Statistical analysis of machine learning techniques for predicting powdery mildew disease in tomato plants

Anshul Bhatia*, Anuradha Chug
and Amit Prakash Singh

University School of Information, Communication, and Technology,
Guru Gobind Singh Indraprastha University,
Dwarka, Sector-16C, New Delhi-110078, India
Email: anshul.usict.127164@ipu.ac.in
Email: anuradha@ipu.ac.in
Email: amit@ipu.ac.in
*Corresponding author

Abstract: Powdery mildew is a dangerous disease that reduces the quality and the yield of tomato fruit rapidly. Its early prediction is a prior requirement for obtaining good quality fruit. Therefore, in this study, the best classifier amongst various classifiers has been discovered using different machine learning algorithms. This classifier can precisely classify whether the meteorological conditions of a particular day are conducive to the development of powdery mildew disease or not. Tomato powdery mildew disease dataset has been tested using various performance measures and the results computed for all the classifiers are promising. Friedman test has been used to rank multiple classifiers and post hoc analysis has also been done using the Nemenyi test. It has been observed in comparison that 62.05% of the total pairs of classifiers perform significantly different from each other, and medium Gaussian support vector machine (MGSVM) is the best classifier with 94.74% accuracy.

Keywords: plant disease; tomato; powdery mildew; machine learning algorithm; Friedman test; Nemenyi test; classifier.

Reference to this paper should be made as follows: Bhatia, A., Chug, A. and Singh, A.P. (2021) 'Statistical analysis of machine learning techniques for predicting powdery mildew disease in tomato plants', *Int. J. Intelligent Engineering Informatics*, Vol. 9, No. 1, pp.24–58.

Biographical notes: Anshul Bhatia is currently pursuing PhD from University School of Information, Communication and Technology (USICT), Guru Gobind Singh Indraprastha University (GGSIPU), Delhi since September 2018. She has acquired her Bachelor's and Master's degree in Computer Science and Engineering from Rajasthan Technical University. Her areas of interest include Internet of Things (IoT), machine learning, and data science.

Anuradha Chug is working as Assistant Professor in USICT, GGSIPU. She has long teaching experience of almost 30 years to her credit as faculty and in administration at various educational institutions in India. She has also worked as guest faculty in Netaji Subhash Institute of Information and Technology, Dwarka, New Delhi and Regular Faculty at Government Engineering College, Bikaner. She has earned her doctorate degree in Software Engineering from the Delhi Technological University, Delhi, India. She has achieved top rank in her

MTech (IT) degree and conferred the University Gold Medal in 2006 from GGSIPU.

Amit Prakash Singh is working as Professor in USICT, GGSIPU. He obtained PhD in Information Technology from Guru Gobind Singh Indraprastha University, Delhi in 2011. He has worked in the area of Artificial Neural Networks. He has earlier worked as Lecturer in University of Hyderabad and Banasthali Vidyapith. His areas of interest are artificial neural network, embedded system design and digital system design. He has published more than 50 research papers in international/national journals and attended various national/international workshops and seminar. He also visited UK, Portugal and Singapore to present his research work in international conferences.

1 Introduction

Background: Tomato is one of the most popular fruits consumed by a vast population globally. However, this widely used species of fruit is in danger due to rapidly changing climatic conditions and global warming. This crop suffers from multiple contagious diseases such as fusarium wilt, grey mould, late blight, early blight, etc. (Jones and Thomson, 1987). Powdery mildew is also one of the dangerous diseases caused by a fungal pathogen 'LeveillulaTaurica' found in all major growing parts of a tomato plant (Bhatia et al., 2020b). It can harm the tomato plant's quality and may reduce its yield by up to 40% (Haroun, 2002). Figure 1 shows the snapshot of powdery mildew disease in infected tomato leaves. This dangerous disease also infects other plants like cucumber, pepper, and eggplant (Palti, 1988). Meteorological conditions such as relative humidity, temperature, global radiations, wind speed, and leaf wetness are critical factors for the development of powdery mildew in tomato plants (Bakeer et al., 2013). Continuous monitoring, prior detection, and treatment of powdery mildew are critical processes. Much before powdery mildew can be observed in the farm field, many leaves have already been contaminated without revealing any symptoms and signs of the disease. The most common symptoms of this disease are green to yellow stains on the upper surface of the foliage and whitish grey to purple blotches on the lower portion of the leaf surface. Later on, the contaminated part of the leaves get dry and change into a dark brown colour. Lastly, the plant leaves may start withering. Usually, older leaves are more prone to this disease than the younger ones (Holliday, 1980; Correll et al., 1988; McGrath et al., 2001; Haroun, 2002; Reis et al., 2004). However, powdery mildew does not contaminate the fruit or stem of the tomato plant, but it can diminish the quality of tomato fruit, lessen the growth of the plant and can gradually decrease the tomato production (Braun, 1987; Jones and Thomson, 1987; Farag, 2003).

Problem statement: In 1997, Guzman-Plazola stated that one of the ways to tackle powdery mildew is fungicide spraying, a necessary and helpful approach to control the disease when environmental conditions are conducive for the disease to develop (Guzman-Plazola, 1997). However, extensive use of fungicides may increase the growth of pathogens, which can further increase the severity of powdery mildew in tomato plants (Avenot and Michailides, 2010). This creates an alarming situation for farmers and agricultural experts, which motivates them to optimise the prevention strategy for powdery mildew by reducing chemical application.

Motivation: For optimum use of fungicides, a farmer should spray chemicals only on the day when meteorological conditions are conducive to the powdery mildew disease instead of following a routine schedule of spraying. Hence, an efficient and on-time application of fungicides is necessary for the prevention of this disease. Implementation of a disease prediction model is an essential strategy for the early detection of powdery mildew disease in tomato plants, which in turn helps in the precise and timely estimation of usage of fungicides on crops (Verma et al., 2020). If the model classifies a particular day as conducive, a warning may be sent to the farmer, indicating the need to spray the fungicide at that point. This way, the recommendations of the proposed disease forecasting models can reduce the unnecessary fungicide spray with no significant impact on fruit yield and quality.

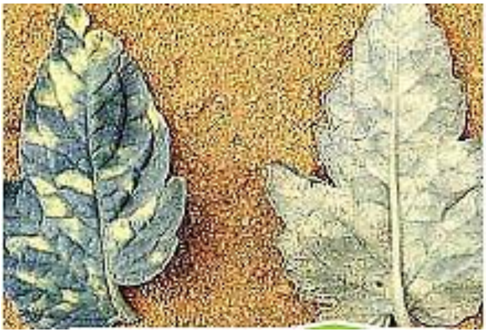
Contribution: In this research, Tomato powdery mildew disease (TPMD) dataset collected by Bakeer et al. (2013) was used to build a prediction model to decide the optimum quantity of fungicide spray, in turn, saving the tomato plant from powdery mildew disease with minimal spray. Overall, 23 different classification algorithms were used in this study namely, fine tree (FT), quadratic discriminant analysis (QDA), cubic support vector machine (CSVM), quadratic support vector machine (QSVM), linear support vector machine (LSVM), coarse Gaussian support vector machine (CGSVM), linear discriminant analysis (LDA), logistic regression (LR), coarse *k*-nearest neighbour (C*k*NN), medium *k*-nearest neighbour (M*k*NN), weighted *k*-nearest neighbour (W*k*NN), medium Gaussian support vector machine (MGSVM), fine Gaussian support vector machine (FGSVM), cubic *k*-nearest neighbour (Cub*k*NN), coarse tree (CT), RUSBoosted trees (RBT), boosted trees (BTT), subspace *k*-nearest neighbour (S*k*NN), fine *k*-nearest neighbour (F*k*NN), cosine *k*-nearest neighbour (Cos*k*NN), medium tree (MT), bagged trees (BGT), and, subspace discriminant (SD). An extensive comparison of the results of these classification algorithms using seven performance measures, namely classification accuracy (CA), Precision, Matthews correlation coefficient (MCC), Recall, area under curve (AUC), Specificity, and F1-Score has been done. Advance statistical Friedman test followed by post hoc analysis using the Nemenyi test was also performed to analyse if there exists a significant difference among the performance of any particular classification algorithm. This study helps us to determine which classifier performs the best on this specific type of dataset so that correct predictions can be made during the real-time application of these classification models. This analysis can predict whether the meteorological conditions for a particular day are conducive for the development of the disease or not and also help farmers to forecast precise intervals of fungicide spray to control the powdery mildew in the tomato plants. To understand which classifier is most suitable for plant disease prediction, following research questions (RQs) are framed:

RQ$_1$: Can classification techniques used in this study be successfully applied for precise prediction of powdery mildew disease in tomato plant?

RQ$_2$: What is the comparative performance of different classification techniques, used for prediction of powdery mildew disease in tomato plant?

RQ$_3$: Which pair of classification techniques performs significantly different from each other in terms of performance measures?

Figure 1 Snapshot of symptoms of powdery mildew disease caused by LeveillulaTaurica (see online version for colours)



Source: Shankar *et al.* (2014)

Outline of research: The rest of the paper is further divided into six sections, stated as: Section 2 presents the previous study, and section 3 focuses on the research methodology used in this paper. Also, section 4 shows the material and methods used in this research, which explains the overview and descriptive statistics of the dataset, classification modelling methods, classifier evaluation techniques, and statistical tests. Section 5 presents the results obtained from the classifiers' evaluation and analysis. Lastly, Section 6 provides the conclusion and direction for future work.

2 Related work

Various researchers have proposed different classifiers for plant disease detection, as demonstrated in Table 1. Steddom et al. (2003) used the concept of LR for the classification between the Rhizomania inoculated diseased and healthy sugar beet leaves. Subsequently, Yao et al. (2009) worked on the support vector machine (SVM) classifier in their research to classify three rice crop diseases: rice sheath blight, rice blast, and leaf blight. Similarly, in the next year, Rumpf et al. (2010) also examined the SVM classifier for discrimination between diseased and non-diseased sugar beet leaves. They have used multiple classification algorithms like decision tree, artificial neural network (ANN), and SVM to differentiate between the diseases found in sugar beet leaves named powdery mildew, leaf rust, and cercospora leaf spot. Authors have used CA, Specificity, and Recall as the accuracy metrics. Later, Romer et al. (2011) developed an SVM based classifier for the categorisation of non-diseased and leaf-rust infected wheat leaves. They achieved the CA of 93% in their study (Römer et al., 2011). In the same year, Bauer et al. (2011) built a *k*-nearest neighbour (*k*NN) classifier to categorise healthy and infected sugar beet leaves having diseases named as 'Cercosporabeticola' and 'Uromycesbetae'. In another study, Sankaran et al. (2011) used multiple classifiers (LDA, QDA, *k*NN, etc.) for classifying the healthy and Huanglongbing (HLB) infected citrus plants. They used only CA as the performance measure and achieved 91% accuracy. In one of the studies conducted by Patil and Zambre (2014), the concept of image processing with SVM classifier has been proposed for discrimination of diseased and healthy leaves to detect cotton leaf spot disease. They have used only CA as a performance measure. Pujari et al. (2015) further worked in the field of fungal disease detection using image processing techniques. They have used multiple classifiers like *k*NN, SVM, and ANN in their study. In the next year, Padol and Yadav (2016) also contributed to this research by focusing on the identification and classification of fungal diseases of grape plants like powdery mildew and downy mildew using image processing and SVM classifier. Subsequently, ANN and SVM algorithms were also used with a feature reduction approach to identify various plant diseases (Pujari et al., 2016).

Table 1 Past empirical studies on plant disease datasets

Reference (Year)	*Diseases*	*Plant/crop*	*Performance measures*	*Classifiers/ method used*
Steddom et al. (2003)	Rhizomania	Sugar beet	CA, Specificity, Recall	LR
Yao et al. (2009)	Rice sheath blight, rice blast, and leaf blight	Rice	CA	SVM
Römer et al. (2011)	Leaf rust	Wheat	CA	SVM
Bauer et al. (2011)	Cercosporabeticola and Uromycesbetae	Sugar beet	Classification Rate/CA	*K*NN
Sankaran et al. (2011)	Huanglongbing (HLB)	Citrus	CA	LDA, QDA, *k*NN
Patil and Zambre (2014)	Leaf Spot	Cotton	CA	SVM
Pujari et al. (2015)	Fungal diseases as grey mildew, anthracnose, alternaria leaf spot smut, powdery mildew, downy mildew, rot, wilt, leaf blight, leaf spot, smut, rust	Mango, Pomegranate, Grape, Chili, Cotton, Sugarcane, Cereal,	CA	*K*NN, SVM, and ANN
Padol and Yadav (2016)	Fungal diseases as powdery mildew and downy mildew	Grape	CA	SVM
Pujari et al. (2016)	Leaf rust, powdery mildew, anthracnose, downy mildew, stalk rot, bacterial wilt, angular leaf spots, galls, mosaic, yellow virus, veinal chlorosis, enations, foliar nematode, cyst disease, ear cockle, ear rot	Wheat, sunflower, grape, maize, cucumber, cotton, lime, tomato, sugarbeet, soybean	Recall, False Positive Rate, Precision, True Positive Rate, CA, F1-score	SVM, ANN

Table 1 Past empirical studies on plant disease datasets (continued)

Reference (Year)	*Diseases*	*Plant/crop*	*Performance measures*	*Classifiers/ method used*
Naik et al. (2017)	Soybean disease	Soybean	CA, MPCA, Cost Metric, Interpretability, and Cross-Validated MPCA	Decision Tree, LD Analysis, Naïve Bayes, Random Forest, SVM, *k*NN, QDA, Gaussian Mixture Model
Mattihalli et al. (2018)	Leaf Spot, Early Blight, Late Blight, Downy Mildew	–	–	Internet of Things (No classifiers)
Mohammadpoor et al. (2020)	Grapevine Fanleaf Virus	Grape	CA, Specificity, Sensitivity	Fuzzy C-Means, SVM
Bhatia et al. (2020b)	Powdery Mildew	Tomato	CA, AUC, F1-score	SVM, LR
Bhatia et al. (2020a)	Powdery Mildew	Tomato	CA, AUC	ELM

Further, a research was conducted in which Naik et al. (2017) used 10 classification-based machine learning techniques to determine the plant stress severity rating in the soybean plant. They have used CA, cost metric, interpretability, and mean per class accuracy (MPCA) as the accuracy measures (Naik et al., 2017). Later, Mattihalli et al. (2018) worked to detect plant diseases using the Internet of Things approach (Mattihalli et al., 2018).

Apart from the above studies, Verma et al. (2018) presented a review of various models developed for predicting several diseases of tomato plants. Further, Verma et al. (2019) also developed a mobile application based on deep learning architecture for plant disease prediction. Later, they extended their work by developing a CNN-based model for evaluating disease severity in tomato plants (Verma et al., 2020).

Afterward, Mohammadpoor et al. (2020) proposed a Fuzzy C-Means and SVM-based hybridised technique to predict Grapevine Fanleaf Virus disease in grape plants. Their prediction system was 98.6% accurate. Recently, Bhatia et al. (2020a) have applied extreme learning machine (ELM) algorithm with various resampling techniques on TPMD dataset and achieved the highest accuracy of 89.19%. In one of the studies, they have also proposed a Hybrid SVM-LR classifier for powdery mildew disease prediction in the tomato plant. They have conducted the experiment on the TPMD dataset and achieved an accuracy of 92.37% (Bhatia et al., 2020b).

Although several prediction models have already been proposed by several researchers for predicting diseases in different plants, even then we cannot determine a generalised model to predict a specific disease for a specific plant. Moreover, none of the researchers has used statistical tests in their studies. Statistical analysis is necessary because it provides some crucial evidence that claims that there exist some classifiers that are considered superior to others based on the statistical structure of a particular dataset. Hence, to overcome the above-identified gaps of existing studies, 23 machine learning based classification methods for powdery mildew prediction are employed and compared in this paper to identify which classification modelling method is most appropriate for the TPMD dataset. Further, statistical analysis has also been done using Friedman test and Nemenyi post hoc analysis to establish confidence over the performance of one classifier over the other.

3 Research methodology

The proposed methodology introduces an efficient and robust approach for powdery mildew disease prediction in tomato plants. Powdery mildew disease severely damages the quality of tomato crops; hence, early prediction of the disease and timely application of fungicides are much needed. Therefore, in the current study, an attempt has been made to find out an efficient and precise plant disease prediction model for the prior detection of powdery mildew disease. The suggested prediction model will be helpful in effective control of powdery mildew in tomato plants, which also results in a reduction of the amount of fungicide spray.

Although thorough research has been conducted over prior studies even then, it cannot be concluded that which classifier will predict tomato powdery mildew disease with the highest accuracy. Therefore, in this study, various classifiers have been used to develop a disease prediction model for tomato crops. Afterward, to identify the best classifier for powdery mildew disease prediction, these prediction models' results are further evaluated and compared using several different performance metrics.

Further, the statistical significance of the proposed classifiers has also been tested using statistical Friedman and Nemenyi test. The overall methodology of this study is described as follows. A diagrammatic representation of the proposed approach is shown in Figure 2.

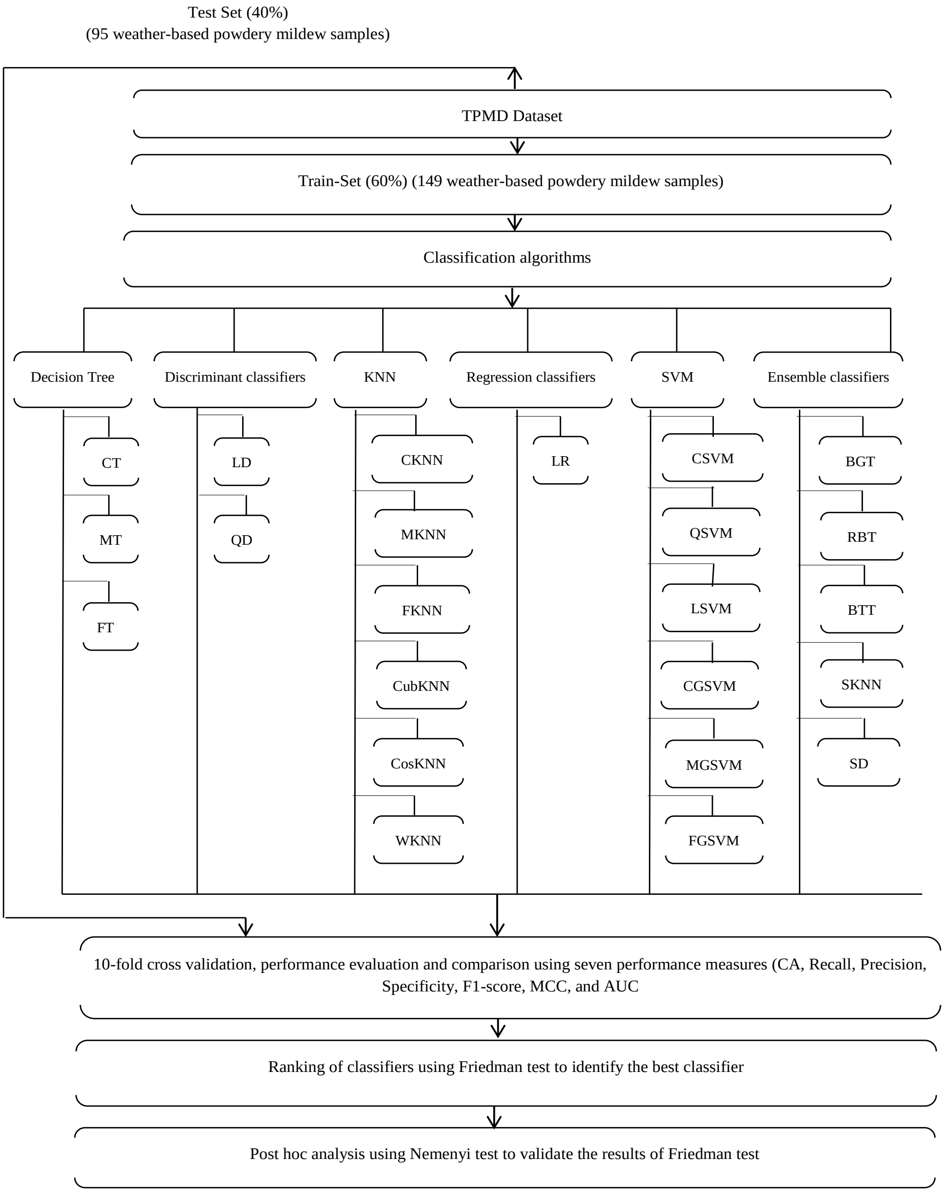
3.1 Dataset selection

Initially, to conduct this study, a weather-based TPMD dataset was selected, collected by the scientists of the Agricultural Research Center, Plant Pathology Institute, Egypt (Bakeer et al., 2013). This dataset contains 244 sensor-based records, which show whether the meteorological conditions on a particular day are conducive for the development of tomato powdery mildew disease or not. The dataset contains various attributes like date, relative humidity (RH), leaf wetness (LW), temperature (T), global radiations (GR), wind speed (WS), and day-prediction (DP). In the current study, GR (watt/m^2), LW (%), WS (KM/h), RH (%), and T (°C) are used as the independent or the predicted variables and day prediction (DP) (conducive or non-conducive) is used as dependent or the target variable for developing the prediction model. The dataset has been divided into 60–40 (Training data = 149 samples and Testing data = 95 samples) train-test ratio for developing the disease prediction models.

3.2 Training and validation of models

Next, training data has been used to develop plant disease prediction models based on each of the 23 classifiers namely, FT, CT, MT, QDA, LDA, LR, CSVM, QSVM, LSVM, CGSVM, MGSVM, FGSVM, C*k*NN, M*k*NN, F*k*NN, Cos*k*NN, W*k*NN, Cub*k*NN, BGT, RBT, BTT, S*k*NN, and, SD. These classifiers are the variants of six most prominent machine learning algorithms i.e., decision tree, discriminant classifier, regression classifier, SVM, *k*NN, and ensemble classifier.

Figure 2 Research methodology performance evaluation of models


Test Set (40%)
(95 weather-based powdery mildew samples)
TPMD Dataset
Train-Set (60%) (149 weather-based powdery mildew samples)
Classification algorithms
Decision Tree
Discriminant classifiers
KNN
Regression classifiers
SVM
Ensemble classifiers
CT
MT
FT
LD
QD
CKNN
MKNN
FKNN
CubKNN
CosKNN
WKNN
LR
CSVM
QSVM
LSVM
CGSVM
MGSVM
FGSVM
BGT
RBT
BTT
SKNN
SD
10-fold cross validation, performance evaluation and comparison using seven performance measures (CA, Recall, Precision, Specificity, F1-score, MCC, and AUC
Ranking of classifiers using Friedman test to identify the best classifier
Post hoc analysis using Nemenyi test to validate the results of Friedman test

FT, CT, and MT are the variants of the decision tree algorithm. For each decision tree variant, there are two parameters i.e., Maximum number of splits (ψ) and split criterion (φ). Here, the maximum number of splits (ψ) indicates the maximum number of branch nodes used for the decision tree. In contrast, the split criterion (φ) is another parameter based on which decision tree splits the nodes. The most widely used split criteria of the decision tree are maximum deviance reduction, twoing rule, and Gini's diversity index.

LDA and QDA are variants of discriminant analysis classifier, and for each of these variants, there is only one parameter i.e., discriminant type (ꞔ). There are six discriminant types of discriminant analysis classifier for each LDA and QDA with pseudo and diagonal variants of each type.

For the LDA classifier, each class has the same covariance matrix, whereas the covariance matrix in the QDA classifier can vary among classes. These six discriminant types are as follows: linear, pseudo linear, diagonal linear, quadratic, pseudo quadratic, and diagonal quadratic. CSVM, QSVM, LSVM, CGSVM, MGSVM, FGSVM are the variants of the SVM algorithm, and for each of these variants, there are five parameters, namely, Kernel Function (K_f), Kernel Scale (£), box constraint level (*ƀ*). Kernel Function (K_f) is a parameter used by each type of SVM algorithm to convert the non-linear separable problem into a linearly separable problem by adding more dimensions. The most common kernel functions are linear, Gaussian, cubic, and quadratic functions. Kernel Scale (£) is a parameter whose value is set in the range of 0.01–1000. Its value can be optimised only when the kernel function is set to Gaussian kernel. For all other kernel functions, the kernel scale is set to 1. Box Constraint Level (*ƀ*) is set in the range of 0.01–1000. The increasing value of this parameter can reduce the number of support vectors during training, and it can also increase training time.

Similarly, C*k*NN, M*k*NN, F*k*NN, Cos*k*NN, W*k*NN, Cub*k*NN are the variants of *k*NN classifiers. All of these variants include mainly three parameters i.e., number of neighbours (*k*), distance metric (*ĐM*), and distance weight (*ĐW*). The value of the number of neighbours (*k*) is set in the range of [1, max (2, round (ϕ/2))], where ϕ represents the total number of data samples present in the dataset. Distance metric (*ĐM*) parameter is used to determine the distance between different data samples. The most common distance metrics (*ĐMs*) are mahalanobis, Euclidean, hamming, minkowski (cubic), chebyshev, city block, jaccard, correlation, cosine, and spearman. Distance weight (*ĐW*) is another parameter whose value is selected from the following values: equal (no weights), inverse (weight is 1/distance), or squared inverse (weight is 1/distance^2).

BGT, RBT, BTT, S*k*NN, and SD are the ensemble classifiers that commonly include three parameters, namely ensemble method (*È*), learner type (*l*), and number of Learners (*ʒ*). Each variant of the ensemble classifier includes different ensemble methods (*È*) like AdaBoost, RUSBoost, LogitBoost, GentleBoost, and Bag. As ensemble classifier includes a combination of various techniques; therefore, the higher value of the number of learner (*ʒ*) can improve the accuracy of the model. Its value is set in the range of 10 to 500. Learner type (*l*) shows the primary learning method used in the ensemble classifier like, decision tree, *k*NN, and discriminant analysis. In conjunction with these three parameters, BGT and RBT include two more parameters, i.e., learning rate (*ɳ*) and the maximum number of splits (*ψ*). The value of learning rate (*ɳ*) should be less than 1 for the highly precise model. Also, the value of maximum number of splits (*ψ*) or the number of branch nodes is set in the range of [1, max (2, round (ϕ/2))], where ϕ represents the total number of data samples presents in the dataset. The values of parameters that have been used for each of these classifiers are shown in the result section.

10-fold cross-validation has also been used during training to reduce validation bias. This procedure divides all the data samples into 10 subgroups, from which nine subgroups are used to train model, and the tenth subset validate that model. This process is repeated continuously to achieve disease prediction values on all the data samples.

After training, each of the models obtained from classifiers is evaluated using testing data. The result obtained from this testing procedure is further used for the performance evaluation of classifiers. Subsequently, each classifier's performance has been evaluated using seven different accuracy metrics, namely, i.e., CA, Precision, Recall, Specificity, F1-Score, MCC, and AUC.

However, these different accuracy measures produce very different results about the supremacy of one classifier over the other. Therefore, in the next step, the statistical significance of proposed classifiers has also been checked to determine the best classifier for the disease prediction.

3.3 Statistical analysis of proposed models

Lastly, the Friedman statistical test and post-hoc Nemenyi test have also been used to evaluate the predictive capability of proposed classifiers. Initially, the Friedman test has been applied to check whether there is any significant difference between the proposed classifiers' performance.

The Friedman test provides a ranking amongst different techniques depending upon their predictive performance. The Friedman test results are also validated through post-hoc analysis using the Nemenyi test to compare the best performing classifier with all the other classifiers. This test determines which pairs of techniques are statistically significantly different from each other.

The proposed approach will be helpful to find out the best classifier for powdery mildew disease prediction in tomato plants. Based on this prediction, if a particular day is identified as conducive, then a message can be sent to farmers to take necessary preventive measures well in advance. In this way, the proposed approach will help the farmers in forecasting much more precise intervals of fungicide spray compared to the routine schedule of spraying to control the tomato powdery mildew disease. All the materials and methods used for the proposed methodology are explained in detail in the next section.

4 Materials and methods

This section includes the materials and methods used in this study. The implementation of this work was done using MATLAB R2018b.

4.1 Overview of dataset

TPMD dataset used in this study includes information about the severity of powdery mildew disease based on daily weather parameters like RH, LW, T, GR, and WS (Bakeer et al., 2013). A snapshot of the TPMD dataset is also shown in Table 2. The overall dataset contains 244 data points, including 217 non-conducive and 27 conducive classes. The dataset contains various attributes like date, T, RH, LW, WS, GR, and day-prediction (DP). T, RH, LW, WS, GR are the critically responsible factors for the growth of a plant. As the proper range and values of these parameters can enhance the quantity and quality of the plants' yields; likewise, their inadequate range can infect the plant and develop diseases by producing bacteria, viruses, and fungi. So, these attributes have to be considered during the evaluation of the severity of plant disease.

Table 2 Sample of TPMD dataset

Date	*T*	*RH*	*LW*	*WS*	*GR*	*DP*
20-09-06	26.6	64	22	1	56	Non-Conducive
21-09-06	27.9	68	23	2	55	Non-Conducive
22-09-06	26.5	72	25	2	54	Non-Conducive
23-09-06	25.3	76	24	1	56	Conducive
24-09-06	25.8	80	22	1	55	Conducive
25-09-06	26.6	83	25	1	58	Conducive
26-09-06	27.7	78	24	2	56	Non-Conducive
27-09-06	28.8	74	22	1	54	Non-Conducive
28-09-06	29.4	70	21	2	51	Non-Conducive
29-09-06	28.2	67	23	1	48	Non-Conducive
30-09-06	28.6	65	25	1	50	Non-Conducive
01-10-06	29.3	66	35	2	52	Non-Conducive
02-10-06	28.2	63	33	3	54	Non-Conducive
03-10-06	29.5	61	31	4	56	Non-Conducive

Table 3 shows a description of these attributes. DP (conducive/non-conducive) is the response attribute (Class label), and all the other attributes (T, RH, LW, WS, and GR (except Date)) are predictors whose values conclude the result. Thus, the attribute DP measures whether the meteorological condition of a particular day is conducive or not for the occurrence of powdery mildew in the tomato plant.

Table 3 Description of TPMD dataset

S. No.	*Attributes*	*Variables* (units)	*Description*
1	Date	NA	The researchers collected this dataset in two cycles: From the date 1/09/2006 to 31/12/2006 From the date 1/09/2007 to 31/12/2007
2	Temperature	T (°C)	The higher or lower range of temperature within surroundings can cause various diseases in any plant or crop. The temperature sensor present in the weather station, as mentioned above, has been used to collect the temperature every day
3	Relative humidity	RH (%)	Humidity is the most critical meteorological factor which directly affects the growth of a plant. High humidity can cause infection and disease development by various bacteria and fungi. A relative humidity sensor is used for gathering the information of humidity per day

Table 3 Description of TPMD dataset (continued)

S. No.	*Attributes*	*Variables* (units)	*Description*
4	Leaf wetness	LW (%)	Plant surface wetness (Leaf Wetness) is a crucial meteorological variable for predicting disease in any plant. Leaf wetness sensor present in the weather station has been used to collect the temperature every day
5	Wind speed	WS (KM/h)	Wind can spread pathogens from one place to another, even many miles away. So wind speed can infect the plant gradually. A wind speed sensor is used for measuring the wind speed per day
6	Global radiations	GR (watt/m^2)	Global radiation also enhances the severity of disease in plants. Global radiation sensor has been used for collecting data regarding global radiations
7	Day-prediction	DP(conducive or non-conduciveness)	This variable tells us whether the meteorological condition of a particular date is conducive or not for the development of powdery mildew in the tomato plant

4.2 Descriptive statistics of TPMD dataset

A mathematical description of the TPMD dataset is presented in Table 4. As shown, the distribution of variables T, RH, LW, WS, and GR are platykurtic, whereas the distribution of one variable, i.e., DP is leptokurtic. We can see that the distribution of all the variables except WS and GR are negatively skewed. These observations regarding the mathematical structure of TPMD dataset results in a hypothesis that the variables' distribution is not normal. Shapiro–Wilk Test of normality has been used to test this hypothesis (Shapiro and Wilk, 1965). This test of normality is suitable for small sample sizes and can handle large sample sizes up to 2000. A Shapiro-Wilk test state that if the significance value obtained from the test is greater than 0.05, then the data is normal; otherwise, it diverts from a normal distribution.

Table 4 Descriptive statistics for TPMD dataset

Variable	*Min*	*Max*	*Median*	*Mean*	*Standard deviation*	*Skewness*	*Kurtosis*
T	18.3	33.2	25.7	25.40	3.18	–0.10	–0.76
RH	26.6	96.0	72.0	71.95	11.07	–0.59	0.91
LW	11.0	40.0	29.0	28.30	6.00	–0.45	0.03
WS	1.0	5.0	2.0	2.34	0.96	0.19	–0.80
GR	25.0	58.0	39.0	39.90	7.54	0.33	–0.62
DP	1.0	2.0	2.0	1.89	0.31	–2.47	4.10

Table 5 shows the results of Shapiro–Wilk for the TPMD dataset. From Tables 4 and 5, it is evident that none of the six variables are normally distributed in the TPMD dataset. Hence, a non-parametric test known as the Kruskal Wallis test has also been conducted in this study to check whether the population distribution of variables are identical or not (McKight and Najab, 2010). If the significance value of the Kruskal Wallis test is greater than 0.05, then it can be concluded that the population distribution of the two variables is identical; otherwise, both the variables do not follow the same type of distribution. Table 6 shows the statistics obtained from the Kruskal Wallis test. From the Table 6, it can be observed that the population distribution of different variable pairs, namely T-RH, T-LW, T-WS, T-DP, LW-WS, LW-GR, WS-GR, and GR-DP (marked as bold) are identical. In contrast, the remaining variable pairs, i.e., T-GR, RH-LW, RH-WS, RH-GR, RH-DP, LW-DP, and WS-DP, are not identical. The descriptive statistics of the TPMD dataset, Shapiro–Wilk test, and Kruskal Wallis test are useful in understanding and comparing the characteristics of the TPMD dataset.

Table 5 Shapiro-Wilk statistics for TPMD dataset

Variable	*Shapiro-Wilk statistic*	*Significance*
T	0.982	0.004
RH	0.973	0.000
LW	0.976	0.001
WS	0.884	$1.024e^{-12}$
GR	0.977	0.001
DP	0.362	$2.2e^{-16}$

Table 6 Kruskal Wallis test for TPMD dataset

Variables	*Kruskal Wallis Chi-Square statistic*	*Significance*	*Df*
T-RH	**63.89**	**0.09**	**50**
T-LW	**38.12**	**0.08**	**27**
T-WS	**5.728**	**0.2**	**4**
T-GR	179.87	$2.2e^{-16}$	32
T-DP	**0.19701**	**0.65**	**1**
RH-LW	105.93	$2.68e^{-11}$	27
RH-WS	13.02	0.01	4
RH-GR	69.686	0.00	32
RH-DP	48.829	$2.793e^{-12}$	1
LW-WS	**7.9837**	**0.09**	**4**
LW-GR	**34.089**	**0.36**	**32**
LW-DP	23.115	$1.526e^{-06}$	1
WS-GR	**36.708**	**0.26**	**32**
WS-DP	25.303	$4.9e^{-07}$	1
GR-DP	**0.037592**	**0.85**	**1**

4.3 Classification modelling methods

In this section, we have presented 23 machine learning-based classification approaches used in this study. The existing dataset can suffer from various problems like missing values, a significant difference between the number of labelled classes (like conducive – 25, non-conducive – 100), the low accuracy rate of data collection process opted, identical values of parameters, etc. Traditional statistical approaches may not be suitable here since the relation between predicted variables, and the response variable is non-linear. So, it can be assumed that using machine learning-based classifiers may lead us towards better prediction accuracy. A lot of research work has been carried out in the field of plant disease prediction with the use of machine learning classification. Many researchers have applied a wide range of machine learning-based classification algorithms like SVM (Yao et al., 2009; Rumpf et al., 2010; Römer et al., 2011; Tian et al., 2012), LR (Steddom et al., 2003), Naïve Bayes Classifier and Decision Tree (Sankaran and Ehsani, 2012), *k*-nearest neighbour (*k*NN) (Bauer et al., 2011; Sankaran et al., 2011) LDA and QDA (Sankaran et al., 2011) for plant disease prediction. In this research, different classification algorithms have been used to predict powdery mildew disease in tomato plants. These classification algorithms are summarised in Table 7, including three decision tree approaches, two discriminant classifiers, six SVM classification algorithms, six *k*NN classifiers, five ensemble techniques, and one regression classifier. These classifiers are explained in detail as follows:

4.3.1 Decision tree

The decision tree is the most popular machine-learning based classification algorithm, which was first introduced by Quinlan in 1993 (Quinlan, 1993). It is a tree data structure in which each non-leaf/internal node symbolises a test on the variables, each external/leaf node contains the label for class, and each branch shows the result of the test (Kaur and Gosain, 2018; Zouggar and Adla, 2019). As shown in Figure 3, the decision tree divides the whole dataset into mutually exclusive spaces. Each space has a class label that describes all the data points associated with the dataset.

4.3.2 Discriminant analysis

Discriminant analysis is a machine learning approach that finds a set of equations based on predicted variables used for the classification of individual data points into priory known groups or clusters (Ramayah et al., 2010). The main goal of discriminant analysis is to find out the best predictive equation for labelling new data point and use the same equation to understand the correlation among the variables. Discriminant analysis is performed by assigning individual weight for each variable and can be written as equation (1). The linear combination of discriminant analysis is termed as the discriminant function.

Figure 4 shows the graphical representation for the discriminant function of two groups G1 & G2.

$$X_{iM} = d_{0i} + d_{1i} Z_{1M} + \cdots + d_{ji} Z_{jM} \tag{1}$$

X_{iM}: discriminant score for object M of discriminant function

Z_{vM}: predicted (Independent) variable v for object M, $v = 1, 2, 3 \ldots j$

d_{vi}: discriminant weight for predicted variable v and discriminant function i, $v = 1, 2, 3 \ldots j$

d_{0i}: constant of discriminant function i.

Equation (2) highlights the estimation of discriminant functions:

$$Max_v \vartheta = \frac{\sum_{gr=1}^{c} n_{gr} \left(T_{gr} - T\right)^2}{\sum_{gr=1}^{c} \sum_{j}^{n_{gr}} \left(T_{grj} - T_{gr}\right)^2} = \frac{SSD_b}{SSD_w} \tag{2}$$

ϑ: eigen value (discriminant criteria)

n_{gr}: size of group gr

T_{gr}: mean of group gr

T_{grj}: jth discriminant value of group gr

SSD_b: sum of squared deviations between groups, unexplained deviation

SSD_w: $_s$um of squared deviations within groups, unexplained deviation

Figure 3 Sample decision tree

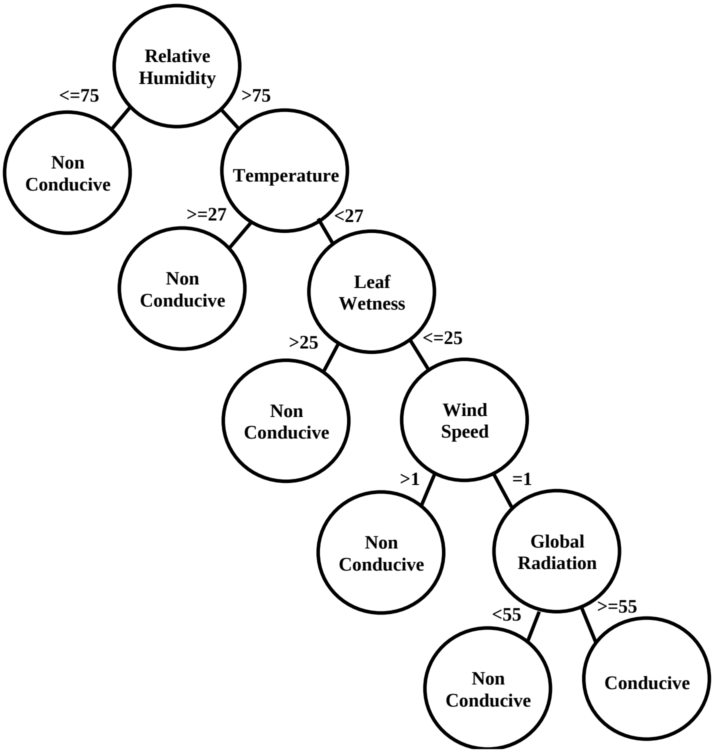
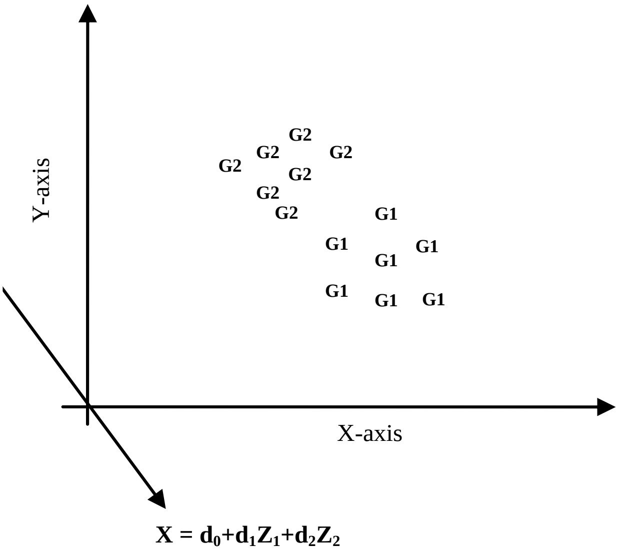


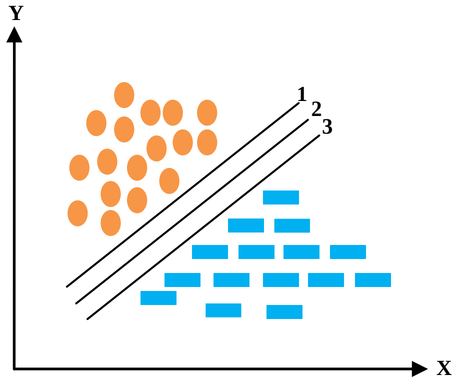
Figure 4 Graphical representation of the discriminant function



4.3.3 Support vector machine

Support vector machine (SVM) is a commonly used classification algorithm in which a point is plotted for each data item in an *f*-dimensional space where *f* indicates the number of attributes present in the dataset (Arabaci and Mohamed, 2020). Each coordinate's value indicates the value of a particular attribute. Further, classification is performed by finding the best hyperplane, which segregates the two labelled classes precisely. The best hyperplane is one that can divide the two classes with a maximum marginal difference. Figure 5 shows an SVM in which we have three hyperplanes (1, 2 & 3), and each divides two classes circle and rectangle accurately but we can notice that the marginal distance for hyperplane 2 is high compared to 1 and 3. Hence, the hyperplane 2 is said to be the perfect one because the hyperplane with maximum marginal distance has less chances of misclassification.

Figure 5 Support vector machine (see online version for colours)



4.3.4 kNN

*k*NN is the most used classification algorithm in which class or label of a new data point is determined by the majority votes of its nearest neighbours (Vishwakarma and Dalal, 2020). The data-point is assigned to the class, which is most common amongst its *k*-nearest neighbours (Kaur and Gosain, 2018). These *k*-nearest neighbours are evaluated by the Euclidean function (most commonly used), as shown in equation (3), where *s* and *t* are the two data points which we have to compare, each having *n* attributes. The term s_1 refers to the value of the first attribute of *s* data-point, while term t_1 denotes the value of the first attribute of data-point *t* (Vishwakarma et al., 2018).

$$\text{distance}(s,t)=\sqrt{(s_1-t_1)^2+(s_2-t_2)^2+\cdots+(s_n-t_n)^2} \tag{3}$$

4.3.5 Logistic regression (LR)

The main aim of LR methodology is to introduce the most efficient and best fit model (yet biologically reasonable) for demonstrating the relationship between the categorical dependent variable and a set of predicted (independent) variables (Gosain and Singh, 2019). Prediction obtained from the LR model presents the probabilities of successful and unsuccessful events for the collection of independent variables. If *Y* is a response variable and X_1 to X_n are predicted variables than the equation of LR can be written as follows (equation (4)):

$$\ln\left[\frac{p(Y)}{1-p(Y)}\right]=\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_nX_n \tag{4}$$

In the above equation, the left-hand side is known as log odd or logit function, and *p* (*Y*)*/1-p* (*Y*) is termed as odds. Odds denote the ratio of the probability of success to failure. On the other hand, the right-hand side β_0 to β_n represent regression coefficients. These must be calculated to determine probabilities to predict whether the response variable *Y* is 0 or 1. These coefficients can be computed using a popular approach known as Maximum Likelihood Estimation. On taking the inverse of equation (4), we get:

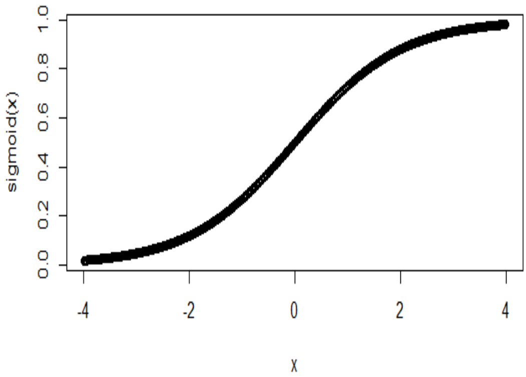
$$p[Day\text{Prediction}]=\frac{e^{\beta_0+\beta_1X_1+\beta_2X_2+\ldots+\beta_nX_n}}{1+e^{\beta_0+\beta_1X_1+\beta_2X_2+\ldots+\beta_nX_n}} \tag{5}$$

The above function (equation (5)) is known as the sigmoid function, which gives the value of probabilities within the range of 0 and 1. If the value of p comes out to be greater than 0.5, then the value of response variable Y is 1; otherwise, it is 0. Figure 6 denotes a sample sigmoid curve.

4.3.6 Ensemble classification algorithms

Ensemble algorithms are made up of various classification-based machine learning techniques to produce one optimum classifier (Kaur and Gosain, 2018). These techniques are significantly more precise than the individual classifier that makes them up. Various algorithms like Boosting, Bagging, and stacking are some standard ensemble techniques.

Figure 6 Sigmoid curve



4.4 Classifier evaluation techniques

This section highlights various classifier evaluation techniques used in this paper, including cross validation, accuracy metrics, and an overview of statistical Friedman and Nemenyi test.

4.4.1 Cross-validation technique

The cross-validation technique is used for improving the performance of classification models. In this study, we have used the 10-fold cross-validation technique. 10-fold cross-validation technique is applied by randomly dividing the available dataset into 10 equivalent size of sample data points (Mohammadpoor et al., 2020). From these 10 sample data points, only one data point is reserved as the validation data point for testing the classifier and the rest of the 9 data points are used as training data. The cross-validation process occurs 10 times (fold) iteratively, with each of the 10 data points used exactly once as the validation data. The results achieved from 10-folds validation are then averaged out to produce a distinct outcome. The main reason to use this technique is that all the data points are used for validation and training, which can further produce the best test results.

Table 7 Description of classification algorithms

Classification method	*Brief description*
1. Decision tree (Kotsiantis, 2013)(The MathWorks, 2019)	
a) Fine tree (FT)	FT classifier is highly flexible and for fine classification, it has many numbers of leaves. Usually, the maximum number of splits in the FT is 100
b) Medium tree (MT)	MT classifier is moderately flexible and for finer classification, it has medium numbers of leaves. Usually, the maximum number of splits in the MT is 20
c) Coarse tree (CT)	CT classifier has low flexibility and for coarse classification, it has very less numbers of leaves. Usually, the maximum number of splits in the CT is 4

Table 7 Description of classification algorithms (continued)

Classification method	*Brief description*
2. Discriminant classifier (Ramayah et al., 2010; The MathWorks, 2019)	
a) Linear discriminant analysis (LDA)	LDA has very low flexibility and it creates linear boundaries between the labelled classes
b) Quadratic discriminant analysis (QDA)	QDA also has very low flexibility, but it creates nonlinear boundaries (parabola, ellipse, and hyperbola) between the labelled classes
3. Regression classifier	
Logistic regression (LR)	LR (Kleinbaum et al., 2002) has low flexibility and we cannot alter any parameters to control the flexibility of the classifier
4. Support vector machines (SVM) (Durgesh and Lekha, 2010; The MathWorks, 2019)	
a) Linear SVM (LSVM)	LSVM classifier is easy to interpret but has low flexibility. It makes a linear separation between the labelled classes
b) Quadratic SVM (QSVM)	QSVM classifier is hard to interpret but has medium flexibility. It makes the quadratic separation between the labelled classes
c) Cubic SVM (CSVM)	CSVM classifier is hard to interpret but has medium flexibility. It makes a cubic separation between the labelled classes
d) Fine Gaussian SVM (FGSVM)	FGSVM classifier is hard to interpret but is highly flexible. This makes a finely detailed distinction between labelled classes with kernel scale = sqrt (PV)/4, where PV represents the number of predicted variables
e) Medium Gaussian SVM (MGSVM)	MGSVM classifier is also hard to interpret and has medium flexibility. This makes a medium distinction between labelled classes with kernel scale = sqrt (PV), where PV represents the number of predicted variables
f) Coarse Gaussian SVM (CGSVM)	CGSVM classifier is also hard to interpret and has low flexibility. This makes a medium distinction between labelled classes with kernel scale = sqrt (PV)*4, where PV represents the number of predicted variables
5. k-Nearest Neighbours (kNN) (Cunningham and Delany, 2007; The MathWorks, 2019)	
a) Fine *k*NN (F *k*NN)	F*k*NN classifier is hard to interpret and provide a finely detailed distinction between the labelled classes. The number of neighbours (k = 1) are set to 1 for this classifier
b) Medium *k*NN (M *k*NN)	M*k*NN classifier is hard to interpret and provide a medium distinction between the labelled classes. The number of neighbours are set to 10 (k = 10) for this classifier
c) Coarse *k*NN (C*k*NN)	C*k*NN classifier is hard to interpret and provide a coarse distinction between the labelled classes. The number of neighbours are set to 100 (k = 100) for this classifier
d) Cosine *k*NN (Cos*k*NN)	Co*k*NN classifier is hard to interpret and provide a medium distinction between the labelled classes using a cosine distance metric. The number of neighbours are set to 10 (k = 10) for this classifier
e) Cubic *k*NN (Cub*k*NN)	Cub*k*NN classifier is hard to interpret and provide a medium distinction between the labelled classes using a cubic distance metric. The number of neighbours are set to 10 (k = 10) for this classifier
f) Weighted *k*NN (W*k*NN)	W*k*NN classifier is hard to interpret and provide a medium distinction between the labelled classes using a distance weight metric. The number of neighbours are set to 10 (k = 10) for this classifier

Table 7 Description of classification algorithms (continued)

Classification method	*Brief description*
6. Ensembles techniques(The MathWorks, 2019)	
a) Boosted tree (BTT)	Boosted Tree classifier is hard to interpret and its flexibility changes from medium to high with increases in the number of learners or the maximum number of splits. AdaBoost technique with decision tree learners is used for building this classifier
b) Bagged tree (BGT)	Boosted Tree classifier is hard to interpret and its flexibility is very high. Random Forest Bag technique with decision tree learners is used for building this classifier
c)Subspace discriminant (SD)	SD classifier is hard to interpret and its flexibility is medium, which can improve with the increase in the number of learners. Subspace technique with discriminant learners is used for building this classifier
d) Subspace *k*NN (S*k*NN)	S*k*NN classifier is hard to interpret and its flexibility is medium, which can improve with the increase in the number of learners. The subspace technique with the nearest neighbour learners is used for building this classifier
e) RUSBoosted trees (RBT)	RBT classifier is hard to interpret and its flexibility is medium, which can improve with the increase in the number of learners or the maximum number of splits. RUSBoosted technique with decision tree learners is used for building this classifier

4.4.2 Accuracy metrics

The development of a classification model is not useful until it provides an accurate prediction for the concerned problem domain. Therefore, measurement of the prediction accuracy of a model becomes an essential task for the researchers. To perform this task, they need to answer a critical question “How can they measure the accuracy of a particular prediction model?” Many such accuracy metrics exist which have been used by scientists from time to time to evaluate the quality of their prediction model.

The most commonly used accuracy metrics are:

Confusion matrix: Correctness of a classifier can be measured by calculating the following parameters:

- the number of correctly identified instances that do not belong to the class (true negatives (TNeg))
- the number of correctly identified cases that belong to the class (true positives (TPos))
- the number of examples that were either incorrectly assigned to the class (false positive (FPos)) or not identified as a class instance (false negative (FNeg)).

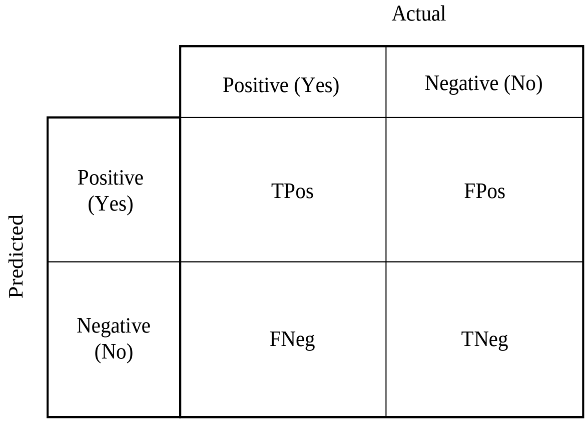
The parameters mentioned above construct a confusion matrix. Figure 7 represents the sample confusion matrix. Confusion matrix is not itself an accuracy metric, but more or less, all other metrics depend on the confusion matrix and the value of parameters inside it (Bhatia et al., 2020b).

Classification accuracy: Classification accuracy (CA) (equation (6)) is the ratio between the number of accurate predictions made by the classification model and the total number of predictions (Bhatia et al., 2020b).

$$CA = \frac{TPos + TNeg}{TPos + TNeg + FPos + FNeg} \tag{6}$$

CA alone is not a good measure for classification, so more accuracy metrics should also be considered for evaluating the performance of a classifier.

Figure 7 Confusion matrix



		Actual	
		Positive (Yes)	Negative (No)
Predicted	Positive (Yes)	TPos	FPos
	Negative (No)	FNeg	TNeg

Precision: Precision (equation (7)) is the ratio between the number of correctly identified cases (*TPos*) and the total number of correct predictions (Sharma and Jain, 2019). It is also termed as the positive predicted value. High precision denotes that a classifier gives considerably more relevant outcomes than irrelevant ones and it relates to a low false-positive rate.

$$precision = \frac{TPos}{TPos + FPos} \tag{7}$$

Recall/sensitivity: Recall (equation (8)) is the ratio between the number of correctly identified cases (*TPos*) and the summation of the number of cases that not defined as class instances (*FNeg*) and the number of correctly identified cases (*TPos*). It is also called the true positive rate and detection rate (Wankhade and Jondhale, 2019). High recall denotes that a classifier gives most of the relevant results, which relates to the low false-negative rate. A test that has 100% Sensitivity will 100% correctly identify that the weather conditions are conducive to the powdery mildew disease in this study.

$$sensitivity = \frac{TPos}{TPos + FNeg} \tag{8}$$

Specificity: Specificity (equation (9)) is the ratio between the number of correctly identified instances that do not belong to the class (*TNeg*) and the summation of the number of incorrectly identified cases (*FPos*) and the number of correctly identified instances that do not belong to the class (*TNeg*). It is also known as the true negative rate.

A test that has 100% Specificity will 100% correctly identify that the weather conditions are non-conducive to the powdery mildew disease in this study.

$$specificity = \frac{TNeg}{TNeg + FPos} \tag{9}$$

F1-score: F1-score (equation (10)) is also called as the F-Measure or F-Score. F1-Score of a classifier is the best with value 1 and worst with 0 (Sharma and Jain, 2019). It is the weighted average of the recall and precision which can be calculated through the following formula:

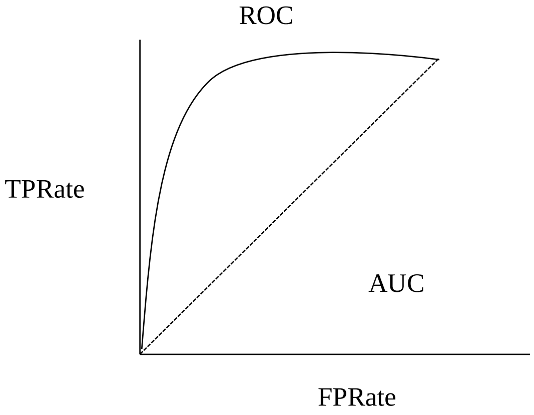
$$f1 - Score = \frac{2*\left(precision * recall\right)}{\left(precision + recall\right)} \tag{10}$$

Matthews Correlation Coefficient (*MCC*)*:* MCC (equation (11)) uses all the parameters of the confusion matrix in its formula (Boughorbel et al., 2017). Its range lies between the –1 and + 1. A classification model with value –1 is an imperfect model, and + 1 is a perfect model.

$$MCC = \frac{TPos * TNeg - FPos * FNeg}{\sqrt{\left(TPos + FPos\right)*\left(TPos + FNeg\right)*\left(TNeg + FPos\right)*\left(TNeg + FNeg\right)}} \tag{11}$$

Area under curve (AUC) *– Receiver operating characteristics (ROC*) *curve:* AUC-ROC curve is one of the essential accuracy metrics for evaluating the performance of any model where AUC denotes the degree of separability and ROC is a type of probability curve (Aarti et al., 2019). The highest value of AUC shows that the classifier has the best prediction degree. The best classifier has AUC value close to 1, which means it has finest measure of separability. In contrast, AUC value close to 0 shows that the weakest measure of separability and AUC value equals 0.5, indicating that the classifier has no class segregate ability. The ROC curve is plotted between the True Positive Rate (TPRate) and the False Positive Rate (FPRate). Figure 8 shows a ROC curve with TPRate is on the y-axis, and FPRate is on the x-axis.

Figure 8 AUC-ROC curve



4.5 Friedman test

Friedman test is a non-parametric, distribution-free test which is used to rank different techniques by identifying the significant difference amongst the performance of those techniques (Friedman, 1940). Friedman test is applied here to rank the performance of various classification algorithms used in this study. Following are the hypothesis made before conducting this test on the results:

Null hypothesis (H_0): There is no difference between the performances of various classification algorithms used in this study.

Alternative hypothesis (H_1): There is a significant difference exists among the performance of various classification algorithms used in this study.

Friedman measure can be evaluated using the following formula:

$$\chi_r^2 = \left(\frac{12}{Pw(w+1)} \sum_{i=1}^{w} R^2 \right) - 3P(w+1) \tag{12}$$

In equation (12), P represents number of performance measures used in this study; R stands for the average rank of each participant classification algorithm and w is the number of classification algorithms considered for ranking. After applying the values of these parameters in equation (12), we get the value of $\chi_{calculated}$ which is further compared with the value of $\chi_{tabulated}$ present in the chi-square distribution table. If the value of $\chi_{calculated}$ lies in the critical region it means value of $\chi_{calculated}$ is larger than the value of $\chi_{tabulated}$ then H_0 is rejected whereas H_1 is accepted and through this it can be concluded that there exists significant difference among the performance of classification algorithms. Else, H_0 is accepted and H_1 is rejected and through this it can be concluded that there does not exist any significant difference among the performance of classification algorithms.

Subsequently, every classification technique is also ranked individually with Friedman's Individual Rank (FIR) formula as shown below in equation (13):

$$FIR = \frac{C}{P} \tag{13}$$

In the above equation, C stands for cumulative rank and P represents the number of performance measures used in this study. In the current study, the classification technique with the largest value of *FIR* is termed as the best performer and the technique with the smallest value of *FIR* is considered as the worst performer. If the mean ranks of classification algorithms obtained from the FIR formula are significant, then it is also necessary to apply the Nemenyi test to check whether the mean rank difference is statistically significant.

4.6 Nemenyi test

Nemenyi Test is a post-hoc analysis that is used to compare the performance of different classification algorithms used in this study to find whether there is any statistically significant difference exists among those techniques or not (Lessmann et al., 2008).

Initially, the Critical Difference (*CD*) is evaluated using equation (14) which depends on the number of classification algorithms and the number of performance measures and the level of significance.

$$CD = q_a \sqrt{\frac{w(\chi)}{6P}} \tag{14}$$

In the above equation, *P* represents the number of performance measures; *w* stands for the total number of classification algorithms and q_a represents the critical value measured by the Studentised range statistics that was suggested by Demsar in his study (Demšar, 2006). After the evaluation of *CD*, during post-hoc analysis the difference between the *FIR* values of various pairs of classification algorithms are computed for the performance comparison of every possible pair of classification algorithms. If the calculated difference between the pairs of algorithms comes out to be superior or equal to *CD*, then it can be concluded that the performance of that specific pair is statistically significant. Else, we can say that the performance of that specific pair is not statistically significant.

5 Results and discussion

This section highlights the effects of plant disease classification models developed using 23 machine learning classification algorithms for the TPMD dataset. We report the quality of all the classifiers using the accuracy metrics mentioned in Section 4.4.2. Table 8 shows the value of parameters selected for each of the classification techniques. Explanation of each parameter has already been mentioned in Section 3. Further, Table 9 displays the outcomes for various accuracy metrics for the available powdery mildew dataset of the tomato plant. These accuracy metrics are described in Section 4.4.2. The following section summarises the observation of Table 9 and answers to all the RQs:

*RQ*1: Can classification techniques used in this study be successfully applied for precise prediction of powdery mildew disease in tomato plants?

Different performance measures, i.e., CA, Precision, Recall, Specificity, F1-score, MCC, and AUC have been used to analyse the performance of different classification algorithms used in this study for TPMD dataset using equations (6)–(11).

Table 8 Parameters values for each classification technique

Classifier name	*Parameters name*	*Values of parameters*
	Decision tree	
FT	Maximum number of splits (ψ), Split Criterion (φ)	ψ = 4, φ = Gini Index
MT		ψ = 20, φ = Gini Index
CT		ψ = 100, φ = Gini Index
	Discriminant classifiers	
LDA	Discriminant type (c)	c = Full
QDA		c = Full

Table 8 Parameters values for each classification technique (continued)

Classifier name	*Parameters name*	*Values of parameters*
	Support vector machine	
LSVM	Kernel function (K_f), Kernel Scale (£), Box Constraint Level (*ƀ*)	K_f = Linear, £= 1, *ƀ* = 1
QSVM		K_f = Quadratic, £= 1, *ƀ* = 1
CSVM		K_f = Cubic, £= 1, ƀ = 1
FGSVM		K_f = Gaussian, £= 0.56, *ƀ* = 1
MGSVM		K_f = Gaussian, £= 2.2, *ƀ* = 1
CGSVM		K_f = Gaussian, £= 8.9, *ƀ* = 1
	k-nearest neighbour	
F*k*NN	Number of neighbours (*k*), distance metric (*Đ*$_\mathrm{M}$), distance weight (*Đ*$_\mathrm{W}$)	*k* = 1, *Đ*$_\mathrm{M}$ = Euclidean, *Đ*$_\mathrm{W}$ = Equal
M*k*NN		*k* = 10, *Đ*$_\mathrm{M}$ = Euclidean, *Đ*$_\mathrm{W}$ = Equal
C*k*NN		*k* = 100, *Đ*$_\mathrm{M}$ = Euclidean, *Đ*$_\mathrm{W}$ = Equal
Cos*k*NN		*k* = 10, *Đ*$_\mathrm{M}$ = Cosine, *Đ*$_\mathrm{W}$ = Equal
Cub*k*NN		*k* = 10, *Đ*$_\mathrm{M}$ = Minkowski (cubic), *Đ*$_\mathrm{W}$ = Equal
W*k*NN		*k* = 10, *Đ*$_\mathrm{M}$ = Euclidean, *Đ*$_\mathrm{W}$ = Squared Inverse
	Ensembles techniques	
BTT	Ensemble method (*È*), learner type (*l*), maximum number of splits (*ψ*), number of learners (*ʒ*), learning rate (*ɳ*)	*È* = AdaBoost, *l* = Decision tree, *Ψ* = 20, *ʒ* = 30, *ɳ* = 0.1
BGT	Ensemble Method (*È*) Learner Type (*l*), Number of Learners (*ʒ*)	*È* = Bag, *l* = Decision Tree, *ʒ* = 30
SD	Ensemble method (*È*) learner type (*l*), number of learners (*ʒ*)	*È* = Subspace, *l* = Discriminant, *ʒ* = 30
S*k*NN	Ensemble method (*È*), learner type (*l*), number of learners (*ʒ*)	*È* = Subspace, *l* = Nearest neighbours, *ʒ* = 30
RBT	Ensemble method (*È*), learner type (*l*) Maximum number of splits (*ψ*) Number of learners (ʒ), learning rate (ɳ)	*È* = RUSBoost, *l* = Decision tree, *Ψ* = 20, *ʒ* = 30, *ɳ* = 0.1

Table 9 shows the performance of each algorithm based on the above mentioned accuracy measures. So, it can be stated that, we have successfully applied all the 23 classification algorithms on the TPMD dataset. It is quite clear from Table 9 that LSVM is most accurate in classification process if precision is taken as the performance measure, whereas MGSVM is most accurate if F1-score and MCC are considered as performance measures. Both LSVM and MGSVM are most accurate in terms of CA, whereas LR, QSVM, CSVM, F*k*NN, W*k*NN, and S*k*NN all are best performers in terms of recall. If the classifiers are compared using Specificity, then CGSVM, C*k*NN, BTT, and SD are most accurate, whereas, in terms of AUC, W*k*NN is the best performer.

Discussion: After analysing the performance of all the 23 machine learning algorithms on the basis of different accuracy measures, it can be stated that no meaningful conclusion can be drawn from different performance metrics as shown in Table 9 as they produce very divergent results. These results can be validated with one of the study of Kitchenham et al. (1999) which claimed that the different accuracy measures produce different outcomes about the supremacy of one classifier over the other. The results are inconclusive because of the highly imbalanced structure of the TPMD dataset as mentioned in Sections 4.1 and 4.2. So, in this study we compare the proposed classifiers by performing Friedman statistical test on all the seven accuracy metrics as explained in RQ$_2$.

Table 9 Performance of various classifiers for TPMD dataset

S. No.	*Techniques*	*Classification accuracy*	*Precision*	*Recall* (Sensitivity)	*Specificity*	*F1-Score*	*MCC*	*AUC*
1	FT	0.9263	0.6923	0.7500	0.9518	0.7200	0.6825	0.8509
2	MT	0.9263	0.6923	0.7500	0.9518	0.7200	0.6825	0.8509
3	CT	0.9263	0.6923	0.7500	0.9518	0.7200	0.6825	0.8509
4	LDA	0.8632	0	0	0.9880	0	–0.0369	0.4940
5	QDA	0.9053	0.6667	0.5000	0.9639	0.5714	0.5169	0.7319
6	LR	0.9263	0.6316	1	0.9157	0.7742	0.7947	0.9578
7	LSVM	0.9474	0.8182	0.7500	0.9759	0.7826	0.7492	0.8629
8	QSVM	0.9263	0.6316	1	0.9157	0.7742	0.7947	0.9578
9	CSVM	0.9158	0.6000	1	0.9036	0.7500	0.7746	0.9518
10	FGSVM	0.8737	0.5000	0.0833	0.9880	0.1429	0.1558	0.5356
11	MGSVM	0.9474	0.7333	0.9167	0.9518	0.8148	0.8059	0.9342
12	CGSVM	0.8737	0	0	1	0	0	0.5000
13	F*k*NN	0.9158	0.6000	1	0.9036	0.7500	0.7746	0.9518
14	M*k*NN	0.9263	0.0789	0.9167	0.9277	0.7586	0.7549	0.9221
15	C*k*NN	0.8737	0	0	1	0	0	0.5000
16	Cos*k*NN	0.9053	0.5882	0.8333	0.9157	0.6897	0.6696	0.8744
17	Cub*k*NN	0.9368	0.6875	0.9167	0.9398	0.7857	0.7792	0.9282
18	W*k*NN	0.9368	0.6667	1	0.9277	0.8000	0.8165	0.9638
19	BTT	0.8737	0	0	1	0	0	0.5000
20	BGT	0.9368	0.6875	0.9167	0.9398	0.7857	0.7792	0.9282
21	SD	0.8737	0	0	1	0	0	0.5000
22	S*k*NN	0.9263	0.6316	1	0.9157	0.7742	0.7947	0.9578
23	RBT	0.9158	0.6429	0.7500	0.9398	0.6923	0.6544	0.8448

*RQ*2: What is the comparative performance of different classification techniques used for the prediction of powdery mildew disease in tomato plants?

We conducted the Friedman test in the current study to compare the performance of 23 different classification algorithms applied on the TPMD dataset based on seven performance measures. Friedman test, a non-parametric test has been used to figure out whether there is any statistical dissimilarity between the accuracy of these classifiers during the prediction of powdery mildew disease in tomato plants. The results of the Friedman test for the available dataset are shown in Table 10. This table indicates that the MGSVM classifier is ranked higher in the prediction of powdery mildew disease in the tomato plant in terms of all seven accuracy measures.

It is also concluded from Table 10 that for the available powdery mildew dataset, Friedman test ranks MGSVM, W*k*NN, BGT, Cub*k*NN, LSVM, S*k*NN, QSVM, LR, CT and MT as top 10 most precise classifiers for the accurate detection of powdery mildew disease in tomato plants. LDA is ranked as the least precise classifier for powdery mildew prediction. These results help us to decide that there exist some classifiers that are most appropriate and truthful to the dataset of powdery mildew disease in tomato plants for detecting disease.

By using equation (14), the value of critical region for the level of significance equal to 5% and degree of freedom equal to 22, i.e., 23 classification algorithms minus 1 (or *w*–1, where *w* is a total number of classification algorithms used in this study) is also calculated. Value for $\chi_{tabulated}$ is read from the Chi-square table corresponding to 95% significance level and degree of freedom equal to 22.

Table 10 Mean ranking of classifiers on applying Friedman test for different performance measures

Classification technique	*Mean rank*	*Classification technique*	*Mean rank*
LDA	4.29	FT	12.93
CGSVM	5.93	MT	12.93
C*k*NN	5.93	CT	12.93
BTT	5.93	LR	15.71
SD	5.93	QSVM	15.71
FGSVM	7.71	S*k*NN	15.71
Cos*k*NN	9.00	LSVM	16.64
QDA	9.57	Cub*k*NN	16.64
RBT	9.86	BGT	16.64
M*k*NN	12.43	W*k*NN	18.79
F*k*NN	12.71	MGSVM	19.36
CSVM	12.71		

According to the null hypothesis of the Friedman test which states that no significant difference exists between the performances of various classification algorithms used in this study, it was found that at 0.05 level of significance, $\chi_{calculated}$ which is actually the

Friedman measure lies in the critical region for the TPMD dataset. Therefore, it is concluded by accepting the alternative hypothesis and rejecting the null hypothesis that a significant difference exists between the performances of various classification algorithms used in this study. Table 11 depicts the test statistics for the Friedman test for the accuracy metrics' values in the powdery mildew dataset.

Table 11 Test statistics of the Friedman test

N	*7*
Chi-Square	70.752
df	22
Asymp. Sig	0.000

Discussion*:* As reported by the results of the Friedman tests, the MGSVM classifier significantly outperforms all other classifiers for developing plant disease prediction models when evaluated using all seven accuracy metrics explained in Section 4.4.2. MGSVM is a special case of the SVM algorithm where the Gaussian kernel is used with kernel set equal to sqrt(G), where G indicates the number of independent variables used during training of the model. The favourable nature of MGSVM could be due to the classifier's efficient learning ability that helps in reducing error during classification task. Also, MGSVM is highly robust towards outliers as well as it is a well-suited classifier for binary classification. The Friedman test results lead to the conclusion that there is a classifier that can give a highly precise and significant performance for predicting powdery mildew disease in tomato plants.

*RQ*3: Which pairs of classification techniques perform significantly different from each other in terms of performance measures?

After Friedman test, post-hoc analysis using the Nemenyi test is performed to check if the differences that exist between the performances of various classification algorithms based on *FIR* values as concluded in RQ_2 above are statistically significant or not. The value for *CD* is calculated to be equal to 3.6 using equation (14) where w is taken to be 23 (number of classification algorithms), and P is taken to be 7 (number of performance measures). After this, all the possible pairs of classification algorithms are formed with every other classification algorithm for calculating the rank differences between them, i.e., between the *FIR* values so obtained. Here, 253 such combinations are formed for 23 different classification algorithms for the TPMD dataset and the results are compiled in Table 13.

Values for differences in ranks which are greater than or equal to *CD*, i.e., 3.6 are shown in bold in Table 13. It is observed that 157 out of 253, i.e., 62.05% of the total pairs of classification algorithms have been highlighted, which means 62.05% of the pairs have the difference above or equal to *CD* showing that the performance of these pairs is found to be significantly different using Nemenyi test. However, the differences between the performances of all other pairs of classification algorithms have not been found significant.

Discussion: Nemenyi test has been conducted in this study to verify the results of Friedman Test. As already proven in RQ$_2$, MGSVM performed the best as compared to all other classifiers. The results of the Nemenyi test also reveal that the MGSVM classifier performed significantly different than nineteen other techniques. Only, four classifiers, namely, LSVM, Cub*k*NN, BGT, and W*k*NN do not show significantly different performance than MGSVM. So, from the findings of the Nemenyi test, it can be concluded that MGSVM performed significantly different and better than 19 other techniques. The other four classifiers' results are not significantly different from MGSVM because their mean ranks are almost equal to MGSVM as shown in Table 10.

Table 12 Comparison with existing approach

Techniques	*CA*	*AUC*
	Proposed techniques	
FT	**0.9263**	0.8509
MT	**0.9263**	0.8509
CT	**0.9263**	0.8509
LDA	0.8632	0.4940
QDA	0.9053	0.7319
LR	**0.9263**	**0.9578**
LSVM	**0.9474**	0.8629
QSVM	**0.9263**	**0.9578**
CSVM	0.9158	**0.9518**
FGSVM	0.8737	0.5356
MGSVM	**0.9474**	**0.9342**
CGSVM	0.8737	0.5000
F*k*NN	0.9158	**0.9518**
M*k*NN	**0.9263**	0.9221
C*k*NN	0.8737	0.5000
Cos*k*NN	0.9053	0.8744
Cub*k*NN	**0.9368**	**0.9282**
W*k*NN	**0.9368**	**0.9638**
BTT	0.8737	0.5000
BGT	**0.9368**	**0.9282**
SD	0.8737	0.5000
S*k*NN	**0.9263**	**0.9578**
RBT	0.9158	0.8448
	Existing techniques	
IMPS-ELM	0.8991	0.8857
Hybrid SVM-LR	0.9237	0.9270

Table 13 Pair-wise rank differences between different classification algorithms for TPMD dataset in terms of seven performance measure

Techniques	*LDA*	*CGSVM*	*CkNN*	*BTT*	*SD*	*FGSVM*	*CoskNN*	*QDA*	*RBT*	*MkNN*	*FkNN*	*CSVM*	*FT*	*MT*	*CT*	*LR*	*QSVM*	*SkNN*	*LSVM*	*CubkNN*	*BGT*	*WkNN*	*MGSVM*
LDA	–	1.64	1.64	1.64	1.64	3.42	**4.71**	**5.28**	**5.57**	**8.14**	**8.42**	**8.42**	**8.64**	**8.64**	**8.64**	**11.42**	**11.42**	**11.42**	**12.35**	**12.35**	**12.35**	**14.5**	**15.07**
CGSVM		–	0	0	0	1.78	3.07	**3.64**	**3.93**	**6.5**	**6.78**	**6.78**	**7**	**7**	**7**	**9.78**	**9.78**	**9.78**	**10.71**	**10.71**	**10.71**	**12.86**	**13.43**
C*k*NN			–	0	0	1.78	3.07	**3.64**	**3.93**	**6.5**	**6.78**	**6.78**	**7**	**7**	**7**	**9.78**	**9.78**	**9.78**	**10.71**	**10.71**	**10.71**	**12.86**	**13.43**
BTT				–	0	1.78	3.07	**3.64**	**3.93**	**6.5**	**6.78**	**6.78**	**7**	**7**	**7**	**9.78**	**9.78**	**9.78**	**10.71**	**10.71**	**10.71**	**12.86**	**13.43**
SD					–	1.78	3.07	**3.64**	**3.93**	**6.5**	**6.78**	**6.78**	**7**	**7**	**7**	**9.78**	**9.78**	**9.78**	**10.71**	**10.71**	**10.71**	**12.86**	**13.43**
FGSVM						–	1.29	1.86	2.15	**4.72**	**5**	**5**	**5.22**	**5.22**	**5.22**	**8**	**8**	**8**	**8.93**	**8.93**	**8.93**	**11.08**	**11.65**
Cos*k*NN							–	0.57	0.86	3.43	**3.71**	**3.71**	**3.93**	**3.93**	**3.93**	**6.71**	**6.71**	**6.71**	**7.64**	**7.64**	**7.64**	**9.79**	**10.36**
QDA								–	0.29	2.86	3.14	3.14	3.36	3.36	3.36	**6.14**	**6.14**	**6.14**	**7.07**	**7.07**	**7.07**	**9.22**	**9.79**
RBT									–	2.57	2.85	2.85	3.07	3.07	3.07	**5.85**	**5.85**	**5.85**	**6.78**	**6.78**	**6.78**	**8.93**	**9.5**
M*k*NN										–	0.28	0.28	0.5	0.5	0.5	3.28	3.28	3.28	**4.21**	**4.21**	**4.21**	**6.36**	**6.93**
F*k*NN											–	0	0.22	0.22	0.22	3	3	3	**3.93**	**3.93**	**3.93**	**6.08**	**6.65**
CSVM												–	0.22	0.22	0.22	3	3	3	**3.93**	**3.93**	**3.93**	**6.08**	**6.65**
FT													–	0	0	2.78	2.78	2.78	**3.71**	**3.71**	**3.71**	**5.86**	**6.43**
MT														–	0	2.78	2.78	2.78	**3.71**	**3.71**	**3.71**	**5.86**	**6.43**
CT															–	2.78	2.78	2.78	**3.71**	**3.71**	**3.71**	**5.86**	**6.43**
LR																–	0	0	0.93	0.93	0.93	3.08	**3.65**
QSVM																	–	0	0.93	0.93	0.93	3.08	**3.65**
S*k*NN																		–	0.93	0.93	0.93	3.08	**3.65**
LSVM																			–	0	0	2.15	2.72
Cub*k*NN																				–	0	2.15	2.72
BGT																					–	2.15	2.72
W*k*NN																						–	0.57
MGSVM																							–

Comparison with the existing approaches: Initially, in 1997, Guzman-Plazola has developed an LDA based spray prediction model for powdery mildew disease in the tomato plant (Guzman-Plazola, 1997). This model was further validated by the weather-based TPMD dataset, which was collected by Bakeer et al. (2013) during their research. Later, Bhatia et al. (2020a) have used this TPMD dataset for powdery mildew disease prediction in tomato plant using ELM algorithm. They have used various resampling techniques namely, random over sampling (ROS), random under sampling (RUS), synthetic minority over-sampling technique (SMOTE), and importance sampling (IMPS) in their study for balancing TPMD dataset. They have found that IMPS-ELM approach performed the best in terms of CA and AUC metrics. In another study, a new Hybrid SVM-LR classifier has been proposed by Bhatia et al. (2020b) for the prediction of powdery mildew disease in tomato plants using the TPMD dataset. They have used only three performance metrics, namely, CA, F1-Score, and AUC to analyse their proposed approach. The authors of these studies have not used any statistical tests to validate their findings.

Table 12 compares the proposed technique with the exiting IMPS-ELM and Hybrid SVM-LR classifiers in terms of CA and AUC metrics. It is evident from this table that 12 out of 23 (52.17%) proposed techniques (marked in bold) have performed better than IMPS-ELM and Hybrid SVM-LR classifiers in terms of CA metric.

Similarly, 9 out of 23 (39.13%) proposed techniques (marked in bold) have shown better results than the existing Hybrid SVM-LR and IMPS-ELM approach in terms of the AUC metric. Also, in terms of CA and AUC metrics, 7 out of 23 (30.43%) proposed classifiers gave a better performance than the existing classifiers. In the current study, the best performer obtained from the Friedman test, i.e., MGSVM, outperformed hybrid SVM-LR and IMPS-ELM in terms of both CA and AUC metrics.

Hence, it can be stated that the proposed approach is helpful to find out the best classifier for the powdery mildew disease prediction in tomato plants. Also, the statistical Friedman and Nemenyi tests help in strengthen the results of the current study.

6 Conclusion and future scope

The main objective of the current study was to determine the best classifier among the various classification algorithms for predicting powdery mildew disease in tomato plants. Statistical analysis and comparison of 23 different classification algorithms was conducted using TPMD dataset. The performance of various classification algorithms was evaluated using CA, Precision, Recall, Specificity, F1-score, MCC, and AUC as the prediction accuracy measures. Further, to find out if there is a significant difference between the performances of different classification algorithms and to find their mean ranks, a non-parametric statistical test named Friedman test was conducted. Lastly, post-hoc analysis using an advanced statistical test called the Nemenyi test was also performed to identify if the difference in the performance of various classification algorithms, if it exists, is statistically significant or not. The major findings of the current study are presented below.

- After evaluating each of the seven prediction-accuracy measures for all the 23 classifiers, it was found that based on CA, LSVM and MGSVM classifier performed the best with 94.74% accuracy, whereas LDA came out to be the worst performer with an accuracy of 86.32%. Further, we also compared the performance of these classifiers with the remaining accuracy metrics.

- Further, it is shown that different accuracy measures calculated on predicted variables produce very different results regarding the superiority of one classifier over the other.
- Therefore, a statistical Friedman test is used to rank multiple classifiers based on all the performance measures.
- The results of the Friedman test and post-hoc analysis using the Nemenyi test further revealed the superiority of MGSVM over other selected classification algorithms used in the study for powdery mildew prediction in the tomato plant.

Hence, prediction models developed using various classification algorithms can successfully be implemented for the TPMD dataset. However, this is a limited implementation of the proposed study. We are planning to use Neuro-fuzzy classification techniques with evolutionary or genetic algorithms to improve performance in the future. Another research criterion can also be used to compare the 23 classifiers used in this study in terms of the efficiency and understandability of these classification models. Researchers can use more effective and proficient accuracy measures for the significant evaluation of classifiers. Wilcoxon test and McNemar's test can also be applied for the statistical analysis of various classification models to identify the best classifier for predicting the powdery mildew disease in the tomato plant.

Acknowledgement

This work is financially supported by the Department of Science and Technology (DST) under a project with reference number "DST/Reference. No. T-319/2018-19". We are grateful to them for their immense support.

References

Aarti, Sikka, G. and Dhir, R. (2019) 'Threshold-based empirical validation of object-oriented metrics on different severity levels', *International Journal of Intelligent Engineering Informatics (IJIEI),* Vol. 7, Nos. 2–3, pp.231–262.

Arabaci, H. and Mohamed, M.A. (2020) 'A knowledge-based diagnosis algorithm for broken rotor bar fault classification using FFT, principal component analysis and support vector machines', *International Journal of Intelligent Engineering Informatics* (*IJIEI*), Vol. 8, No. 1, pp.19–37.

Avenot, H.F. and Michailides, T.J. (2010) 'Progress in understanding molecular mechanisms and evolution of resistance to succinate dehydrogenase inhibiting (SDHI) fungicides in phytopathogenic fungi', *Crop Protection,* Vol. 29, No. 7, pp.643–651.

Bakeer, A.R.T., Abdel-Latef, M.A.E., Afifi, M.A. and Barakat, M.E. (2013) 'Validation of tomato powdery mildew forecasting model using meteorological data in Egypt', *International Journal of Agriculture Sciences,* Vol. 5, No. 2, p.372.

Bauer, S.D., Korč, F. and Förstner, W. (2011) 'The potential of automatic methods of classification to identify leaf diseases from multispectral images', *Precision Agriculture*, Vol. 12, No. 3, pp.361–377.

Bhatia, A., Chug, A. and Singh, A.P. (2020a) 'Application of extreme learning machine in plant disease prediction for highly imbalanced dataset', *Journal of Statistics and Management Systems*, pp.1–10, doi: 10.1080/09720510.2020.1799504.

Bhatia, A., Chug, A. and Singh, A.P. (2020b) 'Hybrid SVM-LR classifier for powdery mildew disease prediction in tomato plant', *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, Delhi-NCR, India, pp.218–223.

Boughorbel, S., Jarray, F. and El-Anbari, M. (2017) 'Optimal classifier for imbalanced data using matthews correlation coefficient metric', *PloS One*, Vol. 2, No. 6, p.e0177678.

Braun, U. (1987) *A Monograph of the Erysiphales (Powdery Mildews)*, Beihefte zur Nova Hedwigia, No. 89.

Correll, J., Gordon, T. and Elliott, V. (1988) 'The epidemiology of powdery mildew on tomatoes', *California Agriculture*, Vol. 42, No. 2, pp.8–10.

Cunningham, P. and Delany, S.J. (2007) 'k-nearest neighbour classifiers', *Multiple Classifier Systems*, Vol. 34, No. 8, pp.1–17.

Demšar, J. (2006) 'Statistical comparisons of classifiers over multiple data sets', *Journal of Machine Learning Research*, Vol. 7, January, pp.1–30.

Durgesh, K.S. and Lekha, B. (2010) 'Data classification using support vector machine', *Journal of Theoretical and Applied Information Technology*, Vol. 12, No. 1, pp.1–7.

Farag, F.M. (2003) *Studies on Powdery Mildew Disease Affecting Solanaceous Plant*, MSc Thesis, Cairo Univ, Cairo.

Friedman, M. (1940) 'A comparison of alternative tests of significance for the problem of m rankings', *The Annals of Mathematical Statistics*, Vol. 11, No. 1, pp.86–92.

Gosain, A. and Singh, J. (2019) 'Empirical investigation of dimension hierarchy sharing-based metrics for multidimensional schema understandability', *International Journal of Intelligent Engineering Informatics (IJIEI)*, Vol. 7, Nos. 2–3, pp.141–163.

Guzman-Plazola, R.A. (1997) *Development of a Spray Forecast Model for Tomato Powdery Mildew* (*Leveillula Taurica* (*Lev*). *Arn.*), University of California, Davis.

Haroun, S.H. (2002) *Studies on Powdery Mildew Disease on Tomatoes in Egypt*, PhD Thesis, Fac. Agric. Moshtohor, Zaga-zig Univ., p.166.

Holliday, P. (1980) *Fungus Diseases of Tropical Crops*, Cambridge, p.147.

Jones, W.B. and Thomson, S.V. (1987) 'Source of inoculum, yield, and quality of tomato as affected by Leveillula taurica', *Plant Disease*, Vol. 71, No. 3, pp.266–268.

Kaur, P. and Gosain, A. (2018) 'An intelligent undersampling technique based upon intuitionistic fuzzy sets to alleviate class imbalance problem of classification with noisy environment', *International Journal of Intelligent Engineering Informatics (IJIEI)*, Vol. 6, No. 5, pp.417–433.

Kitchenham, B., MacDonell, S., Pickard, L. and Shepperd, M. (1999) *Assessing Prediction Systems*, University of Otago, Dunedin, New Zealand.

Kleinbaum, D.G., Dietz, K., Gail, M., Klein, M. and Klein, M. (2002) *Logistic Regression*, Springer-Verlag, New York.

Kotsiantis, S.B. (2013) 'Decision trees: a recent overview', *Artificial Intelligence Review*, Vol. 39, No. 4, pp.261–283.

Lessmann, S., Baesens, B., Mues, C. and Pietsch, S. (2008) 'Benchmarking classification models for software defect prediction: a proposed framework and novel findings', *IEEE Transactions on Software Engineering*, Vol. 34, No. 4, pp.485–496.

Mattihalli, C., Gedefaye, E., Endalamaw, F. and Necho, A. (2018) 'Plant leaf diseases detection and auto-medicine', *Internet of Things*, Vol. 1, pp.67–73.

McGrath, M.T., Shishkoff, N., Bornt, C. and Moyer, D.D. (2001) 'First occurrence of powdery mildew caused by leveillula taurica on pepper in new york', *Plant Disease*, Vol. 85, No. 10, p.1122.

McKight, P.E. and Najab, J. (2010) 'Kruskal-Wallis test', *The Corsini Encyclopedia of Psychology*, pp.1-1.

Mohammadpoor, M., Nooghabi, M.G. and Ahmedi, Z. (2020) 'An intelligent technique for grape fanleaf virus detection', *International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI)*, Vol. 6, No. 1, pp.62–67.

Naik, H.S., Zhang, J., Lofquist, A., Assefa, T., Sarkar, S., Ackerman, D., Singh, A., Singh, A.K. and Ganapathysubramanian, B. (2017) 'A real-time phenotyping framework using machine learning for plant stress severity rating in soybean', *Plant Methods*, Vol. 13, No. 1, p.23.

Padol, P.B. and Yadav, A.A. (2016) 'SVM classifier based grape leaf disease detection', *2016 Conference on Advances in Signal Processing (CASP)*, Pune, Maharashtra, India, pp.175–179.

Palti, J. (1988) 'TheLeveillula mildews', *The Botanical Review*, Vol. 54, No. 4, pp.423–535.

Patil, S.P. and Zambre, R.S. (2014) 'Classification of cotton leaf spot disease using support vector machine', *International Journal of Engineering Research*, Vol. 3, No. 4, pp.1511–1514.

Pujari, D., Yakkundimath, R. and Byadgi, A.S. (2016) 'SVM and ANN based classification of plant diseases using feature reduction technique', *International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI)*, Vol. 3, No. 7, pp.6–14.

Pujari, J.D., Yakkundimath, R. and Byadgi, A.S. (2015) 'Image processing based detection of fungal diseases in plants', *Procedia Computer Science*, Vol. 46, pp.1802–1808.

Quinlan, J.R. (1993) *C4.5: Programs For Machine Learning*, Morgan Kaufmann San Francisco, CA, USA.

Ramayah, T., Ahmad, N.H., Halim, H.A., Zainal, S.R.M. and Lo, M-C. (2010) 'Discriminant analysis: an illustrated example', *African Journal of Business Management*, Vol. 4, No. 9, pp.1654–1667.

Reis, A., Boiteux, L.S., Paz-Lima, M.L., Silva, P.P. and Lopes, C.A. (2004) 'Powdery mildew of allium species caused by oidiopsis taurica in Brazil', *Horticultura Brasileira*, Vol. 22, No. 4, pp.758–760.

Römer, C., Bürling, K., Hunsche, M., Rumpf, T., Noga, G. and Plümer, L. (2011) 'Robust fitting of fluorescence spectra for pre-symptomatic wheat leaf rust detection with support vector machines', *Computers and Electronics in Agriculture*, Vol. 79, No. 2, pp.180–188.

Rumpf, T., Mahlein, A-K., Steiner, U., Oerke, E-C., Dehne, H-W. and Plümer, L. (2010) 'Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance', *Computers and Electronics in Agriculture*, Vol. 74, No. 1, pp.91–99.

Sankaran, S. and Ehsani, R. (2012) 'Detection of huanglongbing disease in citrus using fluorescence spectroscopy', *Transactions of the ASABE*, Vol. 55, No. 1, pp.313–320.

Sankaran, S., Mishra, A., Maja, J.M. and Ehsani, R. (2011) 'Visible-near infrared spectroscopy for detection of huanglongbing in citrus orchards', *Computers and Electronics in Agriculture*, Vol. 77, No. 2, pp.127–134.

Shankar, R., Harsha, S. and Bhandary, R. (2014) *A Practical Guide to Identification and Control of Tomato Diseases*, Tropica Seeds Pvt. Ltd.,Bangalore, Karnataka, India.

Shapiro, S.S. and Wilk, M.B. (1965) 'An analysis of variance test for normality (complete samples)', *Biometrika*, pp.591–611.

Sharma, S. and Jain, A. (2019) 'Dynamic social network analysis and performance evaluation', *International Journal of Intelligent Engineering Informatics (IJIEI)*, Vol. 7, Nos. 2–3, pp.180–202.

Steddom, K., Heidel, G., Jones, D. and Rush, C.M. (2003) 'Remote detection of rhizomania in sugar beets', *Phytopathology*, Vol. 93, No. 6, pp.720–726.

The MathWorks, I. (2019) *Choose Classifier Options*, Available at: https://in.mathworks.com/help/stats/choose-a-classifier.html (Accessed July 2020).

Tian, J., Hu, Q., Ma, X. and Han, M. (2012) 'An improved KPCA/GA-SVM classification model for plant leaf disease recognition', *Journal of Computational Information Systems*, Vol. 8, No. 18, pp.7737–7745.

Verma, S., Bhatia, A., Chug, A. and Singh, A.P. (2020) 'Recent advancements in multimedia big data computing for ioT applications in precision agriculture: opportunities, issues, and challenges', *Multimedia Big Data Computing for IoT Applications*, pp.391–416.

Verma, S., Chug, A. and Singh, A.P. (2018) 'Prediction models for identification and diagnosis of tomato plant diseases', *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Bangalore, Karnataka, India, pp.1557–1563.

Verma, S., Chug, A. and Singh, A.P. (2020) 'Application of convolutional neural networks for evaluation of disease severity in tomato plant', *Journal of Discrete Mathematical Sciences and Cryptography*, Vol. 23, No. 1, pp.273–282.

Verma, S., Chug, A., Singh, A.P., Sharma, S. and Rajvanshi, P. (2019) 'Deep learning-based mobile application for plant disease diagnosis: a proof of concept with a case study on tomato plant', *Applications of Image Processing and Soft Computing Systems in Agriculture*, pp.242–271.

Vishwakarma, V.P. and Dalal, S. (2020) 'Generalized DCT and DWT hybridization based robust feature extraction for face recognition', *Journal of Information and Optimization Sciences.*, Vol. 41, No. 1, pp.61–72.

Vishwakarma, V.P., Dalal, S. and Sisaudia, V. (2018) 'Efficient feature extraction using DWTDCT for robust face recognition under varying illuminations', *2018 2nd IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, New Delhi, Delhi, India, pp.982–987.

Wankhade, K.K. and Jondhale, K.C. (2019) 'An ensemble clustering method for intrusion detection', *International Journal of Intelligent Engineering Informatics (IJIEI)*, Vol. 7, Nos. 2–3, pp.112–140.

Yao, Q., Guan, Z., Zhou, Y., Tang, J., Hu, Y. and Yang, B. (2009) 'Application of support vector machine for detecting rice diseases using shape and color texture features', *2009 International Conference on Engineering Computation*, Hong Kong, China, pp.79–83.

Zouggar, S.T. and Adla, A. (2019) 'A diversity-accuracy measure for homogenous ensemble selection', *International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI)*, Vol. 5, No. 5, pp.63–70.