COMPUTOLOGY: Journal of Applied Computer Science and Intelligent Technologies
Volume 3, Issue 2, Jul-Dec 2023, pp. 153-170
DOI: 10.17492/computology.v3i2.2308
https://www.journalpressindia.com/cjacsit

A Broad Review of Various Machine Learning Models for Disease Detection in the Domain of Agriculture

Madhavi[1]

ABSTRACT

The Agriculture domain is a vast domain with various grain and food categories which helps to boost the economy. Among all these categories leaf disease is the one common issue that impacts the overall growth of crops and can be dealt with with the use of models that have been proposed in ML and DL. In this paper, we have given an introduction to such models which have performed extraordinarily for disease classification and detection in the field of agriculture. Various crops have been listed according to their accuracy and the model has been proposed by various researchers. The algorithms' performance is typically assessed using a variety of metrics, such as F1 score, accuracy, and precision. Researchers working in this field who are searching for different effective ML and DL-based classifiers for leaf disease detection would find this review to be useful.

Keywords: *ML; DL; Agriculture; Plant Disease.*

1.0 Introduction

Any nation's ability to identify leaf diseases is critical to its ability to flourish econom- ically. While viruses, fungi, bacteria, and other infectious organisms can infect many areas of a plant, the focus of this study was primarily on the identification of plant diseases that affect the leaves. The biggest threat to the integrity of the food supply is plant diseases, which can be difficult to detect in the early stages to minimize the possibility of consequential economic harm. Because they feed us food and protect us from radiation, plants are essential to our existence. It is hard to believe that life could have ever existed on Earth without plants, as they provide food for all terrestrial ani- mals and protect the ozone layer from damaging UV radiation.

[1]*Computer Science Engineering, Bennett University, Greater Noida, Uttar Pradesh, India* [2]

Early disease detection would help the farmers to increase their overall profit and help them to reduce the losses by using pesticides. For identifying the disease in plants or crops at an early stage NLP, ML, and DL algorithms are very helpful. Expert disease detection is necessary to address the issue of declining crop yields and quality degradation in pearl millet due to diseases like rust and blast, which are still out of reach for many farmers. Plant dis- ease detection by traditional approaches is expensive, labor-intensive, and unfeasible. This emphasizes how urgently we need automated methods for classifying and detect- ing diseases. It has been suggested that DL and IoT-based methods be integrated, but there is a great deal of potential to create systems that combine these technologies at a reasonable cost to improve agricultural disease management. These systems could be used for effective data collection, feature visualization, and accurate disease detection. The advancement of computer technology has made it possible to successfully apply traditional ML techniques in numerous fields. Beltran-Perez successfully used the dis- crete cosine transform to a multiscale generalized radial basis function network and used this technique to create a computer-aided diagnosis system for the identification of breast cancer[1]. Pham assessed four ML techniques for mapping and predicting fire vulnerability. The models were evaluated using seven additional performance criteria in addition to the area under the receiver operating characteristic curve.

2.0 Literature Review

This section describes various research studies proposed to detect plant leaf disease using ML and DL methods. The researchers have already proven the improved perfor- mance of models for multiple crops, including tomato, potato, cassava, olive, grape, betel, melon, and fava beans using classification evaluation metrics. In [2] work on Unmanned aerial vehicles (UAV) multispectral data and ML approaches to estimate the oat biomass has been presented. They have used airborne remote sensing with ML which has the ability for above-ground biomass estimation. Authors in [3] has proposed hyperspectral imaging (HSI) and DCNN to differentiate the varieties of oats seeds. They have merged DCNN with LR and RBF SVM and achieved the highest accuracy of 99.19% To identify the yield of black gram crop the authors in [4] has worked on GoogLeNet, DarkNet-19, SqueezeNet, and AlexNet to identify the disease caused by yellow mosaic disease and achieved the highest accuracy of 96.09% with the DarkNet-19 model. To identify pests and diseases in mung bean the authors in [5] proposed a novel technique based on DL for this they have used a transfer learning approach to

detect four types of pests. By screening 96 genotypes of urdbean against MYMIV, this study [6] was able to identify DPU88-31 as resistant and IC436566 as vulnerable to the virus. It laid the groundwork for better urdbean breeding by revealing increased expression of stress- related genes and enzyme activity in resistant genotypes. The authors in [7] uncovered a monogenic recessive control of YMV-tolerance in Vignas, identified through forced inoculation and segregation analysis in progenies, revealing a novel DNA marker

Table 1: Summary of Recent Work on DL and ML models

Year	Crop	Model	Accuracy
2022	Wheat	Fine-tuned RFC[9]	98.80%
2021	Wheat	CNN[10]	89.90%
2021	Rice	MobileNetV2[11]	94.65%
2020	Rice	SVM[12]	98.38%
2020	Corn	Improved CNN model[13]	98.78%
2023	Maize	CNN-RB model[14]	98.12%
2023	Mungo	DarkNet19[4]	96.69%
2021	Peanut	ResNet50[15]	97.59%
2022	Soybean	DenseNet201[16]	96.80%
2022	Pepper	Optimized CNN[17]	99.99%
2022	Bansal Stem	MPNet[18]	97.11%

'VMYR1' linked to YMV resistance. This marker, showing homology with the NB- ARC domain, marks a significant advancement in resistance breeding and is the first reported YMV-resistance-linked DNA marker in crop species, offering the potential for commercial breeding applications. The author[8] presents an image-based approach to black gram crop disease detection with particular emphasis on powdery mildew, anthracnose, leaf crinkle, and yellow mosaic diseases. It does a comparative analysis between ML algorithms (DT, KNN, RF) and DL techniques (ANN, CNN) using the BPLD dataset. The CNN performed better than the others, with an accuracy rate of 89%. This demonstrated the network's value in detecting black gram crop diseases and offering data that might be utilized to enhance agricultural disease surveillance. Summary of recent work is in Table 1.

2.1 Decision trees

The decision tree is a popular and flexible ML method for applications involving regression and classification. It works by breaking up a dataset into smaller, more manageable chunks and then gradually building a decision tree to go along with it.[19].

In the end, we get a tree with decision[20] and leaf nodes, where an internal node represents a ”test” on an attribute, a branch the test’s outcome, and a leaf node the class label—a decision made after all attributes are computed. The paths that link the leaf and the root represent the categorization rules.

Tree parameters:

- *Root Node:* Splits into two or more homogenous sets and represents the total population or sample.
- *Splitting:* The act of splitting a node into two or more smaller nodes is known as splitting.
- *Decision Node:* The sub-node transforms into a decision node following splitting.
- *Terminal Nodes:* Nodes that don’t split any farther.
- *Pruning:* Eliminating a decision node’s sub-nodes; this can be done

2.2 Random forest

A popular ensemble learning technique in ML for both regression and classification problems is called Random Forest[21]. Typically trained using the ”bagging” approach, it generates a decision tree ”forest”. Building many decision trees and merging them to get a prediction that is more reliable and accurate is the basic idea. It can handle complicated, huge datasets and simulate non-linear relationships, Random Forest[22] is a versatile ML technique that works incredibly well in agriculture. Utilizing an analysis of symptoms and environmental data, it is used to diagnose crop diseases; it also evaluates soil quality using a variety of soil characteristics; and it predicts yields taking into account soil, weather, and agricultural methods.

The optimal distribution of resources, such as irrigation and fertilizer, is achieved through precision agriculture. Random Forest also aids in livestock health monitoring, weed and insect detection, and agricultural resource management. Enhancing agricul- tural output and efficiency is made possible by its ability to regulate many variables and resilience against overfitting.

Salient features:

- *Ensemble of Decision Trees:* Random Forest creates a more accurate and reliable model by combining several decision trees. Each tree is trained on a random subset of the data, making the trees somewhat independent of each other.
- *Bagging:* Also known as bootstrap aggregating, involves training every tree in a random forest using a random sample (bootstrap sample) that is selected at random from the training set.

- *Randomness of Features:* During the tree-building process, a node is divided, and the optimal split is selected from a random subset of its features. As a result, the trees become more diverse and the forest is more resilient to data noise.
- *Prediction:* The class that the majority of trees (the mean) choose for classification problems is the Random Forest output. It is the mean forecast made by each tree in a regression job.

2.3 Support Vector Machines(SVM)

(SVM)[23] is a powerful and flexible supervised ML technology that can be used for both regression and classification applications. One of its main uses is the classification of complex small- or medium-sized datasets.

- Support vector machines' main objective is to find the hyperplane in an N-dimensional space that best classifies the data points. SVM's core concepts and characteristics include:
- A hyperplane is the designated class division line. SVM finds the hyperplane to maximize the margin between the classes.
- The margin in the closest class points is the separation between two lines. This place does not have any data points available. SVM aims to increase this margin to reduce misclassification.
- Support vectors are the closest data points to the hyperplane that have an impact on its orientation and location. These are useful factors to consider while building the SVM model.
- SVM employs a method called the kernel trick to subtly convert inputs into high-dimensional feature spaces to produce non-linear classifications. ordinary kernels

2.4 Naive Bayes

The Naive Bayes algorithm is a simple yet powerful ML technique that relies on applying the Bayes theorem under the assumption of independence between each pair of features[24]. Despite its simplicity, its efficacy has been shown in a range of applications, particularly in classification tasks.

The following are Naive Bayes' main features:

- *The Bayes Theorem:* The Bayes[25] theorem, which expresses the likelihood of an event based on past knowledge of event-related factors, is the basis of the algorithm.
- The foundation of Naive Bayes is the feature independence hypothesis, which postulates that the presence (or absence) of a given feature within a class is inde-

pendent of the presence (or absence) of any other characteristic. This assumption simplifies the calculation, but it is not necessarily true with real-world data.

Three main models are available for the Naive Bayes model:

- *Gaussian:* Assumes that features follow a normal distribution. In situations where characteristics are continuous, this is helpful.
- *Multinomial:* Usually used for document categorization, where words' frequencies or occurrences serve as features or variables.
- *Bernoulli:* Applies to boolean and binary characteristics.

2.5 K-Nearest Neighbors(KNN)

The ML technique KNN[26] has several applications in agriculture that can be highly advantageous. The K-NN algorithm is a simple, flexible, and uncomplicated technique that finds the nearest data points in the feature space and makes predictions based on these points' labels.

K-NN has the following applications in the agricultural domain:

- Using attributes taken from photos of the crops, KNN[27] can be used to categorize and identify diseases in crops. The system can determine which crop disease is afflicting a certain crop by comparing the attributes of that image with those in a database.
- Precision Agriculture: By evaluating data from numerous sensors positioned around the field, KNN can help in precision agriculture. Making educated deci- sions about irrigation, fertilizing, and harvesting can be aided by this data, which can include soil moisture levels, temperature, crop health, and other factors.

2.6 Linear regression

Logistic regression is a basic statistical and ML approach used to model and analyze the relationship between a dependent variable and one or more independent variables. Finding the best-fitting straight line through the data points is the basic goal of linear regression.

The following are linear regression's main features:

- Two variables are used in simple linear regression: A response, which is the dependent variable, and a predictor, which is the independent variable.
- By utilizing more than one independent variable, multiple linear regression builds on the principles of simple linear regression[28].

- Assumptions: The main presumptions of linear regression are that the errors have a normal distribution, are homoscedastic (constant variance), are linear and independent.
- The Least Squares Method reduces the total squared discrepancies between the model's predicted values and the observed values.

Benefits

- Ease of Interpretation and Simplicity: The model is easy to comprehend and apply.
- Basis for other Techniques: Provides the framework for numerous other regression methods.

Drawbacks

- Sensitivity to Outliers: Outliers have a big impact on this.
- Assumption of Linearity: Because it assumes a linear relationship between variables, this approach may not be suitable for modeling complex relationships.

2.7 Logistic Regression

A popular ML approach, logistic regression is especially well-suited to tasks involving binary categorization. The dependent variable in this regression model is categorical.

The salient features of Logistic Regression are as follows:

- *Binary Classification:* Based on one or more predictor variables, logistic regression[29] is primarily used to predict a binary outcome (1/0, Yes/No, True/False).
- *Sigmoid Function:* In logistic regression, the likelihood that a given input is a member of the default class is modeled using the logistic function, also referred to as the sigmoid function. The probability value produced by the sigmoid function ranges from 0 to 1.

Benefits

- *Efficiency:* It is a suitable option for big datasets because it is not computationally demanding.
- *Probabilistic Interpretation:* Rather than providing only a basic classification, the output can also be understood as a probability.
- *Interpretability:* It is simple to comprehend how each feature affects the likelihood of a particular result.

Drawbacks

- The assumption of linearity states that there is a linear relationship between the log odds of the dependent variable and the independent variables.
- Performance problems: Non-linear interactions, complicated relationships, and vast feature spaces may cause the system to perform poorly.

2.8 AdaBoost and Gradient Boosting

Both Gradient Boosting and AdaBoost (Adaptive Boosting) are well-liked ensemble learning strategies in ML that increase model accuracy by fusing several weak learners to produce a strong learner.

The two approaches are contrasted here:

AdaBoost Concept: AdaBoost[30] focuses on cases that were incorrectly clas- sified in earlier rounds and add weak learners—usually decision trees—in a stepwise manner. The correctness of each learner's input to the final model determines its weight.

- Weight Adjustment: The weights of instances that were wrongly classified are increased and the weights of instances that were correctly identified are dropped at the end of each iteration. In the next iterations, this approach causes the algorithm to concentrate more on challenging scenarios.

Gradient Boosting Concept: Gradient Boosting employs gradient descent to mini- mize loss while adding new models, but it constructs the model in a stage-wise manner similar to AdaBoost. Decision trees are usually used for weak learners.

- Loss Function Optimization: This technique can be used for a variety of issues, such as regression and classification, as it optimizes a differentiable loss function.
- Sequential Improvement: Every new model introduced is trained to address the flaws in the previously contributed models.

2.9 Principal component analysis(PCA)

PCA is a statistical method for dimensionality reduction in ML that preserves as much of the data's variance as feasible. It's very helpful when handling data that has many dimensions.

The main facets of PCA are as follows:

- *Dimensionality Reduction:* By extracting the most crucial information and eliminating the less crucial information, PCA[31] lowers the number of variables in your data. To do this, the original variables are changed into a new set known as the principal components, which are orthogonal (or uncorrelated) to one another.

- *Eigenvectors and Covariance Matrix:* PCA entails determining the data's eigenvectors as well as calculating the covariance matrix. The directions of maximum variance are the covariance matrix's eigenvectors, and the amount of variance that each primary component captures is indicated by its corresponding eigenvalue.
- *Data Transformation:* To convert the data into the new space, the original data points are projected onto the primary components. The new features become the major components of a lower-dimensional representation of the data as a result.

Benefits: Eliminates Correlated Features: Principal component analysis (PCA) eliminates feature correlation by converting the data into principal components. Enhances Algorithm Performance: By eliminating noise and unnecessary information, dimensionality reduction can result in quicker training periods and improved performance.

Drawbacks: Information Loss: During the procedure, some information that could be crucial to the analysis is lost. Interpretability: The principal components might not be interpretable because they are just linear combinations of the original features. Sensitive to Scaling: Before using PCA, feature scaling is frequently necessary since PCA is sensitive to the scaling of the variables.

2.10 Linear Discriminant Analysis(LDA)

A statistical method for both dimensionality reduction and classification in ML is called LDA. It's particularly successful when the classes are linearly separable.

The following are linear discriminant analysis's main features:

- *Goal:* Finding a linear feature combination that best separates two or more classes of objects or events is the goal of lattice distance analysis (LDA)[32], as opposed to principal component analysis (PCA), which only focuses on maximiz- ing variance. To guarantee that classes are as distinct as feasible, it maximizes the ratio of within-class variance to between-class variation.
- *Dimensionality Reduction:* LDA minimizes the number of dimensions (features) while retaining the majority of the class-discriminating data. It projects the data onto a lower-dimensional space where the classes are most recognizable.
- *Class Separation:* LDA assumes that classes are equally covariant and regularly distributed. It determines a feature combination that is linear and maximizes the distance between different classes.
- *Computation:* The covariance matrix as a whole and the means and variances of the individual classes are computed. The linear discriminants are the eigenvectors of the covariance matrix that correspond to the biggest eigenvalues.

Benefits

- *Efficiency:* When there are fewer samples than features, LDA is especially helpful.
- *Interpretability:* Compared to PCA features, the resulting dimensions—also known as linear discriminants—are a linear mixture of characteristics and may be simpler to understand.

Drawbacks

- Normality and Equal Covariance Assumptions: LDA makes the assumptions that the classes' covariance matrices are equal and that the features are normally distributed, which may not be the case in real-world data.
- Binary and Multi-Class Classification: LDA is a binary classifier by nature, but it may also be used to solve multi-class problems, albeit the performance may not always be ideal.

2.11 Hidden Markov Model(HMM)

A class of statistical models known as HMMs describes systems with probability transitions between hidden states across time. When applied to sequences of observed data that may be represented as a set of outputs produced by a stochastic process, they are especially helpful.

The following are the main features of Hidden Markov Models: Parts:

- *States:* There is a finite number of states in HMMs[33], and each state has a corresponding probability distribution. The states are "hidden" because they cannot be observed directly.
- *Observations:* One can immediately observe an observation that is produced by every state.
- *Transitions:* The possibility of changing from one state to another is defined by a set of probabilities in the model.
- *Emissions:* The likelihood of each potential observation when is defined by the probability distribution of each state.

Benefits

- *Simulating Temporal Dynamics:* HMMs work effectively when simulating time-series data in which the system's state changes over time.
- *Managing Hidden States:* They can simulate intricate systems whose true state is not readily apparent.
- *Flexibility:* A wide range of applications in various domains can be tailored to HMMs.

Drawbacks

- *Assumption of Independence:* The Markov property assumes that the current state is the sole thing that influences future states, independent of the events that led up to it.
- *Scalability:* The calculating of probability can become laborious for large state spaces.
- *Model Complexity:* Choosing the right amount of states and the model's architecture can be difficult.

3.0 DL Models/Algorithm

3.1 Convolutional Neural Networks (CNNs)

DL models known as CNNs are mainly employed in computer vision, though they can also be used in other domains. When analyzing data with a grid-like architecture, like photographs, CNNs do especially well.

CNNs include the following important features:

- *Architecture:* CNNs[34] is made up of several layers that combine to alter the input image and extract features (such as detection and classification) that are pertinent to the task at hand. The primary kinds of layers found in a CNN are:
- *Convolutional Layers:* Feature maps are produced by applying various filters to the input. These filters pick up on regional characteristics including patterns, textures, and edges.
- *Layers of Activation:* These are usually ReLU layers that provide non-linearity to the model so that it may pick up more intricate patterns.
- *Pooling Layers:* To make room for the subsequent convolutional layer, reduce the width and height of the input volume. The two most common types are average and maximum pooling.

Benefits

- *Feature Learning:* CNNs do not require human feature extraction; instead, they automatically identify and learn the key features.
- *Efficiency:* CNNs are more efficient to train than fully connected networks since they require fewer parameters because of parameter sharing and pooling.
- *Robustness:* CNNs can withstand minor rotations, translations, and other types of visual distortion in the input.

Drawbacks

- *Large Dataset Requirement:* For CNNs to train efficiently, a sizable amount of labeled data is frequently needed.
- *Computational Intensity:* CNN training can be a computationally demanding process that frequently calls for strong GPUs.
- *Black Box Nature:* CNNs, like many DL models, have a tendency to be viewed as "black boxes," making it challenging to understand how they determine a particular conclusion.

3.2 Recurrent Neural Networks(RNNs):

Artificial neural networks that are specifically made for processing sequential data are known as RNNs. This type of network is especially useful for jobs where the sequence and context of data points are crucial. DL activities that involve sequences, such as speech recognition, natural language processing, and time series analysis, frequently employ RNNs.

The main features of recurrent neural networks are as follows:

- *Processing Sequential Data:* RNNs[35] feature a memory that stores information about previous calculations, unlike regular neural networks. For sequential data, where the current output depends on both the current and prior inputs, they are therefore perfect.
- *Design:* A loop, which enables information to persist from one network step to the next, is a fundamental component of an RNN's design. An RNN's neurons or units get input from both the previous step's hidden state (output) and the data in the current phase.
- *Hidden States:* Each step's output, or hidden state, acts as a kind of memory. Because it includes details about earlier inputs, the network can make decisions by considering the entirety of the data sequence that it has received thus far.

Benefits

- *Sequence Length Flexibility:* RNNs are capable of processing input sequences with different lengths.
- *Contextual Information:* They are strong at sequential tasks because they can process incoming sequences using their internal state, or memory.

3.3 Fully Connected Neural Networks(FCN)

Often used in deep learning (DL), an FCN is an artificial neural network design that is often referred to as a multilayer perceptron (MLP) or a dense network. Every neuron in this network is connected to every other neuron in the layer above it.

The following are the main features of FCNs:

FCNs typically consist of three layers: an input layer, one or more hidden layers, and an output layer [36]. Every layer's neurons, or nodes, are completely coupled to every other layer's neurons.

Layers and Neurons

- Represents the input data in the input layer. This layer's neuron count is equal to the number of features in the input data.
- The final output is produced by the output layer. Depending on the task, this layer has a different number of neurons (e.g., one neuron for binary classification, and many neurons for multiclass registration or regression).
- Activation functions: These functions give the network non-linearities, which let it learn intricate patterns. ReLU, Sigmoid, and Tanh are examples of common activation functions.
- Forward Propagation: This approach uses a network to transfer input data from the input layer to the output layer. Each neuron processes the weighted total of its inputs using an activation function.
- Backpropagation and Training: Backpropagation is used to train the network. To do this, the difference between the expected and actual outputs is computed, and then the result is sent back through the network to modify the weights. Usually, gradient descent or its variations are used for this technique.

Benefits

- Simplicity: Straightforward architecture and easy to understand.
- Flexibility: Suitable for a broad spectrum of jobs, from easy to moderately difficult.

Drawbacks

- Parameter-intensive: Needs a lot of parameters, or weights, which raises the computational expense.
- Prone to Overfitting: Especially in networks with several layers or neurons.
- Restricted Capabilities for Feature Extraction: FCNs do not naturally extract sequential or spatial features from data, in contrast to CNNs.

3.4 Autoencoders

DL uses autoencoders, a kind of neural network, to accomplish unsupervised learning. An autoencoder's main goal is to learn a condensed representation of the input data, usually for feature learning or dimensionality reduction. When you need to learn a compact representation of high-dimensional data, autoencoders come in handy.

The following are autoencoders' main features:

Autoencoder Types

- The main goal of basic autoencoders is to encode and decode input into and out of a lower-dimensional space.
- Sparse Autoencoders: To learn more resilient features, apply a sparsity constraint to the hidden layers.
- Denoising Autoencoders: Trained to eliminate noise from the input, thereby developing a more robust representation.
- In latent space, variational autoencoders (VAEs) offer a probabilistic way to describe an observation.
- The encoder component of the network compresses the input data into a representation in latent space. The input data is encoded in a reduced dimension using a compressed form.
- The network's decoder uses the latent space representation to recreate the input data. It returns the encoded data to its original dimension by decoding it.
- Latent Space Representation: The encoder's output is a bottleneck-related reppresentation of the input data in the latent space. A condensed version of the supplied data is this representation.

Benefits

- Effective Data Encoding: Data can be effectively compressed into a lower-dimensional space via autoencoders.
- Unsupervised Learning: They don't need labeled data to learn meaningful representations.
- Flexibility: Adaptable to different kinds of applications and data.

Drawbacks

- The danger of Overfitting: An overly strong network may memorize the input data instead of developing a useful representation.
- Lossy Reconstruction: If the latent space has substantially lower dimensions than the input space, the reconstructed output might not match the input exactly.

4.0 Conclusion

The study successfully illustrates how agricultural crop disease management could be revolutionized by deep learning and machine learning models. It demonstrates the superiority of DL techniques—particularly CNNs—in precisely identifying and categorizing crop diseases through a comparison of different algorithms. This provides a

promising avenue for increasing agricultural productivity and decreasing losses through technology-driven solutions. This exhaustive survey will help the researcher to have a better understanding of the potential models and to implement them for their future work.

References

[1] Beltran-Perez, C., Wei, H.-L., & Rubio-Solis, A. (2020). Generalized multiscale rbf networks and the dct for breast cancer detection. *International Journal of Automation and Computing*, *17*, 55–70.

[2] Sharma, P., Leigh, L., Chang, J., Maimaitijiang, M. & Caffé, M. (2022). Above-ground biomass estimation in oats using uav remote sensing and machine learning. *Sensors, 22*(2), 601.

[3] Wu, N., Zhang, Y., Na, R., Mi, C., Zhu, S., He, Y. & Zhang, C. (2019). Variety identification of oat seeds using hyperspectral imaging: Investigating the representation ability of deep convolutional neural network. *RSC Advances, 9*(22), 12635–12644.

[4] Kumar, S., Deo, M. M., Kumar, K., Rathore, M., Akram, M. & Pratap, A. (2023). Detection and classification of yellow mosaic disease in vigna mungo using convolutional neural network deep learning models. *Ecology Environment and Conservation, 29*, 177-181.

[5] Mallick, M. T., Biswas, S., Das, A. K., Saha, H. N., Chakrabarti, A., & Deb, N. (2023). Deep learning based automated disease detection and pest classification in indian mung bean. *Multimedia Tools and Applications, 82*(8), 12017–12041.

[6] Kumar, S., Kumar, K., Tewari, K., Sagar, P., Pandey, J., Shanmugavadivel, P., Rathore, M., Kumar, V., Akram, M., & Singh, A.K., (2022). Gene expression and biochemical profiling of contrasting vigna mungo genotypes against mungbean yellow mosaic india virus (mymiv). *Journal of Food Legumes, 35*(2), 107–116.

[7] Basak, J., Kundagrami, S., Ghose, T. & Pal, A. (2005). Development of yellow mosaic virus (ymv) resistance linked dna marker in vigna mungo from populations segregating for ymv-reaction. *Molecular Breeding, 14*, 375–383.

[8] Harika, S., Sandhyarani, G., Sagar, D. & Reddy, G. S. (2023). Image-based black gram crop disease detection. *In 2023 International Conference on Inventive Computation Technologies (ICICT)*, pp. 529–533. IEEE.

[9] Khan, H., Haq, I.U., Munsif, M., Khan, S.U. & Lee, M.Y. (2022).Automated wheat diseases classification framework using advanced machine learning

technique. *Agriculture, 12*(8), 1226. Retrieved from DOI:10.3390/agriculture12081226

[10] Kumar, D. & Kukreja, V. (2021). N-cnn based transfer learning method for classification of powdery mildew wheat disease. *In 2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pp. 707–710. IEEE.

[11] Wang, Y., Wang, H. & Peng, Z. (2021). Rice diseases detection and classification using attention based neural network and bayesian optimization. *Expert Systems with Applications, 178*(2), 114770.

[12] Sethy, P. K., Barpanda, N. K., Rath, A. K. & Behera, S. K. (2020). Deep feature based rice leaf disease identification using support vector machine. *Computers and Electronics in Agriculture, 175*(1), 105527.

[13] Panigrahi, K. P., Sahoo, A. K. & Das, H. (2020). A cnn approach for corn leaves disease detection to support digital agricultural system. *In 2020 4th International Con- ference on Trends in Electronics and Informatics (ICOEI)(48184)*, pp. 678–683. Retrieved from https://doi.org/10.1109/ICOEI48184.2020.9142871

[14] Xu, P., Fu, L., Xu, K., Sun, W., Tan, Q., Zhang, Y., Zha, X. & Yang, R. (2023). Investigation into maize seed disease identification based on deep learning and multi-source spectral information fusion techniques. *Journal of Food Composition and Analysis, 119*, 105254.

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

[16] Bevers, N., Sikora, E. J., & Hardy, N. B. (2022). Soybean disease identification using original field images and transfer learning with convolutional neural networks. *Computers and Electronics in Agriculture, 203*, 107449.

[17] Mustafa, H., Umer, M., Hafeez, U., Hameed, A., Sohaib, A., Ullah, S. & Madni, H.A. (2023). Pepper bell leaf disease detection and classification using optimized convolutional neural network. *Multimedia Tools and Applications, 82*(8), 12065–12080.

[18] Hridoy, R.H., Arni, A.D. & Hassan, M.A. (2022). Recognition of mustard plant dis- eases based on improved deep convolutional neural networks. *In 2022 IEEE Region 10 Symposium (TENSYMP)*, pp. 1–6 (2022). Retrived from https://doi.org/10.1109/ TENSYMP54529.2022.9864487

[19] Song, Y. & Lu, Y. (2015). Decision tree methods: Applications for classification and prediction. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/ PMC4466856/

[20] Navada, A., Ansari, A.N., Patil, S. & Sonkamble, B.A. (2011). Overview of use of decision tree algorithms in machine learning. *In 2011 IEEE Control and System Graduate Research Colloquium*, pp. 37–42.

[21] Biau, G. & Scornet, E. (2016). A random forest guided tour – TEST. Retrieved from https://link.springer.com/article/10.1007/s11749-016-0481-7

[22] Liu, Y., Wang, Y. & Zhang, J. (2012). New machine learning algorithm: Random forest. *In Information Computing and Applications: Third International Conference, ICICA 2012*, Chengde, China, September 14-16, 2012. Proceedings 3, pp. 246–252. Springer.

[23] Suthaharan, S. & Suthaharan, S. (2016). Support vector machine. Machine learning models and algorithms for big data classification: thinking with examples for effective learning, 207–235. Retrieved from DOI:10.1007/978-1-4899-7641-3

[24] Rish, I. (2001). An empirical study of the naive Bayes classifier. Retrieved from https://sites.cc.gatech.edu/home/isbell/classes/reading/papers/Rish.pdf

[25] Berrar, D. (2018). Bayes' theorem and naive bayes classifier. Encyclopedia of bioinformatics and computational biology. *ABC of Bioinformatics 403*, 412. Retrieved from DOI:10.1016/B978-0-12-809633-8.20473-1

[26] Peterson, L. E. (2009). K-nearest neighbour. *Scholarpedia*. Retrieved from http://scholarpedia.org/article/K-nearest neighbor.

[27] Bijalwan, V., Kumar, V., Kumari, P. & Pascual, J. (2014). Knn based machine learn- ing approach for text and document mining. *International Journal of Database Theory and Application, 7*(1), 61–70.

[28] Maulud, D. & Abdulazeez, A. M. (2020). A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends, 1*(4), 140–147.

[29] Christodoulou, E., Ma, J., Collins, G.S., Steyerberg, E.W., Verbakel, J.Y., & Van Calster, B. (2019). A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *Journal of Clinical Epidemiology, 110*, 12–22.

[30] Ying, C., Qi-Guang, M., Jia-Chen, L. & Lin, G. (2013). Advance and prospects of adaboost algorithm. *Acta Automatica Sinica, 39*(6), 745–758.

[31] Howley, T., Madden, M. G., O'Connell, M.-L. & Ryder, A.G. (2005). The effect of principal component analysis on machine learning accuracy with high dimensional spectral data. *In International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pp. 209–222. Springer

[32] Tharwat, A., Gaber, T., Ibrahim, A. & Hassanien, A. E. (2017). Linear discriminant analysis: A detailed tutorial. *AI Communications, 30*(2), 169–190.

[33] Fine, S., Singer, Y. & Tishby, N. (1998). The hierarchical hidden markov model: Analysis and applications. *Machine Learning, 32*, 41–62.

[34] O'Shea, K. & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.

[35] Medsker, L.R. & Jain, L. (2001). Recurrent neural networks. *Design and Applications, 5*(64-67), 2.

[36] Sainath, T. N., Vinyals, O., Senior, A. & Sak, H. (2015). Convolutional, long short-term memory, fully connected deep neural networks. *In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4580–4584. IEEE.