset with replacement and different DTs are built. But since the trees are built using the same dataset, the correlation between the trees would be high. In order to reduce the correlation, the trees are built with randomly selected *t* features out of *n* features for each sample taken. The final result is calculated using the majority voting rule. Hence, Random Forest gives better results by reducing the variance. The diagrammatic representation is shown in Fig. 5.

2.4 Performance Evaluation Metrics

To evaluate the performance of the above-mentioned classifiers, four different evaluation metrics have been used and explained in Table 3. TC denotes the number of conducive class samples correctly classified as conducive, *TN* denotes the number of non-conducive class samples correctly classified as non-conducive, *FC* denotes the number of non-conducive class samples falsely classified as conducive, and *FN* denotes the number of conducive class samples falsely classified as non-conducive.

2.5 Experimental Framework

The proposed methodology is shown in Fig. 6. Initially, the imbalanced Aonla dataset was pre-processed with resampling techniques—ROS and SMOTE to get a balanced dataset. The changes in the dataset after resampling have been shown in Table 4. After that, both the balanced and imbalanced datasets have been split such that 70% goes to training data and 30% to testing data. The datasets are then classified using five classifiers—SVM, LR, kNN, DT and RF resulting in 15 different models—5 for the imbalanced dataset and 10 for the balanced dataset, namely—SVM,

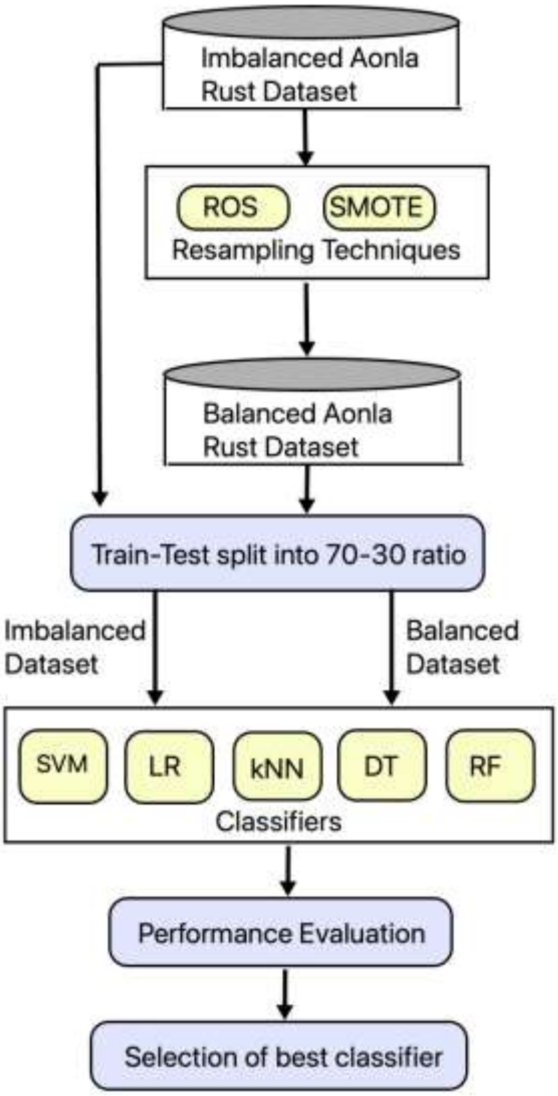

Imbalanced Aonla Rust Dataset
ROS
SMOTE
Resampling Techniques
Balanced Aonla Rust Dataset
Train-Test split into 70-30 ratio
Imbalanced Dataset
Balanced Dataset
SVM
LR
kNN
DT
RF
Classifiers
Performance Evaluation
Selection of best classifier

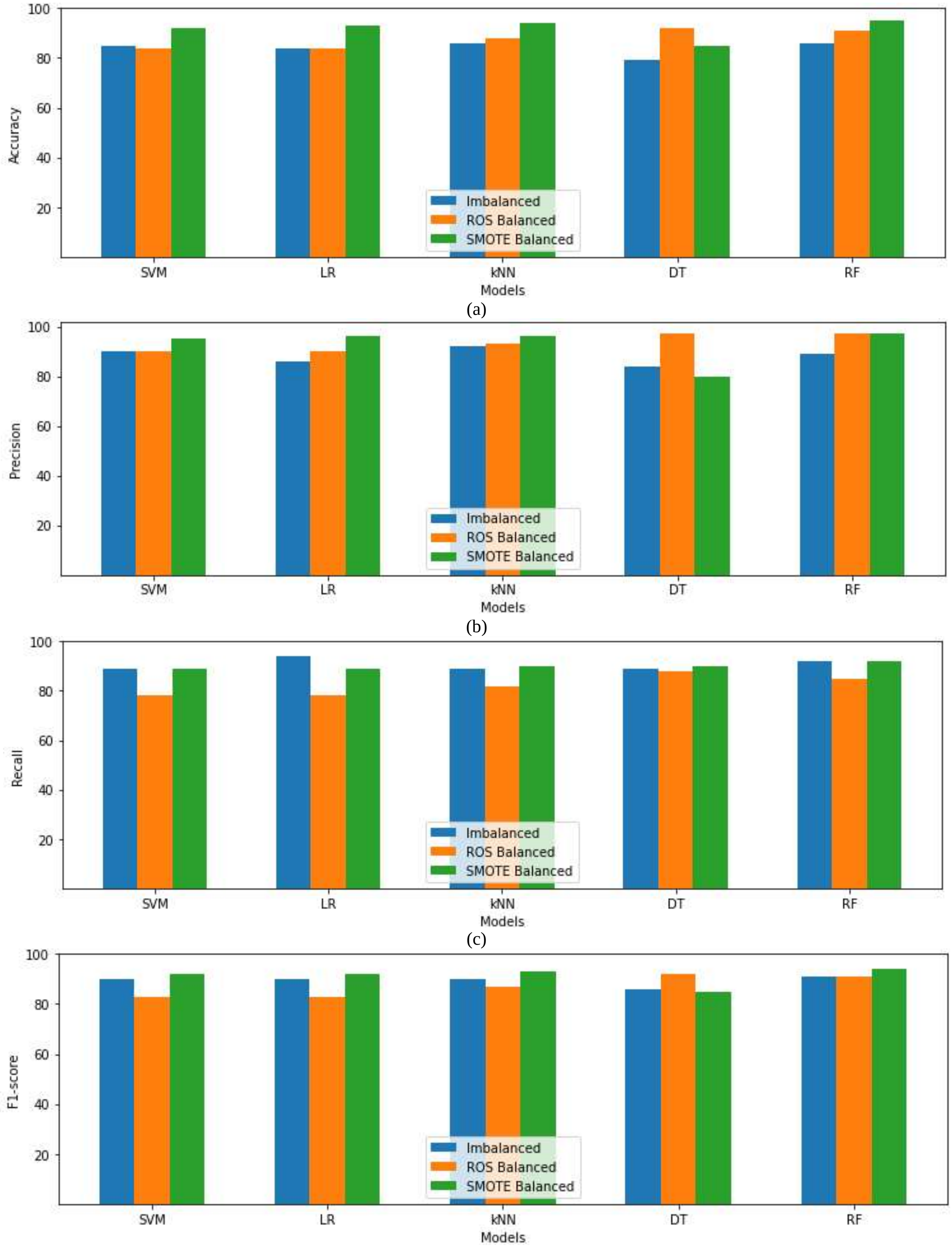
Fig. 6 Research methodology.

LR, kNN, DT, RF, ROS-SVM, ROS-LR, ROS-kNN, ROS-DT, ROS-RF, SMOTE-SVM, SMOTE-LR, SMOTE-kNN, SMOTE-DT and SMOTE-RF. For the SVM algorithm, linear kernel has been chosen since the dataset is small. For the LR, DT and RF algorithms, the parameter random state has been given the value 1 to produce the same results across different calls, and the rest of the parameters have taken the default values. For the kNN algorithm, the best value for parameter *k* has been chosen in the range [1, 16] using GridSearch optimization. For all the other parameters, the default values have been taken. The estimates have been taken over 20 iterations, and the average values are taken as the final result. After classification, all the 15 models are evaluated using performance metrics—Accuracy, Precision, Recall and F1-Score. Accuracy captures the fraction of the number of correct predictions to the total number of data points predicted [22]. However, in the case of highly imbalanced datasets, for example—out of 100 samples, there are 95 conducive and only 5 non-conducive, if the model classifies all the 100 examples as conducive, Accuracy would be 95%. Hence it will not be able to capture the importance of non-conducive class data points. Here, Precision comes into the picture. It captures the fraction of correct conducive predictions to the total number conducive predictions by the model [22]. If the model classifies a single data point correctly as conducive, Precision turns out to be 1, which will not be helpful for measuring the performance. Recall captures the fraction of correct conducive predictions to the total number of data points that are actually conducive in the dataset [22]. Again, if all the data points are classified as conducive, Recall goes to 1. Hence a balance between Precision and Recall values is necessary. Therefore F1-Score is considered, which takes both Precision and Recall and computes their harmonic score. Finally, based on these metrics, the best model is selected.

All the implementation has been done using the scikit-learn library in python, and the results are plotted using the matplotlib library in python. The implementations were executed on macOS Big Sur Version 11.3.1 with 8 GB RAM.

3. Results and Discussion

This section presents the results of all 15 models—SVM, LR, kNN, DT, RF, ROS-SVM, ROS-LR, ROS-kNN, ROS DT, ROS-RF, SMOTE-SVM, SMOTE-LR, SMOTE-kNN, SMOTE-DT and SMOTE-RF. A comparison of these models is made based on the four performance metrics explained above and shown in Fig. 7. In each Figs 7a-7d., the blue-colored bars represent the imbalanced models, the orange-colored bars represent the ROS balanced models, and the green-colored bars



(a)

(b)

(c)

(d)

Fig. 7 Evaluation results of the models.

represent the SMOTE balanced models. The following observations have been made:

- Considering only ROS balanced models (the orange bars), ROS-DT performs best among all with the highest accuracy (92.3%), precision (96.7%), recall (88%) and F1-score (92.1%).
- If Accuracy is considered on imbalanced datasets, kNN and RF perform the best among all, with a value 86.2%. Although, the balanced datasets should be looked upon since accuracy is inappropriate for imbalanced datasets. In Fig. 7a, the SMOTE balanced models perform better than the ROS balanced models in majority of cases—SVM, LR, kNN, RF. Only in the case of DT, ROS-DT performs better than SMOTE-DT. Also, SMOTE balanced RF model—SMOTE-RF performs best among all with 94.6% accuracy.
- In Figs. 7b and 7c, if only imbalanced dataset models (the blue bars) are considered, then again, kNN performs the best with a balance in precision (91.9%) and recall (89%) values. Also, LR seems to have the highest recall (93.7%) but a low precision (85.7%). This indicates that the model is able to classify a large number of conducive data points correctly, but it is misclassifying many non-conducive data points as conducive, which may lead to fungicide over spraying. Hence a balance between precision and recall is required.
- In Figs. 7b and 7c, considering both balanced as well as imbalanced dataset models, all ROS balanced models have better Precision values (ROS-DT and ROS-RF having 96.7% and 96.6% Precision) than the imbalanced models but these models have a low Recall when compared to both imbalanced models as well as SMOTE balanced models. This implies that these models are able to correctly classify a subset of conducive samples as conducive but are misclassifying a lot of conducive samples as non-conducive. This will lead to non-spraying of fungicide, and the disease will grow. However, the model SMOTE-RF gives comparable performance both in case of Precision (96.5%) and Recall (91.8%), again proving to be the best among all.
- Considering F1-Score of imbalanced dataset models, kNN and RF give comparable performance (90.4% and 90.7%). Also, four of the SMOTE balanced models performed better than the imbalanced models and ROS balanced models (except ROS-DT). Among all, the model SMOTE-RF performs best with an F1-Score of 94.1%.

These observations imply that the model kNN is appropriate for an imbalanced dataset with a balance in precision (91.9%) and recall (89%) values. Also, SMOTE-RF turns out to be the best model with 94.6% accuracy among all models and shall be recommended for the present study. This is because SMOTE resampling technique overcomes the overfitting limitation of ROS technique and RF classifier uses bagging of DTs to create different trees with different features, thus reducing the variance in the results. Hence, the model can be fairly used in real time scenarios for the fungicide spray prediction based on the weather conditions as conducive or non-conducive for the growth of Aonla Rust disease on a particular day. As far as we know, there are no previous research works that have worked upon Aonla Rust disease using machine learning techniques. Hence a comparison cannot be made.

4. Conclusion

In this research study, the authors have used two resampling techniques—ROS and SMOTE along with five machine learning classifiers—SVM, LR, kNN, DT and RF to develop 15 different models—5 on imbalanced data and 10 on balanced data, for spray prediction in Aonla plants to prevent rust disease. It has been found that the SMOTE balanced data with RF classifier, SMOTE-RF has turned out to be the best model among all with 94.6% accuracy. If the imbalanced dataset is considered, then the model kNN turns out to be the best among the five imbalanced dataset models with a precision of 91.9% and Recall value of 89%. Hence these models can be used to predict whether the weather conditions of a particular day—minimum temperature, maximum temperature, morning relative humidity, evening relative humidity, rainfall and sunshine hours, are conducive or non-conducive for the growth of Rust disease in Aonla plants. Accordingly, the appropriate fungicide can be sprayed only on conducive days preventing the overuse of fungicides which may otherwise lead to the degradation of fruit quality.

In future, a mobile-based application can be developed for spray prediction in Aonla plants based on weather conditions. Another model can also be developed which will predict the disease severity as not severe, low, moderate or high, based on weather conditions, and the use of fungicides then can even be reduced on low severity days as compared to moderate or high severity days. Disease forecasting models based on time-series data can also be explored. Further, several deep learning models can also be applied and compared them for better performance.

Acknowledgement

Authors are thankful to Director, Central Institute of Arid Horticulture, Bikaner (Rajasthan) and Director Research, Acharya Narendra Deva University of Agriculture and Technology, Kumarganj, Ayodhya (Uttar Pradesh) for providing the necessary facilities during the course of investigation.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflict of Interest

All authors declare that they have no conflicts of interest.

References

[1] Jarial, K., Banyal, S., Mandradia, R., and Sharma, S. 2011. "Occurrence of Aonla Leaf Rust Caused by *Phakopsora phyllanthi* in Himachal Pradesh." *Journal of Mycology and Plant Pathology* 41 (2): 319.

[2] The Editors of Encyclopaedia. 2021. Accessed on September 23, 2021. https://www.britannica.com/science/rust.

[3] Hub, H. N. 2021. Accessed on November 29, 2021. https://hindustannewshub.com/india-news/disease-increased-trouble-in-amla-farmers-are-suffering-due-to-pestalotia-cruenta-in-this-district-of-up/.

[4] Moshou, D., Bravo, C., West, J., Wahlen, S., McCartney, A., and Ramon, H. 2004. "Automatic Detection of 'Yellow Rust' in Wheat Using Reflectance Measurements and Neural Networks." *Computers and Electronics in Agriculture* 44 (3): 173-88.

[5] Apan, A., Held, A., Phinn, S., and Markley, J. 2004. "Detecting Sugarcane 'Orange Rust' Disease Using EO-1 Hyperion Hyper-Spectral Imagery." *International Journal of Remote Sensing* 25 (2): 489-98.

[6] Franke, J., and Menz, G. 2007. "Multi-temporal Wheat Disease Detection by Multi-spectral Remote Sensing." *Precision Agriculture* 8 (3): 161-72.

[7] Ashourloo, D., Mobasheri, M. R., and Huete, A. 2014. "Evaluating the Effect of Different Wheat Rust Disease Symptoms on Vegetation Indices Using Hyperspectral Measurements." *Remote Sensing* 6 (6): 5107-23.

[8] Ashourloo, D., Aghighi, H., Matkan, A. A., Mobasheri, M. R., and Rad, A. M. 2016. "An Investigation into Machine Learning Regression Techniques for the Leaf Rust Disease Detection Using Hyperspectral Measurement." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9 (9): 4344-51.

[9] Gu, Y., Yoo, S., Park, C., Kim, Y. H., Park, S., Kim, J. S., and Lim, J. 2016. "Blite-SVR: New Forecasting Model for Late Blight on Potato Using Support-Vector Regression." *Computers and Electronics in Agriculture* 130: 169-76.

[10] Quinlan, J. R. 2018. "Simplifying Decision Trees." *International Journal of Man-Machine Studies* 27 (3): 221-34.

[11] Sharma, P., Singh, B., and Singh, R. 1987. "Prediction of Potato Late Blight Disease Based Upon Weather Parameters Using Artificial Neural Network Approach." In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1-13. New York: IEEE.

[12] Zhang, X., Han, L., Dong, Y., Shi, Y., Huang, W., Han, L., Gonz´alez-Moreno, P., Ma, H., Ye, H., and Sobeih, T. 2019. "A Deep Learning-Based Approach for Automated Yellow Rust Disease Detection from High-Resolution Hyperspectral UAV Images." *Remote Sensing* 11 (13): 1554.

[13] Bhatia, A., Chug, A., and Prakash Singh, A. 2020. "Application of Extreme Learning Machine in Plant Disease Prediction for Highly Imbalanced Dataset." *Journal of Statistics and Management Systems* 23 (6):

1059-68.

[14] Maneesha, A., Suresh, C., and Kiranmayee, B. 2021. “Prediction of Rice Plant Diseases Based on Soil and Weather Conditions.” In *Proceedings of International Conference on Advances in Computer Engineering and Communication Systems*. New York: Springer, pp. 155-65.

[15] Bhatia, A., Chug, A., Singh, A. P., Singh, R. P., and Singh, D. 2022. “A Forecasting Technique for Powdery Mildew Disease Prediction in Tomato Plants.” In *Proceedings of Second Doctoral Symposium on Computational Intelligence.* New York: Springer, pp. 509-20.

[16] Ling, C. X., and Li, C. 1998. “Data Mining for Direct Marketing: Problems and Solutions.” *In Kdd* 98: 73-9.

[17] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. 2002. “Smote: Synthetic Minority Over-Sampling Technique.” *Journal of Artificial Intelligence Research* 16: 321-57.

[18] Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B. 1998. “Support Vector Machines.” *IEEE Intelligent Systems and Their Applications* 13 (4): 18-28.

[19] Wright, R. E. 1995. “Logistic Regression.” In *Reading and Understanding Multivariate Statistics*. Washington DC: American Psychological Association, pp. 217-44.

[20] Cover, T., and Hart, P. 1967. “Nearest Neighbor Pattern Classification.” *IEEE Transactions on Information Theory* 13 (1): 21-7.

[21] Breiman, L. 2001. “Random Forests.” *Machine Learning* 45 (1): 5-32.

[22] Kumar, S. 2020. “Metrics to Evaluate Machine Learning Algorithms.” Accessed on November 12, 2020. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234.