International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

"Enhancing Precision in Tulsi Leaf Infection Classification: A Stacking Classifier Ensemble Strategy"

Manjot Kaur[1*], Someet Singh[2], Anita Gehlot[3]

Submitted: 16/01/2024 **Revised**: 24/02/2024 **Accepted**: 02/03/2024

Abstract: Tulsi, often known as Holy Basil (*Ocimum sanctum*), is a herb with cultural significance and health benefits. Researchers and farmers have been concerned about the prevalence of illnesses that harm tulsi leaves in recent years. In order to determine the best model for early identification and intervention, we do a thorough evaluation of many stacking classifier algorithms for the prediction of tulsi leaf diseases in this work. The collection of tulsi leaf imagery in the dataset is broad and includes labels designating various disease conditions. We investigate the efficacy of stacking classifiers by employing a blend of foundational models, such as Decision Trees (DT), Random Forest (RF), Support Vector Machines (SVM), and Multi-layer Perceptron (MLP) Classifier. Every base model offers a different perspective on the traits connected to both healthy and infected tulsi leaves. We compare various stacking classifier setups based on their accuracy, precision, recall, and F1 score. We take into account differences in the makeup of base models and how model performance is affected by hyperparameter adjustment. Furthermore, we use cross-validation methods to evaluate the five models' generalizability. Farmers and researchers can rely on the model B that is found to have the best predictive performance with an average accuracy of 98.25%, since it provides a strong means of early disease diagnosis and management. This study advances precise farming methods, encourages tulsi cultivation that is sustainable, and guarantees the plant's continued use in traditional medicine.

Keywords: *Stacking Classifier, Precision Agriculture, Machine Learning, Image Classification*

1. Introduction

In India, tulsi is a commonly grown herb with both medicinal and cultural value [1]. It is produced in farms and backyard gardens frequently, and it's used in traditional Ayurvedic treatment. In India, the amount of tulsi produced varies according to demand, climate, and local farming methods. India was one of the world's top producers of tulsi as of 2021[2]. The plant is cultivated in several states throughout the nation, and both conventional and innovative agricultural techniques have been used to enhance its yield. Precision farmers can target specific portions of a field that exhibit disease indicators by implementing classifier models. Utilising resources like pesticides, water, and fertilisers more effectively is made possible by this targeted strategy. Cost reductions may come from early detection and focused treatment based on classifier predictions. It minimises the requirement for treatments to be applied widely throughout the field, maximising resource utilisation and lowering environmental impact. Large leaf image datasets may be processed quickly and effectively by automated systems driven by classifier models, which makes them scalable for crop monitoring over vast agricultural regions [3].

Classifier models provide an impartial and standardised evaluation of leaf health by generating predictions based on pre-established patterns and attributes. This guarantees a more standardised review procedure and lessens the unpredictability brought about by human observers. For extensive crop health monitoring, classifier models can be incorporated into remote sensing technologies like drones or satellite photos [4]. A more comprehensive perspective of agricultural landscapes is possible with this method. Classifier models that examine patterns and trends in leaf health data over time might produce insightful results. Making educated judgements regarding resource allocation, crop management, and disease prevention techniques is possible with the use of this knowledge. Internet of Things (IoT) devices, like field-positioned sensors and cameras, can be combined with classifier models [5]. The model can interpret real-time data from various devices so that decisions can be made right away. In conclusion, classifier models are essential for agricultural plant disease detection and leaf health prediction. They facilitate data-driven decision-making, focused interventions, and early detection, all of which lead to more effective and sustainable farming methods.

[1*]*Lovely Professional University, School of Electrical and Electronics Engineering, Phagwara, India*
[2,1]*Lovely Professional University, School of Electrical and Electronics Engineering, Phagwara, India*
[3]*Uttaranchal University, Uttaranchal Institute of Technology, Dehradun, India*
manjotkaur.hpk@gmail.com[1], someet.17380@lpu.co.in[2], anita.ri@uttaranchaluniversity.ac.in[3]
**Corresponding Author: Manjot Kaur*
[*]*Lovely Professional University, School of Electrical and Electronics Engineering, Phagwara, India*

1.1 Importance of tulsi

The plant commonly referred to as holy basil (*Ocimum sanctum*), or tulsi, is utilised extensively in traditional medicine and is revered in Hinduism. The tulsi plant's leaves are prized for their cultural importance and numerous health advantages. For millennia, tulsi leaves have been employed in Ayurvedic treatment. They are thought to possess antibacterial, anti-inflammatory, and antioxidant qualities, among other health benefits. Tulsi is frequently used to treat digestive problems, respiratory ailments, and diseases linked to stress. As an adaptogen, tulsi is thought to support equilibrium and assist the body in adjusting to stress. It is frequently used as an all-natural treatment for exhaustion, tension, and anxiety. Compounds with antioxidant qualities found in tulsi leaves can help shield the body from damage brought on by free radicals and oxidative stress. Because of its anti-inflammatory qualities, tulsi may help lessen inflammation throughout the body. It is used in traditional medicine to treat ailments like arthritis in part because of its anti-inflammatory properties. Tulsi is frequently used to treat respiratory ailments like bronchitis, colds, and coughs. It is thought to contain expectorant qualities that aid in the respiratory tract's mucus discharge. The plant may aid in bolstering the immune system because it is believed to have immune-modulating capabilities. Chewing tulsi leaves is thought to improve gum health and alleviate bad breath, among other oral health advantages [6].

The potential of tulsi in contemporary medicine, including its ability to prevent cancer, improve cardiovascular health, and help with diabetic management, is still being investigated [7]. All things considered, tulsi is not just a plant of religious veneration but also a multipurpose cultural, medical, and culinary tool that adds to its value in a variety of spheres of existence as depicted in figure 1.

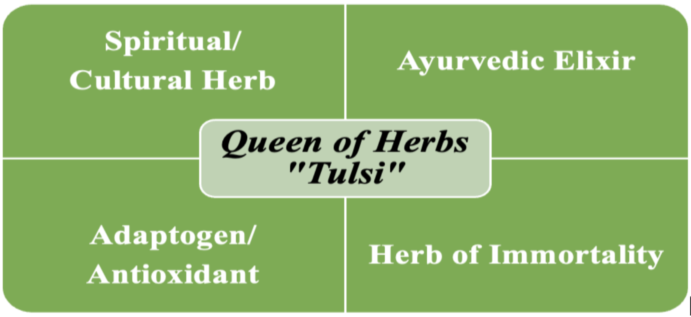

Fig 1. Applications of tulsi plant

1.2 Infections in tulsi

Ocimum sanctum, is prone to a number of pests and illnesses. Numerous illnesses that are brought on by pathogens like nematodes, bacteria, viruses, and fungi can harm plant leaves [8]. Leaf problems can also be caused by nutritional deficits and environmental causes. The following is a classification of possible problems that could impact the tulsi leaves as shown in figure 2:

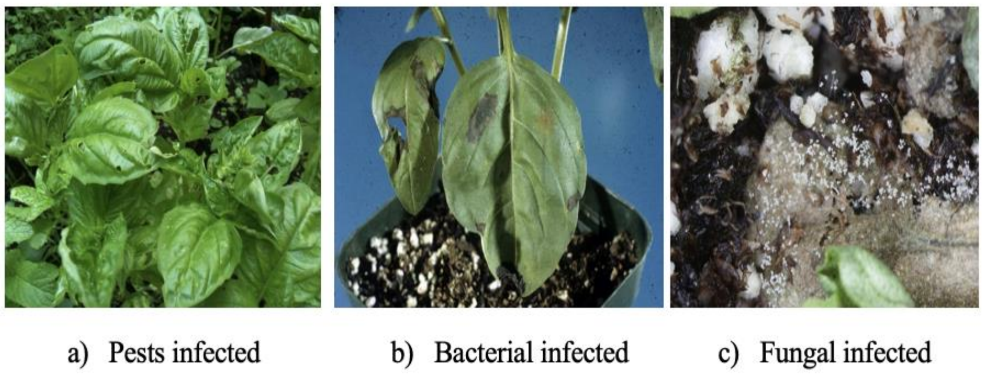

Fig 2. Infections in tulsi leaves

1. **Insects/Pests:**

a) Aphids: Tiny insects feed on sap and form clusters on the undersides of Tulsi leaves. Little white insects known as whiteflies may feed on the sap of tulsi plants.

b) Spider mites: Tiny pests that can produce stippling on leaves and fine webbing on plants.

c) The caterpillars: Tulsi leaves may become damaged by the chewing of several caterpillar species. Tulsi leaf is a potential food source for Japanese beetles and other beetle species.

2. **Bacterial Diseases:**

a) Bacterial Leaf Spot: On tulsi leaves, this condition is characterised by black lesions that appear wet.

b) Bacterial Wilt: A bacterial illness that can lead to a plant's drooping and eventual death.

3. **Fungal Diseases:**

a) Powdery mildew: A fungus known as powdery mildew causes a white, powdery material to develop on leaves.

b) Downy mildew: Another fungus that causes yellowish patches on the undersides of leaves is called downy mildew.

4. **Stress in the Environment**:

a) Stress in Water: The plant may become stressed by being submerged or overwatered, which increases its susceptibility to illness.

b) Deficit in Nutrients: Inadequate supply of vital nutrients can lead to withering leaves, stunted development, and heightened susceptibility to pests and illnesses.

c) Stress Due to Temperature: Abrupt changes in temperature, particularly low ones, can cause stress to plants and increase their susceptibility to disease.

1.3 Stacking classifier prediction model

In machine learning, a stacking classifier is used to combine many base classifiers to enhance overall predictive performance. Stacking combines the advantages of several basic classifiers to increase prediction accuracy and robustness. It makes use of the variety of several models, which may lessen overfitting and enhance generalisation to fresh, untested data. When working with intricate relationships in the data that could be difficult for a single model to adequately reflect, stacking is very helpful. The stacking classifier is able to learn from and adjust to different facets of the data because of the integration of many models. In machine learning, a stacking classifier is used to combine many base classifiers to enhance overall predictive performance. Stacking adds variation to the ensemble by utilising models with varying hyperparameters or distinct basic classifier types. Greater overall performance might result from this diversity, particularly when distinct models catch distinct patterns in the data or perform well in different domains. By combining predictions from several models, stacking can improve robustness to noisy data or outliers. Stacking is a versatile method that may be applied to various datasets and jobs. It can be customised to the unique features of the current challenge and enables experimenting with different combinations of base classifiers [9].

Stacking is an ensemble learning technique that leverages the strength of merging many models to outperform individual models. In many machine learning tasks, ensemble methods—such as stacking—perform better than individual models. Combining a base classifier with several resilient classifiers can enhance overall performance if the base classifier in question is noisy. When working with heterogeneous datasets, stacking comes in handy since different subsets of the data could be better modelled by various techniques. Every base classifier has the ability to focus on identifying patterns in the pertinent subset of data that it handles [10]. By giving base classifiers alternative weights or applying specialised methods for handling imbalanced classes in each basis model, stacking can be useful in resolving class imbalance concerns. Stacking allows for an examination of the contributions made by various base classifiers to the final ensemble prediction, which may improve the interpretability of the model.

There are five sections in this paper. The first section provided an overview of tulsi leaf diseases and machine learning models. The several phases in the prediction model for both healthy and infected tulsi leaves are explained and discussed in Section 2. The study of various image processing methods, their applications, and the choice of suitable stacking classifier algorithms for infection classification are the main topics of this section. A comparison of several machine learning classifier models for infection detection is given in Section 3. The performance metrics that can be used to analyse classifier model performance are covered in Section 4. The paper's recommendations and conclusions are presented in part five, along with some suggestions.

2. Proposed Methodology

Using machine learning techniques to analyse diverse leaf attributes and find patterns linked to specific diseases is a necessary step in developing a model for predicting diseases in leaves [11]. The steps taken to create a leaf disease prediction model are as follows:

1. Data collection: A dataset is created with labelled examples of leaves that indicate whether they are disease-free or impacted by a specific condition. A range of leaf graphics is added to symbolise various illnesses and health stages.
2. Pre-processing the data: Noise, outliers, and missing values from the dataset are removed. The leaf dataset is prepared by applying various image augmentation techniques such as scaling and normalisation.
3. Feature extraction is accomplished by either selecting pertinent characteristics from the leaf images or by applying deep learning models that have already been trained (such as MLP). The details like colour, texture, shape, and any other pertinent attributes are taken into consideration.

4. Model Selection: A machine learning model is selected that works well for tasks involving classification. Typical options consist of: Image data using MPL, SVM, RF, DT.
5. Model Training: The performance of the model is evaluated by dividing the dataset into training and testing sets. The selected model is trained on the training set by utilising the features that were retrieved.
6. Hyperparameter tuning: To enhance the model's functionality, hyperparameters are adjusted accordingly. The methods such as randomised or grid search are used to determine the optimal set of hyperparameters.
7. Model Evaluation: To gauge the performance of the trained model, evaluation is done using the testing set. Measures like accuracy, precision, recall, and F1 score are employed to assess the efficacy of the model.
8. Fine-tuning and Iteration: Iterating the model by modifying parameters or attempting other algorithms is taken into consideration. The model is adjusted in light of the evaluation's findings.
9. Validation and Cross-Validation: To make sure the model is resilient, cross-validation approaches are used. The model is tested using fresh, untested data.

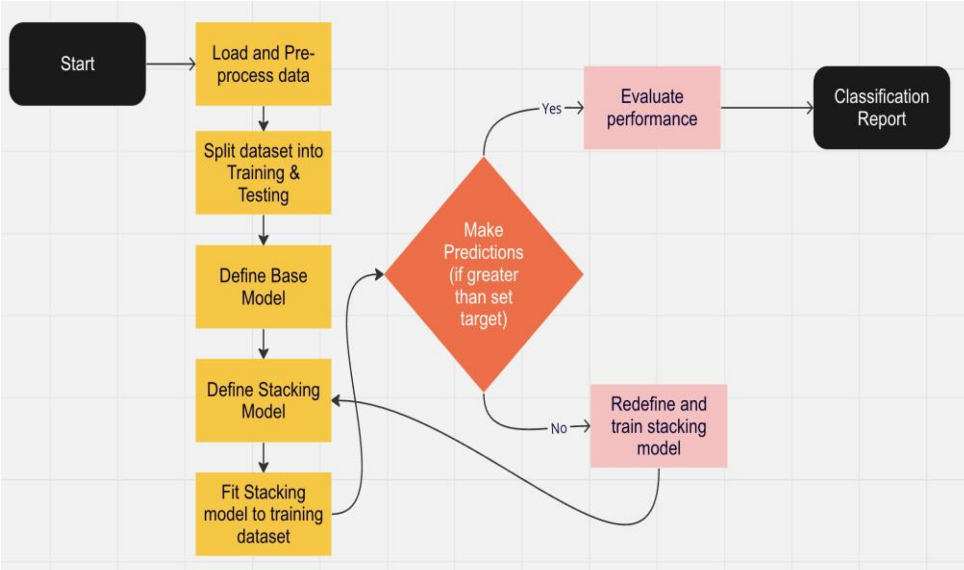

Fig 3. Flowchart for stacking classifier model

The quality of the dataset, the selected features, and the proper model selection and tuning all affect the model's performance. For the model to continue to be useful in forecasting leaf diseases, regular updates and maintenance are essential.

A stacking classifier model's parameters consist of both meta-classifier-related and base classifier-specific parameters (learners). The following is a general summary of the parameters that a stacking classifier model contain.

2.1 Parameters of base classifiers

1. Parameters specific to algorithms: These parameters are unique to each base classifier, such as the kernel type in an SVM, the number of trees in a random forest, or the learning rate for gradient boosting.
2. Hyper-parameters: The parameters (such as the depth of a decision tree, the number of layers and neurons in a neural network, etc.) that govern the general behaviour of the base classifiers are the changing hyperparameters.
3. Parameters for feature engineering: The parameters pertaining to engineering or feature extraction methods are used on incoming data before feeding it into the base classifiers.
4. Pre-processing configurations: These are the parameters pertaining to pre-processing operations on data, including normalisation, scaling, or augmentation for image data.

2.2 Model parameters for stacking method

1. Choose the stacking technique.

a. Every base classifier in homogeneous stacking is the same kind. Using various base classifier classes is known as heterogeneous stacking.
b. Multi-Layer Stacking: Stacking involves several layers and is done iteratively.
c. Architecture of Stacking: It indicates the method used to aggregate the base classifiers' predictions. Typical techniques include using the predictions as features for the meta-classifier, weighted averaging, and simple averaging.

2. Cross-Validation Configurations: These are the parameters pertaining to the cross-validation technique used in assessment and training.

3. Diversity in the Ensemble: The base classifiers' variety is varied to improve the stacking model's overall performance.

2.3 Features of classifier model

1. Support Vector Machine: SVM is an effective technique for classifying data points according to their attributes into distinct groups. SVM relies on the idea of locating the hyperplane in a high-dimensional space that best divides the classes. A hyperplane is a line that divides two classes in two dimensions. It turns into a plane in higher dimensions. Finding the ideal hyperplane that maximally divides the classes is the goal of SVM. The data points that are closest to the decision boundary (the hyperplane) are known as support vectors [12]. These points are essential for figuring out the hyperplane's orientation and location. The distance between the closest data point from each class and the hyperplane is known as the margin. Since this margin shows how reliable the classifier is, SVM aims to maximise it.
2. Random Forest: It is a collection of decision trees, each of which is autonomously built and has a prediction function. The average (for regression) or voting (for classification) over the predictions of individual trees yields the final prediction of the random forest. An ensemble of decision trees makes up a random forest. A random subset of the data is used to train each tree, and an equally random collection of features is chosen for each tree to add randomization. Bagging, also known as bootstrap aggregating, is the technique by which every Random Forest tree is trained using a bootstrap sample of the initial dataset. By selecting random samples from the original dataset and replacing them, a bootstrap sample is produced [13]. This brings variation among the trees. Only a random subset of characteristics is taken into consideration for splitting at each decision tree node. This guarantees that the forest's trees have a variety of characteristics. The Random Forest uses voting to combine the predictions made by each individual tree in classification tasks. The final predicted class is the one that gets the majority of votes. The final prediction for regression problems is calculated by averaging the predictions of each individual tree. Because each tree is trained on a bootstrap sample, some data points are absent from each tree's training set. Without a separate validation set, these out-of-bag data can be used to estimate the model's performance.
3. Decision Tree: This kind of model bases its decisions on the characteristics of the incoming data. Recursively dividing the data into subsets creates the decision tree, where decisions are made at each node according to the value of a specific feature. Nodes, branches, and leaves make up the hierarchical tree-like structure of a decision tree. Every branch indicates the result of a test conducted on a certain feature, and every node represents a choice or test on that feature. Leaves stand for the chosen course of action or anticipated outcome [14]. The tree's decision nodes include conditions that are assessed in accordance with the input feature values. The route through the tree is determined by these factors. In a classification, a leaf node may be a class label; in a regression, it may be a numerical value. Leaf nodes indicate the final result.
4. Multi-layer Perceptron: MLP classifier is an implementation of a feedforward neural network with one or more hidden layers that is a part of the python scikit-learn toolkit. A feedforward neural network is a kind of artificial neural network in which data travels from the input layer through the hidden layers and out to the output layer only in one direction. The word "multi-layer" in MLP describes the possibility of multiple node layers within the network between the input and output layers. It is possible to modify the concealed layers' architecture. To give the network non-linearity and enable it to learn intricate patterns, non-linear activation functions are employed. The rectified linear unit (relu) and hyperbolic tangent (tanh) are examples of common activation functions. In order to reduce the error between the expected and actual outputs, the model is trained using optimisation and backpropagation techniques. The MLP classifier in scikit-learn makes it simple to build and train a multi-layer perceptron for classification tasks. The number of hidden layers, the number of nodes in each hidden layer, activation functions, and other parameters are among the many customisation options it allows [15-16].

3. Analytical Tools and Predictive Algorithms

Choosing the right machine learning libraries, installing them, and using them to construct and train the classifier are the steps involved in setting up a classifier model. Two well-liked python machine learning libraries are TensorFlow and scikit-learn. While TensorFlow is frequently used for deep learning, scikit-learn is a general-purpose open-source machine learning package. These packages can be installed by pipfile in visual studio code with python version 3.11 or 3.12. While TensorFlow is very effective for deep learning applications, scikit-learn is appropriate for a variety of machine learning activities. Depending on the complexity of the classifier model and the type of data the library is selected that best meet the needs.

Metrics including accuracy, precision, recall, and F1 score are frequently employed to assess a classification model's performance. It's crucial to remember that these metrics are related, and the decision of which to give priority to will rely on your model's particular objectives as well as the effects of false positives and false negatives on your application. Scikit-learn in Python can be used to compute these metrics by the anticipated labels (y_pred) and ground truth labels (y_true) are calculated for the proposed stacking classifier model. A table known as a confusion matrix is frequently used to explain how well a classification model performs when applied to a set of test data for which the true values are known. It offers a thorough analysis of the model's performance and divides the predictions into groups like false positives, false negatives, real positives, and true negatives [17].

Table 1. Confusion matrix parameters

Prediction groups	Description
True Positive	The positive class instances accurately predicted by the model
True Negative	The negative class instances accurately predicted by the model
False Positive	Type I mistake occurred when the model anticipated instances of the positive class incorrectly.
False Negative	A Type II error occurred when the model predicted instances of the negative class incorrectly.

We have pinpointed particular kinds of mistakes the model is making by looking at the confusion matrix (e.g., more false positives than false negatives or vice versa). This data is useful for improving the model. The decision threshold of the model is decided with the aid of the confusion matrix. For instance, the threshold can be changed to give priority to one sort of error over another in situations where false positives or false negatives have distinct outcomes. In summary, the confusion matrix is a crucial tool for understanding the performance of a classification model. It enabled to go beyond basic accuracy and offers perceptions into the advantages and disadvantages of the model, directing future model enhancement and decision-making [18].

4. Classifier Model Comparisons

Following are different stacking classifier models being employed with training and testing datasets with 80% and 20% respectively on a dataset of 20,000 tulsi leaves images classified into four sub categories - bacterial, fungal, pests and healthy leaves as shown in figure 5.

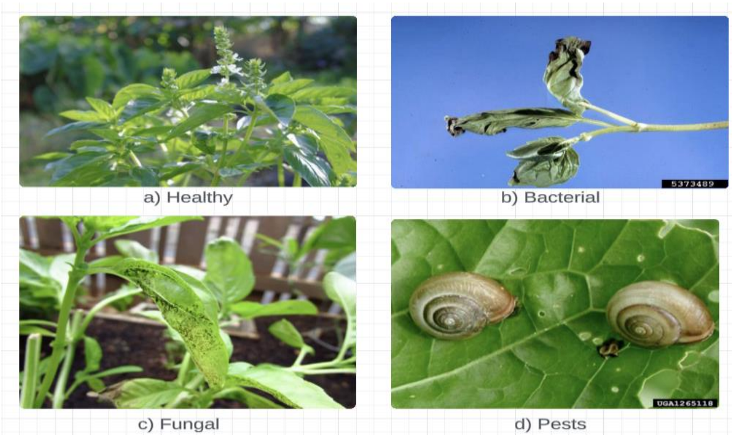

Fig 5. Classification types in tulsi leaves

These classifiers are trained on the input data independently. The base estimators that are supplied are used to instantiate the stacking classifier. To arrive at a final prediction, it integrates the forecasts from the basic estimators. An additional classifier is selected as the final estimator parameter which makes the ultimate choice as shown in table 2 and results in figure 6-10 based on the input of the base classifiers' predictions for five different models.

Table 2 Classifier Model Parameters

Proposed Model	Stacking Estimators	Final Estimator	Accuracy
Model A	SVM, RF	RF	97.65%
Model B	SVM,RF	SVM	98.25%
Model C	SVM, RF, DT	DT	95.38%
Model D	SVM, RF, DT	MLP	97.12%
Model E	SVM, RF, DT, MLP	MLP	97.35%

```
Accuracy: 0.97
Classification Report:
              precision    recall  f1-score   support

   bacterial       1.00      1.00      1.00      1024
      fungal       0.96      0.97      0.96       977
     healthy       0.97      0.97      0.97       995
       pests       0.95      0.95      0.95      1004

    accuracy                           0.97      4000
   macro avg       0.97      0.97      0.97      4000
weighted avg       0.97      0.97      0.97      4000
```

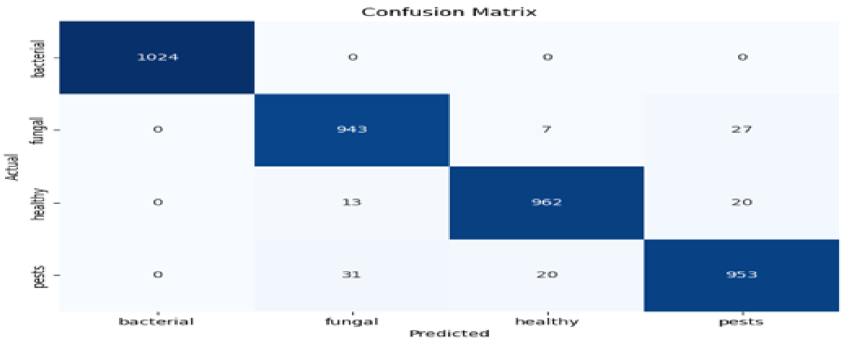

Fig 6. Classification report and confusion matrix for model A

```
Accuracy: 0.98
Classification Report:
              precision    recall  f1-score   support

   bacterial       1.00      1.00      1.00      1024
      fungal       0.97      0.98      0.98       977
     healthy       0.97      0.98      0.97       995
       pests       0.96      0.95      0.96      1004

    accuracy                           0.98      4000
   macro avg       0.98      0.98      0.98      4000
weighted avg       0.98      0.98      0.98      4000
```

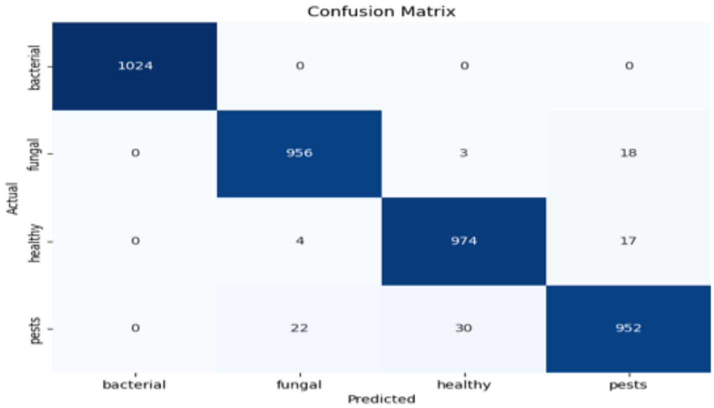

Fig 7. Classification report and confusion matrix for model B

```
Accuracy: 0.95
Classification Report:
              precision    recall  f1-score   support

   bacterial       1.00      1.00      1.00      1024
      fungal       0.94      0.94      0.94       977
     healthy       0.94      0.95      0.95       995
       pests       0.93      0.93      0.93      1004

    accuracy                           0.95      4000
   macro avg       0.95      0.95      0.95      4000
weighted avg       0.95      0.95      0.95      4000
```

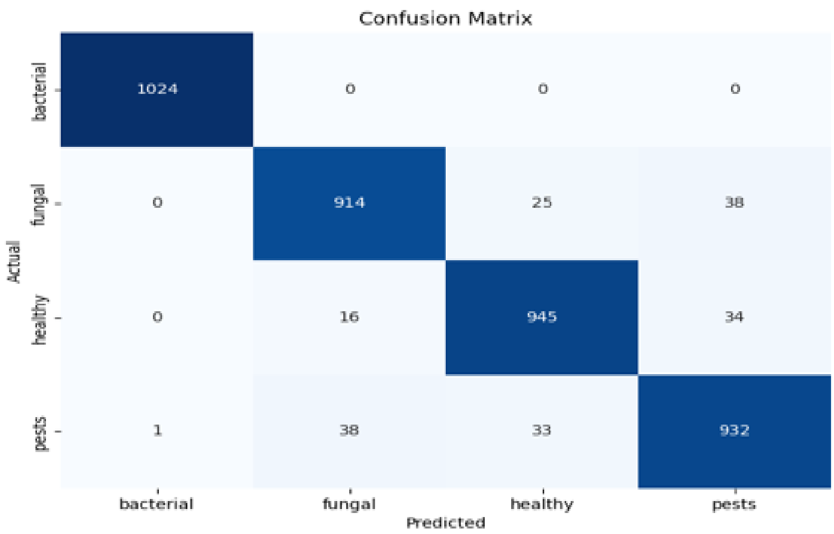

Fig 8. Classification report and confusion matrix for model C

```
Accuracy: 0.97
Classification Report:
              precision    recall  f1-score   support

   bacterial       1.00      1.00      1.00      1024
      fungal       0.96      0.97      0.96       977
     healthy       0.96      0.97      0.97       995
       pests       0.97      0.95      0.96      1004

    accuracy                           0.97      4000
   macro avg       0.97      0.97      0.97      4000
weighted avg       0.97      0.97      0.97      4000
```

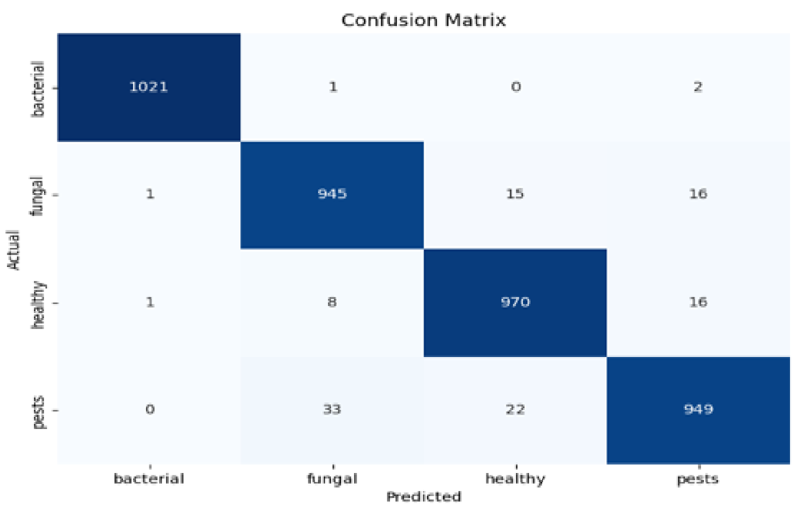

Fig 9. Classification report and confusion matrix for model D

Accuracy: 0.97

Classification Report:

	precision	recall	f1-score	support
bacterial	1.00	1.00	1.00	1024
fungal	0.96	0.97	0.97	977
healthy	0.97	0.97	0.97	995
pests	0.96	0.95	0.96	1004
accuracy			0.97	4000
macro avg	0.97	0.97	0.97	4000
weighted avg	0.97	0.97	0.97	4000

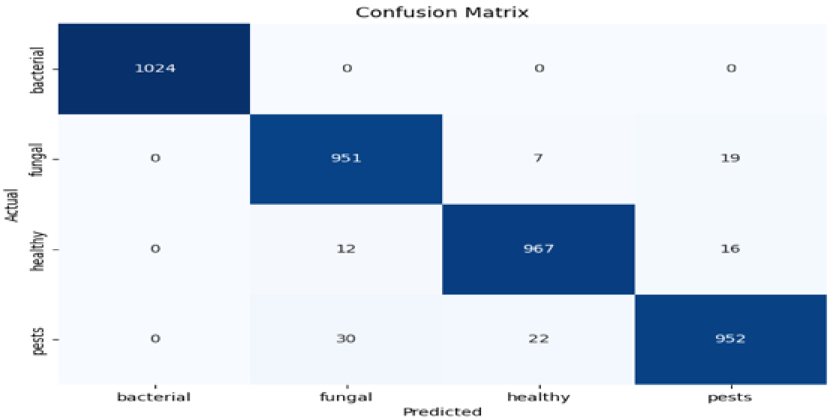

Fig 10. Classification report and confusion matrix for model E

The results demonstrate that the type of data can have an impact on how well SVM and RF function. SVM, for instance, works well in high-dimensional domains and is appropriate in situations where the decision boundary isn't always linear. However, RF is well-known for managing intricate links and interactions among the data. The combination of SVM and RF is able to capture the subtleties in the data as it has both linear and non-linear elements. It was critical to test out various setups in order to determine which one best suits the particular issue, but model B combination has outperformed with average accuracy of 98.25% as demonstrated in figure 11.

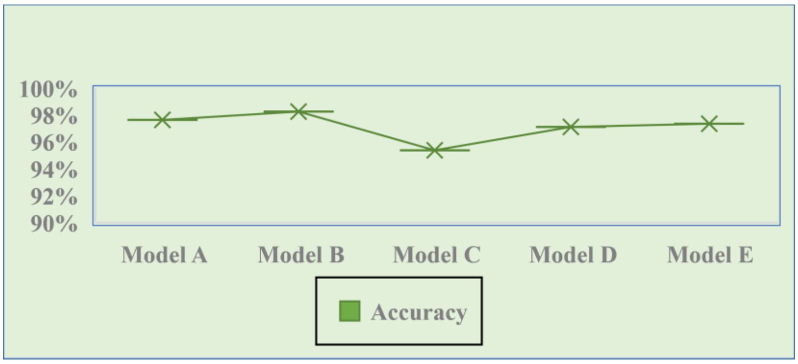

Fig 11. Accuracy comparison of model A, B, C, D, E

A more reliable representation of the underlying data structure can be obtained from the ensemble when paired with RF, which is also efficient in capturing non-linear patterns. By combining the predictions of several models, stacking frequently enhances performance overall by utilising the advantages of each model. As the stacking classifier, SVM gains the ability to integrate the predictions from RF and SVM in a way that maximises performance across the board on the validation set.

The below table compares different models accuracy with existing literature and architecture used for disease prediction in tulsi leaf research:

Table 3. Classifier model comparison with existing literature on tulsi

Model	Stacking Estimators	Dataset	Accuracy
Model B	Stacking Estimators: SVM,RF Final Estimator: SVM	20,000 (4 classes)	98.25%
[9]	Transfer Learning, SVM	7000 (7 classes)	SVM-97% Transfer learning-98%
[17]	CNN	266 (2 classes)	75%
[19]	CNN Inception V3 model	15220 (3 classes)	77.55%
[20]	k-means clustering	1628 (3 classes)	0.59

Hence the performance of a model, or even an ensemble, can differ in various problem domains and datasets. Thoroughly validating the model's performance on separate test sets and taking into account real-world application scenarios is always a smart idea. The model performed well in all assessments, this indicates that the ensemble configuration and models chosen are indeed appropriate for the particular problem we are trying to solve.

5. Conclusion and Future Scope

In this work, we investigated the use of a stacking classifier model to categorize tulsi leaves into four groups and forecast type of infection in them. A thorough comparison of several stacking classifier setups, using decision trees, SVM, random forests, and MLP as basis models, was the focus of the research work. The comparison research showed that the stacking classifier performed better than individual classifiers in terms of accuracy, precision, recall, and F1 score by utilising the combined intelligence of several base models. In order to help farmers and researchers conduct targeted therapies based on specific disease categories, the model showed a high degree of accuracy in differentiating between the four classes of tulsi leaf infections [21-22]. In future this type of stacking classifier model can be employed in many scenarios as described below:

1. Fine-Tuning Model Architectures: To improve the predictive performance of the model, additional base model and final estimator (MLP) optimisation may be investigated. This include experimenting with sophisticated neural network designs, modifying network architectures, and fine-tuning hyper-parameters.
2. Data Expansion and Augmentation: Adding more photos from other sources and diversifying the dataset by using data augmentation techniques can help to make the model more robust and generalizable.
3. Real-Time Monitoring System: By incorporating the stacking classifier model into a real-time monitoring system for Tulsi crops, farmers can receive immediate plant health status information and take prompt action in the event of any infections.
4. Lucidity and discernment: Including strategies for interpretability in a model can improve confidence and comprehension of the model's decision-making process, making it easier to understand for non-specialists and assisting in the discovery of important characteristics linked to various infection classes.
5. Integration with Precision Agriculture Technologies: A comprehensive approach to crop monitoring and management can be achieved by integrating the stacking classifier model with other precision agriculture technologies, such as satellite images and Internet of Things-based sensor networks.
6. Cooperation with Agricultural Experts: The practical relevance and usability of the model for on-field applications can be guaranteed by involving agricultural experts and practitioners in the validation and deployment of the model in real-world contexts.

To sum up, the stacking classifier model exhibits potential for predicting tulsi leaf infections. Its practical utility in precision agriculture and crop health management can be enhanced by additional developments in model optimisation, dataset augmentation, and real-time implementation.

References

[1] Parveen, A., Perveen, S., Ahmad, M., Naz, F., & Riaz, M. (2023). Tulsi. In Essentials of Medicinal and Aromatic Crops (pp. 983-1008). Cham: Springer International Publishing.

[2] Mishra: T., Rai, A. (2021). Antimicrobial potential of Ocimum santum and Adhatoda vasica: The Medicinal Herbs. In-ternational Journal of Pharmacy and Biological Sciences 1191, 17-28. doi:10.21276/ijpbs.2021.11.1.3.

[3] Suryawanshi, S. K., & Chouhan, U. (2018). Computational approaches for the prediction of antimicrobial potential peptides from Ocimum tenuiflorum. Asian J. Pharm. Clin. Res, 11, 398-401.

[4] Khan, A., & Srivastava, A. (2023). PlantDoc-Plant Disease Detection using AI. Journal of Informatics Electrical and Electronics Engineering (JIEEE), 4(1), 1-10.

[5] Mahomodally, A. F. H., Suddul, G., & Armoogum, S. (2023). Machine learning techniques for plant disease detection: an evaluation with a customized dataset. Int J Inf & Commun Technol, 12(2), 127-139.

[6] Agrawal, R., Kumar, A., & Singh, S. (2023). Automatic Detection and Classification of Healthy and Unhealthy Plant Leaves. In Machine Vision and Augmented Intelligence: Select Proceedings of MAI 2022 (pp. 531-537). Singapore: Springer Nature Singapore.

[7] Lavanya, V., Ganapathy, D., Visalakshi, R.M. (2019). Antioxidant and free radical scavenging activity of Ocimum basilicum - An in vitro study. Drug invention Today 12(5), 1004-1007.

[8] Rhoades, H. (April 2021). Diseases And Problems With Growing Basil. https://www.gardeningknowhow.com/edible/herbs/basil/basil-diseases.htm.

[9] Qi, H., Liang, Y., Ding, Q., & Zou, J. (2021). Automatic identification of peanut-leaf diseases based on stack ensemble. Applied Sciences, 11(4), 1950.

[10] Gautam, V., Ranjan, R. K., Dahiya, P., & Kumar, A. (2023). ESDNN: A novel ensembled stack deep neural network for mango leaf disease classification and detection. Multimedia Tools and Applications, 1-27.

[11] Yaswanth, T. N. S., Lakshmi, K. V., Vamsi, K. G., Supraja, D., & Vani, K. S. (2023, July). Classification of Medicinal Leaves using SVM. In 2023 World Conference on Communication & Computing (WCONF) (pp. 1-7). IEEE.

[12] Ganatra, N., Patel, A. (2020). A Multiclass Plant Leaf Disease Detection using Image Processing and Machine Learning Techniques. International Journal on Emerging Technologies 11(2), 1082-1086.

[13] Begue, A., Kowlessur, V., Singh, U., Mahomoodally, F., Sameerchand, P. (2017). Automatic Recognition of Medicinal Plants using Machine Learning Techniques. International Journal of Advanced Computer Science and Applications 8(4), 166-175. doi:10.14569/IJACSA. 2017.080424.

[14] Haagsma, M., Page, G. F., Johnson, J. S., Still, C., Waring, K. M., Sniezko, R. A., & Selker, J. S. (2021). Model selection and timing of acquisition date impacts classification accuracy: A case study using hyperspectral imaging to detect white pine blister rust over time. Computers and Electronics in Agriculture, 191, 106555.

[15] Vijaya Durga Reddy, P., & Amudha, V. (2023, May). Comparison of tulasi leaf diseases identification using different convolutional neural network layers. In AIP Conference Proceedings (Vol. 2602, No. 1). AIP Publishing.

[16] Azadnia, R., Al-Amidi, M. M., Mohammadi, H., Cifci, M. A., Daryab, A., & Cavallo, E. (2022). An AI Based Approach for Medicinal Plant Identification Using Deep CNN Based on Global Average Pooling. Agronomy, 12(11), 2723.

[17] Patil, S. S., Patil, S. H., Bhall, A., Rajvaidya, A., Sehrawat, H., Pawar, A. M., & Agarwal, D. (2022, December). Tulsi Leaf Disease Detection using CNN. In 2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI) (pp. 1-4). IEEE.

[18] Sarkar, C., Gupta, D., Gupta, U., & Hazarika, B. B. (2023). Leaf disease detection using machine learning and deep learning: Review and challenges. Applied Soft Computing, 110534.

[19] Sathiya, V., Josephine, M. S., & Jeyabalaraja, V. (2023). Plant Disease Classification of Basil and Mint Leaves using Convolutional Neural Networks. International Journal of Intelligent Systems and Applications in Engineering, 11(2), 153-163.

[20] Sathiya, V., Josephine, D., & Jeyabalaraja, D. (2022). Identification And Classification Of Diseases In Basil And Mint Plants Using Psorbfnn. J. Theor. Appl. Inf. Technol. 100, 21.

[21] Bharathan, K., & Deepasree, V. P. (2018, December). Tulsi Leaves Classification System. In 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), 1-5. IEEE.

[22] Tangtisanon, P., & Kornrapat, S. (2020, February). Holy basil curl leaf disease classification using edge detection and machine learning. In Proceedings of the 2020 12th International Conference on Computer and Automation Engineering (pp. 85-89).