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

A Comparative Performance Analysis for detection of Red-Rot of Sugarcane

Rahul Kumar[1], Rajeev Paulus[2], Bireshwer Dass Mazumdar[3]

Submitted: 28/01/2024 **Revised:** 06/03/2024 **Accepted:** 14/03/2024

Abstract: Red-rot disease caused by Colletotrichum falcatum is a significant threat for substantial economic losses in sugarcane industry worldwide. Early and accurate detection of disease is crucial for implementing timely control measures for substantial cultivation. This study presents a comparative performance analysis among various leverage machine learning approaches to effectively detect red-rot infections. The research rigorously evaluates the performance of each technique based on accuracy, precision, recall, F-measure, and other relevant error analysis. The data collection, preprocessing, and feature extraction methodologies are meticulously implemented to ensure the credibility and generalizability of the findings. The study's outcomes hold significant implications for precision agriculture and sustainable farming practices. The aim of this study is to implement accurate and efficient method for red-rot disease detection, which can empower farmers to enable targeted intervention to minimize crop losses and reliance on chemical treatments, which will contribute in global movement towards eco-friendly agriculture and enhancement of sugarcane industry. The results of study shed light on the strengths and limitations of each machine learning technique, aiding researchers and practitioners in selecting the most suitable approach for Red Rot detection.

Keywords: *Red-Rot, Machine Learning (ML), Disease Detection, Ensemble Learning, Xgboost*

1. Introduction

Red Rot disease, caused by the fungus Colletotrichum falcatum, remains one of the most devastating threats to the global sugarcane industry. The disease significantly reduces yield, compromises sugar quality, and incurs substantial economic losses for farmers and sugar mills alike. Early detection and accurate diagnosis of Red Rot infections are crucial for effective disease management and the implementation of timely control measures [1]. In recent years, machine learning has emerged as a powerful tool in precision agriculture, offering the potential to revolutionize disease detection and enable proactive intervention. This study aims to conduct a comparative performance analysis of various machine learning methods for the detection of Red Rot of sugarcane. Leveraging the capabilities of machine learning algorithms, such as deep learning, ensemble methods, and classical statistical techniques, the research endeavours to identify the most effective and accurate approach to detecting Red Rot infections. The study's outcomes have far-reaching implications for the agricultural industry, as they can facilitate early detection and targeted treatments, reduce the reliance on chemical inputs, and contribute to sustainable farming practices.

By utilizing machine learning methods, researchers can analyse digital images of sugarcane plants to detect the presence of Red Rot infection. Each machine learning technique brings unique advantages to the task, such as the ability to handle complex patterns, handle high-dimensional data, or provide interpretable results. Through a rigorous and systematic analysis, this study aims to compare the performance of these techniques in terms of accuracy. The significance of this research lies in its potential to optimize disease detection systems and enhance agricultural resilience. An accurate and efficient method for Red Rot detection will enable farmers to swiftly identify infected plants, take targeted actions, and minimize crop losses. By reducing chemical usage through early detection, the research aligns with the global movement towards sustainable agriculture and eco-friendly farming practices.

In the subsequent sections, we will delve into the methodologies employed for data collection, pre-processing, and feature extraction. The performance of each machine learning algorithm will be meticulously evaluated, and the results will be presented and discussed in detail. Ultimately, the study aims to provide valuable insights into the most effective approach to detecting Red Rot of sugarcane using machine learning, paving the way for practical applications in precision agriculture and contributing to the sustainability of the global sugarcane industry.

2. Materials And Method

2.1. Red-Rot Dataset

The primary goal of Red-Rot dataset aims to identify more sensitive and accurate methods to diagnose RR `early and to mark RR's progress with biomarkers. RR data is used to collect various image attributes of sugarcane leaf. A total of 300 records are selected at random from the entire set ensuring that no chosen record has any missing values. Some attributes with no relevance to evaluation are omitted. The rest of the data was normalized.

2.2. Machine Learning Methods

Machine learning is a collection of statistical models and algorithms that allows software applications to perform a specific task without being explicitly programmed. The machine learning methods help to build an adaptive program that automatically adjusts to receive data.[2] The integration of machine learning algorithms offers a transformative approach to combat the pernicious Red Rot disease, leading to improved disease management and sustainable farming practices. The robust

1 Sam Higginbottom University of Agriculture Technology and sciences, Naini, Prayagraj, India-211007
2 Sam Higginbottom University of Agriculture Technology and sciences, Naini, Prayagraj, India-211007
3School of Computer Science Engineering and Technology, Bennett University, Greater Noida, India
**rahkam@gmail.com*
** Corresponding Author Email: author@email.com*

methodology, comparative analysis, and real-world applications make this research a pioneering effort in harnessing machine learning for agricultural challenges. As the agriculture industry continues to evolve, this study sets a remarkable precedent for harnessing cutting-edge technologies to protect and enhance crop production on a global scale. The brief detail of models used in the present study is as follows:

2.2.1. Adaboost

Boosting is a general and effective method for developing accurate prediction rule by merging many fairly susceptible and inaccurate rules. By harnessing the power of Adaboost, the research offers a transformative and efficient approach to combat the devastating Red Rot disease in sugarcane crops. The root of Boosting lies in the theoretical framework of machine learning known as probably approximately correct (PAC) learning model. The AdaBoost by Freund and Schapire was the first realistic boosting algorithm and is one of the most extensively used method, having purposes in different fields. It overcomes many issues of the previous boosting algorithms [3]. This algorithm takes the training set S= $((x_1,y_1)$ ……., $(x_m,y_m))$ of N samples as input, where each instance xi is extracted from certain space X and represented in a vector, and y_i ∈Y is the class label linked with xi [4]. The boosting algorithm additionally can access another unspecified learning algorithm, called a weak learning algorithm represented as Weak Learn. This algorithm repeatedly calls Weak Learning a series of rounds. In round t, the booster runs Weak Learn with distribution Dt on the training set S. To correctly classify a part of the training set with a high probability on D_t,Weak Learn calculates the hypothesis $h_t : X \to Y$. Adaboost is a powerful ensemble learning technique that combines weak classifiers to build a robust and accurate model.

We integrate Adaboost to create a predictive model capable of distinguishing between healthy and infected sugarcane plants based on key image features [5]. This approach ensures a high level of accuracy and robustness in Red Rot disease detection, enabling farmers to take timely actions to mitigate its impact.

2.2.2. Bayesian Network

Bayesian classification depends on Bayes Theorem. It considers the observed data and provides the basis for probabilistic learning which comprises previous knowledge. By harnessing the power of Bayesian Network, the research offers an efficient and accurate approach to combat the devastating Red Rot disease in sugarcane crops. Bayesian Network has been developed to remove the shortcomings of Naïve Bayes classifier. A Bayesian network consists of a directed acyclic graph with set of nodes and a set of edges between nodes. A node denotes the random data and the edges between two nodes represent conditional dependencies between nodes [6]. The Bayesian network learning process is divided into two steps: learning the network and learning the relationship between data.

Firstly, Take domain variable and divided into set of attributes,

X={X1,X2,……,Xn} (1) and C, class variable.

Then, find the value of attribute X with given C:

By using eq. (1), compute predictive distribution P(C, x/7) by marginalizing joint distribution with example x,

$$P(S) = \frac{P(S)}{P(S)} \alpha P(S) \qquad (2)$$

Apply Bayesian Network classification rule using eq. (2)

$$C^* = h_{BNC}(x) = \mathrm{argmax}_{j=1....m} P(x, c_j | S, \theta_s) \qquad (3)$$

Computing $P(x, c_j)$ for each class c_j by joint probability distribution with eq. (3)

$$P(S, \theta_s) = \prod_{i=1}^{n} P(pa(X_i)).P(pa(C)) \qquad (4)$$

The use of Bayesian Network for disease detection is a notable strength of this research. Bayesian Networks are adept at modelling probabilistic relationships between variables, making them highly suitable for complex systems like agricultural diseases.[7] We implement Bayesian Network to create a predictive model capable of identifying Red Rot infections based on various input features. This approach enhances the accuracy and reliability of the disease detection system, empowering farmers with a powerful tool to tackle Red-Rot proactively.

2.2.3. Multilayer-Perceptron (MLP)

MLP by M. Minsky and S.Papert is a feed-forward artificial neural network, composed of more than one perceptron and uses back-propagation for training the network [30]. By harnessing the capabilities of deep learning with MLP, the research offers an efficient and accurate approach to combat the devastating Red Rot disease in sugarcane crops. An MLP consist an input layer for receiving signals, an output layer that makes decisions or predictions for the input values, and number of hidden layers between those input-output layers, which are the real computational engines of the MLP [8]. An MLP with a hidden layer can approximate any continuous function. MLPs are trained on a set of input- output pairs and learn to model the dependencies between input and outputs. It learns how to convert input data to the desired response. MLP calculates a discontinuous function with eq. (5)

$$\vec{x} \mapsto f_{log}(w_0 + (\vec{w}, \vec{x}))$$

where, $f_{loglog}(z) = \frac{1}{1+e^{-z}}$

The use of Multi-Layer Perceptron for disease detection is a standout feature of this research. MLPs are known for their ability to learn complex patterns and relationships within data, making them particularly suitable for intricate problems like Red Rot identification [9]. By adopting this advanced machine learning technique, we create a predictive model capable of accurately distinguishing between healthy and infected sugarcane plants based on key image features. This approach provides an efficient and robust solution to early detection, empowering farmers to take timely actions and manage the disease effectively.

2.2.4. Random Subspace method (RSM)

RSM is an ensemble method that combines several models for classification proposed by Ho [10]. By harnessing the capabilities of ensemble learning with RSM, the research offers an efficient and accurate approach to combat the devastating Red Rot disease in sugarcane crops. RSM tries to reduce the correlation with random samples of features instead of the entire feature set like bagging. Training data can also be modified in RSM and this modification can be done in feature space. Let training sample set S = $(S_1, S_2, ……., S_n)$ and each training object S_i (i = 1…….n) be a p-dimensional vector.[11] We can randomly select r <p attributes from the p-dimensional data set S in RSM. Thus, r-dimensional random subspace can be obtained from p-dimensional feature space. Therefore, modified training set $S^{\sim b} = (S^{\sim b}_1, S^{\sim b}_2, . . ., S^b_n)$ consists of r-dimensional training objects $S^{\sim b}_i = (s^b_{i1}, s^b_{i2}, . . ., s^b_{ir})$

$(i = 1, . . ., n)$, where r components s^{b}_{ij} $(j = 1, . . ., r)$ are randomly selected from p components s_{ij} $(j = 1, . . ., p)$ of the training vector S_i (the choice of each training vector is same). Then we can construct classifiers in random subspaces $S^{\sim b}$ and combines them in a final decision rule by simple majority voting. This approach introduces diversity and robustness to the predictive model, making it adept at handling complex and high-dimensional datasets [12]. By adopting RSM, we create a sophisticated predictive model capable of accurately identifying Red Rot infections based on crucial image features [13]. This methodology provides a reliable and efficient means of early detection, empowering farmers to take proactive actions to manage the disease effectively.

2.2.5. Bagging

Bagging proposed by Breiman is an ensemble method based on the concept of bootstrap and aggregation, so it combines the advantages of both methods [14]. In this method, one tries to combine the predictions from multiple models together to perform better than the original model. Bootstrap is a general method that may be used to minimize the variance for those algorithms that have high variance. This method is powerful because of enhancing the performance of a single model by means of use of more than one copies of it on different sets of data. Bootstrapping consists of random sampling with replacement. However, in order to replicate a bootstrap, $X^{b} = (X^{b}_{1}, X^{b}_{2,}. . . X^{b}_{n})$ of the training set X, it is possible to reduce or even avoid the deceptive training objects in the boot strapping set.[15] Therefore, classifiers that are constructed on these training sets can have improved performance. In general, combined classifiers provide better results than a single classifier because the advantages of each classifier are combined in the last solution. Thus, bagging may help to construct better classifiers on trained misleading sample sets. The utilization of Bagging for disease detection is a noteworthy highlight of this research. Bagging is a powerful learning technique that aggregates predictions from multiple models, reducing the risk of overfitting and enhancing the overall accuracy and robustness of the predictive model. By employing Bagging, we create an effective and accurate predictive model capable of identifying Red Rot infections based on crucial image features [16].

2.2.6. Random Forest

Random forest introduced by Ho is an ensemble method that functions by establishing a multitude of decision trees during training and then gives output class which is the classification or regression mode of the individual trees [17]. The tree is built independently by the general technique of bagging and is randomly selected set of training samples. The final outcome can be determined by voting from all the trees with majority prediction. It turns out that RF is a highly accurate algorithm in various fields such as remote sensing, medical diagnosis, and anomaly detection [18]. RF uses bagging to increase tree diversity by developing trees from different training datasets, thereby reducing overall variance of the model. We adeptly implement Random Forest to develop an effective and accurate predictive model capable of identifying Red Rot infections based on key image features. This approach empowers farmers with a reliable tool for early detection, enabling prompt intervention and targeted treatment to manage the disease effectively [19].

2.2.7. Logistic Regression

Logistic Regression is a classification technique for datasets where dependent variable is dichotomous (binary). It takes only two values 0 and 1 to predict an outcome's probability [20]. A logistic sigmoid function f(x) is employed to transform its output to return a probability value. This value, in turn, is then mapped with respect to a minimum of two classes.

$$f(x) = \frac{1}{1+e^{-x}} \qquad (6)$$

Each feature contributes in predicting the expected outcome of a dataset. Maximum likelihood estimation statistics are used to measure the predictive power of each attribute. The logistic model uses the input dataset to calculate the probability of predicting binary results [21].

2.2.8. Extreme Gradient Boosting

Xgboost is a machine learning method proposed by Chen in 2016 and has been widely used in various data mining fields, especially in kaggle [22]. It is a regression tree whose decision rules are same as the decision tree. Since Xgboost's block structure supports parallelization of the tree structure, it is an effective implementation of the gradient enhanced decision tree (GBDT) [23]. In GBDT, gradient boosting refers to an integrated technique used to create a new model to predict the residuals or errors of previous models and make a final decision by aggregating the predictions of all models. In this paper, Xgboost is applied with the following input parameters, such as logistic regression for classification as objective function, 3 as tree's maximum depth, 0.3 as step shrinkage size, and 1000 as maximum number of iterations.

3. Methodology

In this study, the computation has been done on RR dataset for a case based on data pre-processing and three different sets based on data-partitioning. Firstly, the normalized data is split into train and test data in the ratio of 2:1. Then, the training dataset is used to train ML methods and test dataset is used for validation. Secondly, the normalized data is used to train the ML models which was then subjected to 10-fold cross validation (Case 2) for testing. Thirdly, feature selection is done to remove variables that are highly correlated. In this paper, Boruta feature selection (BFS) method is applied to the normalized data at the pre-processing stage to extract relevant features. It is easy to use feature selection methods which select features using permutation of various features. BFS employs to effectually shortlist the best independent variables out of the total 24 features. After applying BFS approach on the dataset, six features are discarded, and rest of the features are considered to train the model. The pre-processed dataset was then used to train the computing models and the validation was done using 10-fold analysis (Case 3). A total of seven machine learning methods including AdaBoost, Bayes Network, Logistic Regression, Random Sub-Space, MLP, Random Forest and Bagging are used for the classification. The computations are implemented on a 64-bit system with 4GB RAM and Windows 10 operating system using Weka 3.8 software. The performances of various ML models were analysed using Accuracy and Error obtained for all the three cases. Fig. 1 depicts the workflow process for case wise analysis.

Further, the computing models are applied to the individual datasets of 3 different sets from RR database to evaluate accuracy and error analysis. The data from these sets are analyzed to check the homogeneity of data across the sets. The training dataset consisted of 70% records and test dataset had remaining 30% records from each of three sets. Fig. 2 depicts the workflow process for set wise analysis.

4. Performance Analysis

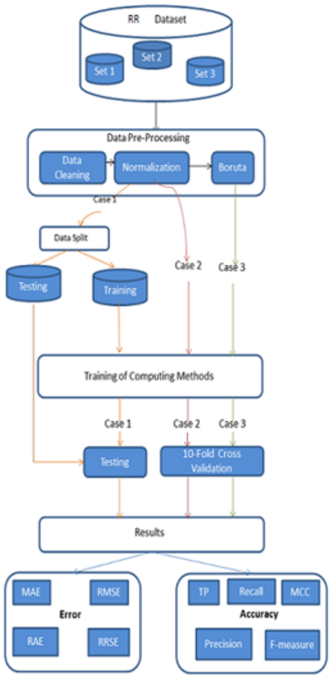

Fig 1. Work process Flow for Case wise data analysis

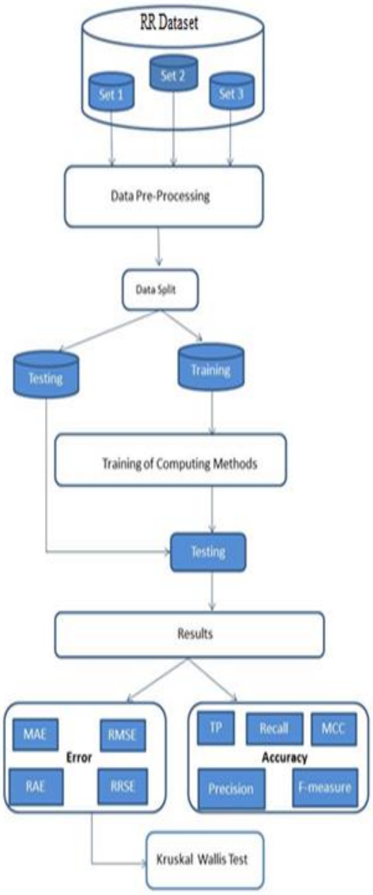

Fig 2. Workflow process for Set wise data analysis

The performance of the proposed method has been evaluated using various accuracy parameters, error parameters and cross-validation. Then, to check the dependency of the data from different sets, Kruskal Wallis test is deployed. There are various accuracy and error parameters used to evaluate the model which are defined in Fig. 3 and Fig. 4 respectively. In Fig. 3, TP depicts True Positive, FP depicts False Positive, TN depicts True Negative and FN depicts False Negative and MCC depicts Matthews Correlation Coefficient. In Fig. 4, MAE depicts Mean Absolute Error, RMSE depicts root mean square error, RRSE depicts root relative squared error and RAE depicts relative absolute error.

Kruskal Wallis Test

The Kruskal Wallis test is described as a distribution-free test which is beneficial for finding the differences among various groups of an independent variable on ordinal as well as continuous dependent variable [24]. When the assumptions of one-way analysis of variance (ANOVA) are not met, this test is taken into consideration. Kruskal Wallis test is used to check the dependency of the data from different sets of databases. This test determines whether the medians of two or more groups are different. The test will tell if there is a significant difference between groups. This test can be used for both continuous and ordinal level dependent variables [25].

5. Result Analysis

The 8ML methods have been efficiently trained on RR dataset. In this study, the efficient data partitioning has been carried out, since data plays an essential role in the training process model. The results produced by case wise, set wise are analyzed in this section.

5.1. Case wise Accuracy and Error Analysis

The following observations are made for the cases to measure the performance of various computing models using accuracy and error parameters:

Case 1 (Validation using test Dataset)

In this case, we randomly select 60% of data as training set and remaining 40% as testing set. Table 1 shows the classification results of various techniques in terms of accuracy and error measures for case 1.

The results indicate that Bagging performs best among other machine learning methods with respect to both accuracy and error parameters whereas Adaboost shows worst performance as shown in Fig 5.

Case 2 (Validation Using 10- fold cross validation)

In this case, all the data is used to train the models and then validation of all the models is done with 10-fold cross-validation. Table 3 shows the classification results of various techniques in terms of accuracy and error measures for case 2. It can be found that both MLP and Bagging method shows best performance among other machine learning methods with respect to accuracy and error parameters a shown in Fig 6.

Case 3 (pre-processing with Boruta feature selection and validation using 10-fold cross validation)

In this case, we applied Boruta [R] feature selection method to find relevant features. Then, the new feature subset is used to train all 7ML methods and the validation is done with 10- fold cross-validation. Table 3 shows the classification results of various techniques in terms of accuracy and error measures for case 3. It can be found that Bagging performed best for all accuracy and error parameters as shown in Fig 7.

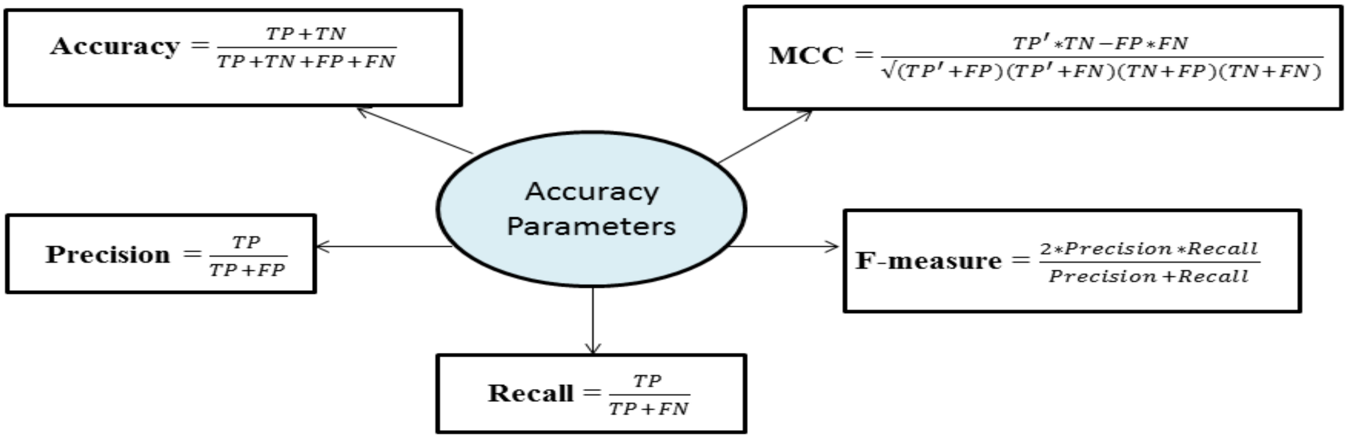
$\mathbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$
$\mathbf{MCC} = \frac{TP' * TN - FP * FN}{\sqrt{(TP'+FP)(TP'+FN)(TN+FP)(TN+FN)}}$
Accuracy Parameters
$\mathbf{Precision} = \frac{TP}{TP+FP}$
$\mathbf{F\text{-}measure} = \frac{2 * Precision * Recall}{Precision + Recall}$
$\mathbf{Recall} = \frac{TP}{TP+FN}$

Fig. 3 Accuracy parameters

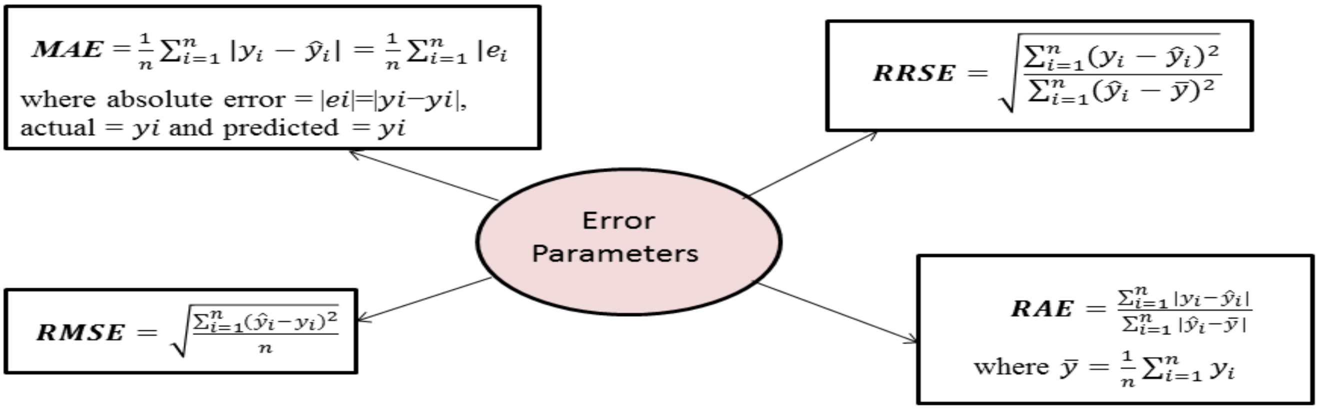
$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| = \frac{1}{n}\sum_{i=1}^{n} |e_i$
where absolute error $= |ei| = |yi - yi|$,
actual $= yi$ and predicted $= yi$
$RRSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}}$
Error Parameters
$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$
$RAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\sum_{i=1}^{n}|\hat{y}_i - \bar{y}|}$
where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

Fig. 4 Error parameter

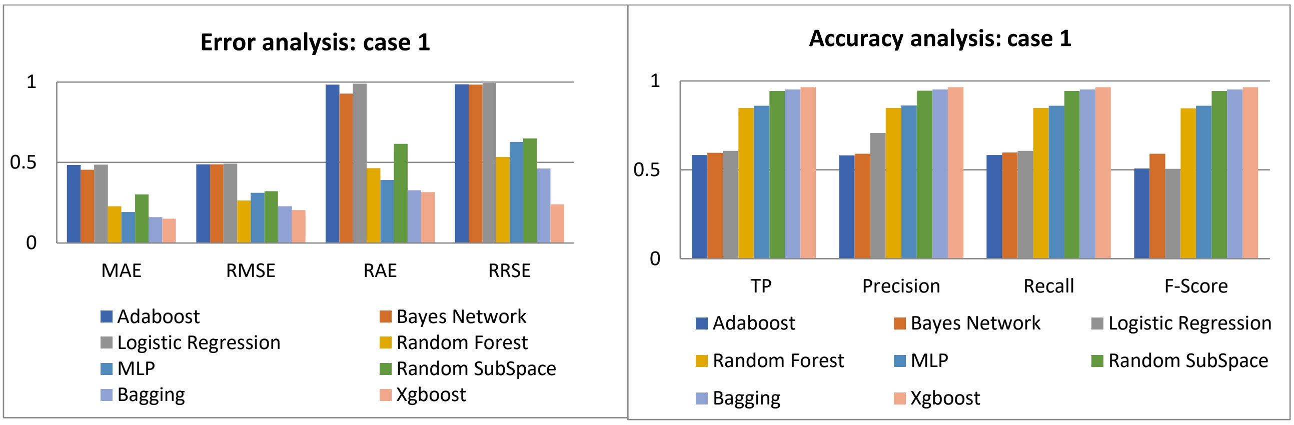

Fig.5 – Accuracy and error analysis for Case 1

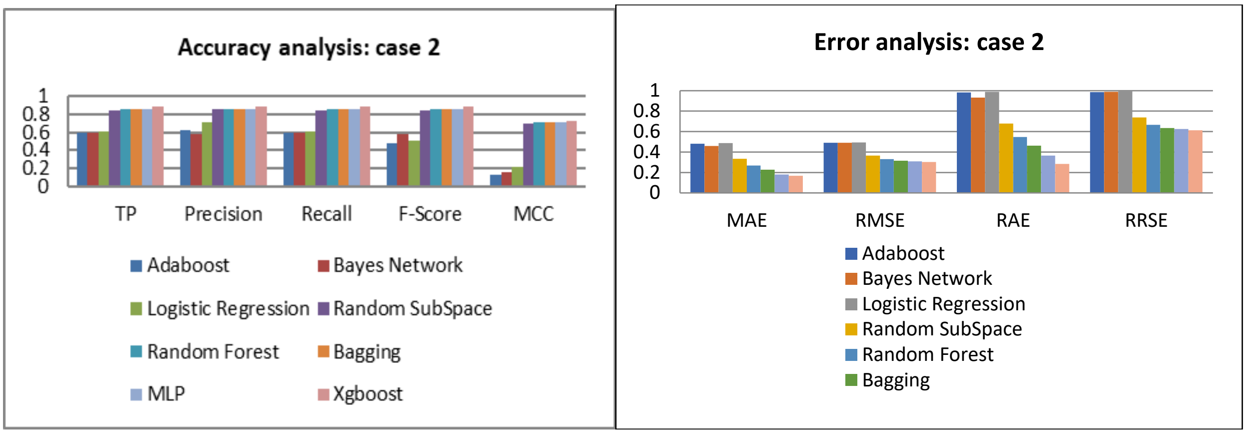

Fig.6– Accuracy and error analysis for Case 2

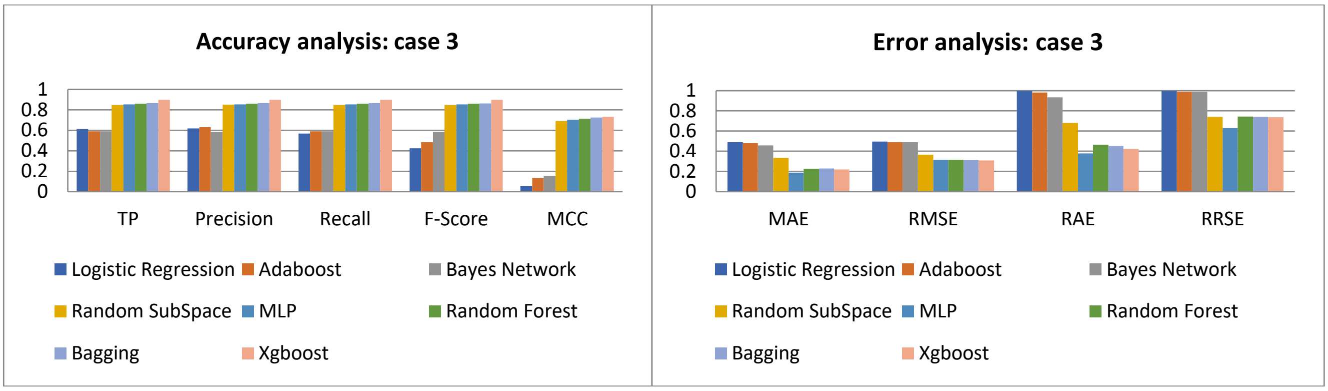

Fig.7 – Accuracy and error analysis for Case 3

Table: 1 – Accuracy and Error analysis for Case 1

	Accuracy analysis					Error analysis			
Computing Model	**TP**	**Precision**	**Recall**	**F-Score**	**MCC**	**MAE**	**RMSE**	**RAE**	**RRSE**
Adaboost	0.583	0.581	0.583	0.507	0.104	0.484	0.489	0.985	0.987
Bayes Network	0.596	0.591	0.597	0.591	0.166	0.456	0.488	0.928	0.985
Logistic Regression	0.607	0.708	0.607	0.505	0.212	0.486	0.492	0.990	0.994
Random Forest	0.847	0.848	0.847	0.846	0.689	0.228	0.265	0.464	0.535
MLP	0.861	0.862	0.861	0.861	0.719	0.192	0.312	0.391	0.629
Random SubSpace	0.943	0.945	0.944	0.944	0.887	0.302	0.321	0.616	0.649
Bagging	0.952	0.952	0.952	0.952	0.903	0.161	0.229	0.327	0.462
Xgboost	**0.964**	**0.964**	**0.964**	**0.964**	**0.905**	**0.150**	**0.205**	**0.315**	**0.241**

Table: 2 – Accuracy and Error analysis for Case 2

	Accuracy analysis					Error analysis			
Computing Model	**TP**	**Precision**	**Recall**	**F-Score**	**MCC**	**MAE**	**RMSE**	**RAE**	**RRSE**
Adaboost	0.589	0.631	0.589	0.485	0.133	0.482	0.489	0.982	0.987
Bayes Network	0.591	0.586	0.592	0.586	0.156	0.458	0.490	0.931	0.988
Logistic Regression	0.606	0.709	0.606	0.503	0.209	0.486	0.493	0.990	0.994
Random SubSpace	0.849	0.852	0.850	0.848	0.694	0.334	0.366	0.679	0.739
Random Forest	0.860	0.860	0.860	0.859	0.714	0.269	0.330	0.547	0.667
Bagging	0.861	0.861	0.861	0.860	0.716	0.227	0.315	0.462	0.636
MLP	0.862	0.862	0.862	0.862	0.719	0.180	0.309	0.366	0.624
Xgboost	**0.891**	**0.891**	**0.891**	**0.891**	**0.723**	**0.169**	**0.301**	**0.284**	**0.614**

Table 3 Accuracy and Error analysis for Case 3

	Accuracy analysis					**Error analysis**			
Computing Model	**TP**	**Precision**	**Recall**	**F-Score**	**MCC**	**MAE**	**RMSE**	**RAE**	**RRSE**
Logistic Regression	0.612	0.619	0.569	0.423	0.053	0.490	0.495	0.997	0.998
Adaboost	0.589	0.631	0.589	0.485	0.133	0.482	0.489	0.982	0.987
Bayes Network	0.591	0.585	0.591	0.585	0.154	0.458	0.490	0.933	0.989
Random SubSpace	0.848	0.850	0.848	0.847	0.691	0.334	0.367	0.681	0.741
MLP	0.853	0.853	0.853	0.853	0.702	0.187	0.317	**0.380**	**0.629**
Random Forest	0.861	0.861	0.861	0.860	0.712	0.228	0.316	0.465	0.745
Bagging	0.865	0.865	0.865	0.864	0.724	0.229	0.311	0.453	0.741
Xgboost	**0.899**	**0.898**	**0.899**	**0.898**	**0.732**	**0.221**	**0.308**	0.424	0.739

Table 4. Accuracy and Error collection for set1

	Accuracy analysis					**Error analysis**			
Computing Model	**TP**	**Precision**	**Recall**	**F-Score**	**MCC**	**MAE**	**RMSE**	**RAE**	**RRSE**
Adaboost	0.625	0.661	0.625	0.566	0.233	0.453	0.468	0.923	0.944
Bayes Network	0.653	0.651	0.654	0.651	0.289	0.397	0.464	0.808	0.936
Logistic Regression	0.794	0.795	0.795	0.795	0.584	0.268	0.369	0.546	0.744
Random SubSpace	0.884	0.884	0.884	0.884	0.764	0.261	0.316	0.532	0.638
Random Forest	0.901	0.902	0.902	0.902	0.800	0.188	0.274	0.388	0.553
Bagging	0.903	0.905	0.904	0.904	0.806	0.158	0.267	0.323	0.539
MLP	0.903	0.908	0.904	0.905	0.904	0.139	0.264	0.283	0.532
Xgboost	0.926	0.929	0.928	0.928	0.915	0.115	0.260	0.254	0.529

Table 5. Accuracy and Error collection for set2

Accuracy analysis	**Error analysis**								
Computing Model	**TP**	**Precision**	**Recall**	**F-Score**	**MCC**	**MAE**	**RMSE**	**RAE**	**RRSE**
Bayes Network	0.627	0.628	0.627	0.628	0.245	0.420	0.471	0.853	0.949
Adaboost	0.640	0.649	0.641	0.642	0.286	0.469	0.477	0.953	0.961
Random Forest	0.984	0.984	0.984	0.984	0.968	0.093	0.147	0.189	0.297
Random SubSpace	0.985	0.985	0.985	0.985	0.970	0.215	0.240	0.437	0.483
Bagging	0.989	0.989	0.989	0.989	0.978	0.032	0.100	0.064	0.202
Logistic Regression	0.991	0.991	0.991	0.991	0.982	0.050	0.106	0.102	0.214
MLP	0.991	0.991	0.991	0.991	0.981	0.010	0.085	0.021	0.171
Xgboost	**0.991**	**0.991**	**0.991**	**0.991**	**0.983**	**0.010**	**0.083**	**0.020**	**0.170**

Table 6. Accuracy and Error collection for set3

	Accuracy analysis					**Error analysis**			
Computing Model	**TP**	**Precision**	**Recall**	**F-Score**	**MCC**	**MAE**	**RMSE**	**RAE**	**RRSE**
Bayes Network	0.655	0.655	0.656	0.644	0.287	0.401	0.473	0.817	0.955
Adaboost	0.657	0.672	0.657	0.629	0.296	0.450	0.468	0.917	0.945
Logistic Regression	0.782	0.785	0.782	0.779	0.556	0.324	0.384	0.660	0.775
Random Forest	0.823	0.827	0.824	0.822	0.642	0.288	0.356	0.587	0.718

Random SubSpace	0.826	0.832	0.826	0.823	0.649	0.342	0.376	0.698	0.758
Bagging	0.834	0.835	0.835	0.833	0.663	0.245	0.340	0.500	0.686
MLP	0.850	0.850	0.850	0.850	0.696	0.186	0.318	0.378	0.642
Xgboost	**0.880**	**0.881**	**0.880**	**0.880**	**0.701**	**0.121**	**0.310**	**0.350**	**0.621**

5.2. Set wise Accuracy and error Analysis

Like the following observations are made for the individual sets to measure the performance of computing models:

Set1: The values of accuracy and error parameters were found to be highest for MLP and least for Adaboost as shown in Table 4.

Set2: MLP performed best with respect to all accuracy and error parameters whereas Bayes Network performed worst with respect to all accuracy parameters and the values of Error analysis were found to be least for Adaboost as shown in Table 5.

Set3: MLP was the best performer for all accuracy and error parameters, whereas Bayes Network performed worst with respect to all accuracy and error parameters except F-measure and RAE. For F-measure and RAE, the values of Adaboost were found to be least as shown in Table 5.

To analyze the homogeneity of data present in the three sets, Kruskal Wallis test was performed on the computing models under consideration. Since, the predicted value of computing methods followed non-normal distribution, Kruskal-Wall is test with 2 degrees of freedom was used for the analysis of data using MAE, RMSE, RAE and RRSE. The results of Kruskal-Wall test are given in Table 7.

Table 7. – Kruskal- Wall's test results for error parameters

	MAE	**RMSE**	**RAE**	**RRSE**
Chi- squared	1.2754	1.4583	1.2754	1.4583
p-value	0.5285	0.4823	0.5285	0.4823
Significance l vel	0.05	0.05	0.05	0.05

For the table 7, H0: the samples of all the sets belong to the same population, Ha: The samples in the selected data sets do not belong to the same population. Since the estimated p-value is very much higher than the significance level, the alternative hypothesis is rejected.

The outcomes of the Kruskal Wallis test indicate that the data in the three sets are not much significantly different and also the performance of seven used computing models are much statistically different from each other for the samples of all the three sets.

6. Threats to validity

6.1. Construct validity

Apart from the seven computing models and the dataset used, any other computing models as well as available datasets may be used. The evaluation of the effectiveness of the computing models may also be done with respect to other evaluation measures like pred (0.3) and completeness measures apart from the accuracy and error parameters used in this paper.

6.2. External Validity

The experimental investigation was done on the datasets made available on an organization's data repository which may or may not be generalized to all situations and datasets. The underlying pattern of software system must be taken care of before applying any computing models.

6.3. Internal Validity

Some of the computing models used in this investigation, such as, Random Forest computing model, may have required the optimization of various control parameters. Although, sincere efforts were put for the optimization of control parameters but the control parameters may vary for different datasets and software metrics.

7. Conclusion

In this study, eight efficient machine learning models were considered to compare the performance in terms of accuracy and error parameters for classifying Red Rot disease. The computing models included AdaBoost, Bayes Network, Random Forest, Logistic Regression, Bagging, Random Subspace, Xgboost and MLP. The evaluation was done on a RR dataset obtained from IRIS dataset. Three different cases including validation using test dataset, 10-fold cross validation and pre-processing with Boruta feature selection technique followed by 10-fold cross validation to calculate the performance of various computing models based on accuracy and error. A set wise analysis was also performed to check the homogeneity of the available dataset and Kruskal Wallis test was done to establish the check for the dependency of the data from different sets. Different models performed differently under different conditions. The results have shown that Xgboost method gives best performance in all cases whereas Adaboost and Logistic Regression performed worst to predict RR. In future, focus will be on verification of our model on more plant disease related datasets.

References

[1] Viswanathan, R., Geetha, N., Anna Durai, A., Prathima, P. T., Appunu, C., Parameswari, B., ... & Selvi, A. (2022). Genomic designing for biotic stress resistance in sugarcane. In Genomic designing for biotic stress resistant technical crops (pp. 337-439). Cham: Springer International Publishing.

[2] Qian, C., Zheng, B., Shen, Y., Jing, L., Li, E., Shen, L., & Chen, H. (2020). Deep-learning-enabled self-adaptive microwave cloak without human intervention. Nature photonics, 14(6), 383-390.

[3] Shahraki, A., Abbasi, M., & Haugen, Ø. (2020). Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost. Engineering Applications of Artificial Intelligence, 94, 103770.

[4] Tharmakulasingam, M. (2023). Interpretable Machine Learning Models to Predict Antimicrobial Resistance (Doctoral dissertation, University of Surrey).

[5] Lodhi, E., Wang, F. Y., Xiong, G., Dilawar, A., Tamir, T. S., & Ali, H. (2022). An AdaBoost Ensemble Model for Fault Detection and

Classification in Photovoltaic Arrays. IEEE Journal of Radio Frequency Identification, 6, 794-800.

[6] Gyftodimos, E., & Flach, P. A. (2002, July). Hierarchical bayesian networks: A probabilistic reasoning model for structured domains. In Proceedings of the ICML-2002 Workshop on Development of Representations (pp. 23-30). The university of New South Wales.

[7] Drury, B., Valverde-Rebaza, J., Moura, M. F., & de Andrade Lopes, A. (2017). A survey of the applications of Bayesian networks in agriculture. Engineering Applications of Artificial Intelligence, 65, 29-42.

[8] Mohammed, Z. A., Abdullah, M. N., & Al Hussaini, I. H. (2021). Predicting incident duration based on machine learning methods. Iraqi Journal of Computers, Communications, Control and Systems Engineering, 21(1), 1-15.

[9] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8, 1-74.

[10] Zhang, Y., Cao, G., Wang, B., & Li, X. (2019). A novel ensemble method for k-nearest neighbor. Pattern Recognition, 85, 13-25.

[11] Yang, Y., Ali, N., Khan, A., Khan, S., Khan, S., Khan, H., ... & Bilal, M. (2021). Chitosan-capped ternary metal selenide nanocatalysts for efficient degradation of Congo red dye in sunlight irradiation. International Journal of Biological Macromolecules, 167, 169-181.

[12] Bhagya Raj, G. V. S., & Dash, K. K. (2022). Comprehensive study on applications of artificial neural network in food process modeling. Critical reviews in food science and nutrition, 62(10), 2756-2783.

[13] Ribeiro, M. H. D. M., & dos Santos Coelho, L. (2020). Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series. Applied soft computing, 86, 105837.

[14] Ribeiro, M. H. D. M., & dos Santos Coelho, L. (2020). Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series. Applied soft computing, 86, 105837.

[15] Skurichina, M., & Duin, R. P. (2002). Bagging, boosting and the random subspace method for linear classifiers. Pattern Analysis & Applications, 5, 121-135.

[16] Rashid, M., Bari, B. S., Yusup, Y., Kamaruddin, M. A., & Khan, N. (2021). A comprehensive review of crop yield prediction using machine learning approaches with special emphasis on palm oil yield prediction. IEEE access, 9, 63406-63439.

[17] Soni, M., & Varma, S. (2020). Diabetes prediction using machine learning techniques. International Journal of Engineering Research & Technology (Ijert) Volume, 9.

[18] Menshawi, A., Hassan, M. M., Allheeib, N., & Fortino, G. (2023). A Hybrid Generic Framework for Heart Problem diagnosis based on a machine learning paradigm. Sensors, 23(3), 1392.

[19] Maftouni, M. (2023). Development of Novel Attention-Aware Deep Learning Models and Their Applications in Computer Vision and Dynamical System Calibration (Doctoral dissertation, Virginia Tech).

[20] Abonazel, M. R., & Ibrahim, M. G. (2018). On estimation methods for binary logistic regression model with missing values. International Journal of Mathematics and Computational Science, 4(3), 79-85.

[21] Nabipour, M., Nayyeri, P., Jabani, H., Shahab, S., & Mosavi, A. (2020). Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. IEEE Access, 8, 150199-150212.

[22] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[23] Pang, L., Wang, J., Zhao, L., Wang, C., & Zhan, H. (2019). A novel protein subcellular localization method with CNN-XGBoost model for Alzheimer's disease. Frontiers in genetics, 9, 751

[24] Begum, K. J., & Ahmed, A. (2015). The importance of statistical tools in research work. International Journal of Scientific and Innovative Mathematical Research, 3(12), 50-58.

[25] Jiang, J., Elguindi, S., Berry, S. L., Onochie, I., Cervino, L., Deasy, J. O., & Veeraraghavan, H. (2022). Nested block self-attention multiple resolution residual network for multiorgan segmentation from CT. Medical Physics, 49(8), 5244-5257.