

Comparison of Artificial Intelligence Algorithms in Plant Disease Prediction

Rutuja Rajendra Patil, Sumit Kumar*, Ruchi Rani

Symbiosis International (Deemed University), Symbiosis Institute of Technology, Pune, Maharashtra 412115, India

Corresponding Author Email: er.sumitkumar21@gmail.com



<https://doi.org/10.18280/ria.360202>

ABSTRACT

Received: 15 July 2021

Accepted: 12 April 2022

Keywords:

artificial intelligence, machine learning, deep learning, plant disease, prediction

The occurrence or change in the diseases in a specific area can be predicted in advance with the help of plant disease forecasting model. This helps to undertake suitable management measures to avoid the losses well in advance. Disease forecasting predicts probable outbreaks or increased disease intensity over a period in a particular area. This technique helps in timely application of chemicals to plants, which also involve all activities of crop protection and intimate the farmers in the community via text messages or e-mail etc. means of communication. Environment controls the evolution and survival period of various pathogens. Environmental conditions like minimum leaf wetness duration, soil moisture, micro-level relative humidity etc. contribute in evolution of disease causing pathogens. Disease forecasting system thus helps in predicting and avoiding evolution and spread of diseases. This paper uses Machine Learning (ML) and Deep Learning (DL) algorithms to detect, classify and predict the possible pathogens/diseases in the particular type of crop/plant considering based on weather conditions. Temperature, moisture and humidity are the parameters taken into consideration. Convolution Neural Networks (CNN), Recurrent Neural Network (RNN), Artificial Neural Network (ANN), Support Vector Machines (SVM) and K-Nearest Neighbours (KNN) are the five algorithms implemented and compared based on the obtained output accuracy. ANN outperforms all the other algorithms compared in this paper with accuracy of 90.79%.

1. INTRODUCTION

Deep learning has made tremendous advancements in machine learning and computer vision areas. The use of deep learning in agricultural applications has gained rapid traction in recent years for a variety of tasks including identification, classification, detection, quantification, and prediction [1]. On the leaf scale, DL can be used to identify diseases and quantify pests; on the canopy scale, DL can be used to identify weeds and classify plants; and on the field scale, DL can be used to assess abiotic stress, monitor plant growth, and nutrient levels, and to calculate yield forecasts [2, 3]. Different deep learning architectures such as Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), Fully Convolutional Networks (FCN) and Convolutional Neural Networks (CNN) have been successfully applied to diverse research areas, including agriculture. CNN, however, is the most popular DL architecture according to the current analysis [4]. Regional CNN (R-CNN) avoids the problem of having to select many regions to classify, instead selecting only a few regions for classification. This reduces the time needed for classification [5].

These ML and DL techniques are also used in agricultural sector mainly in disease forecasting. Various types of pests, fungi, viruses, bacteria, etc. agents cause various diseases in plants. Rotting of roots, fruits, blights on the leaf, spots on leaves or fruits, the decline in the yield of the crop is some of the symptoms of diseased plants [6]. Since ancient time environmental factors are always considered having major impact on disease development. Factors from the soil as well as the aerial environment are considered. Light intensity,

rainfall, duration of available sunlight, temperature, and humidity of that particular area are some of the factors included in the aerial environment. In short, it is the main deciding factor of the total epidemic process and the host-pathogen interaction takes place within the environment. For example, persistent optimum temperature and moisture are needed for the host. Optimum temperature, moisture, light and specific nutrition are also equally important for the development of the disease and epidemic. Despite satisfying all these conditions, the particular fungi will not grow until and unless it has favorable climatic conditions for its growth. Hence congenial environmental conditions, are an important aspect in disease evolution and its spreading. The environmental conditions also have a major impact on soil. The terminology for this is the edaphic environment. Pythiaceus fungus cause damping of soil. The propagules require a fixed saturation period for their germination. Free running water is a must for mobility and release of zoospores. A critical temperature above a specific level is needed for all these activities. Similarly, *Phytophthora cinnamomi*, require optimum soil temperature 150 degrees Celsius for successful infection. The fertility of soil plays a major role not only in soil borne disease but also in airborne diseases. Plants grown under poor fertility conditions are more prone to attack by some facultative pathogens than when they are growing vigorously. Rusts, powdery mildew, etc. are a few of the biotrophic pathogens that attack healthy, well-fertilized plants instead of unhealthy ones. *Rhizoctonia solani* is a type of soil borne pathogen that is mainly developed in sandy soils. Hence the level of organic content in soil matters. There exist some cases wherein soil rich in organic content and having higher

acidic pH, contain antagonistic microorganisms. These microorganisms help reduce the pathogen's surviving time and hence the spread of disease on a larger scale. Herbicide residue is a pollutant belonging to aerial environment that when in contact with soil, damages the plants.

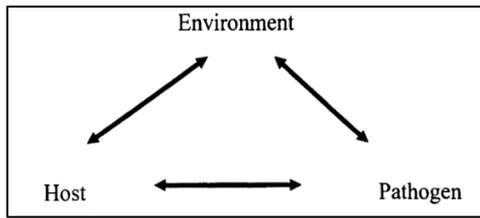


Figure 1. Plant disease triangle

The plant disease triangle as shown in Figure 1 illustrates the relation between environment, pathogens, and host, the three main factors constituting the evolution of diseases [7]. An illustration of the disease triangle would be a rice crop grown in areas with high relative humidity (86–100%) and temperature between 16 to 36°C is infected with rice blast disease. The host is the rice crop, the pathogen is the rice blast (fungus), and the environment is the rice field's relative humidity and temperature parameters. All three of these factors create the perfect atmosphere for a disease to thrive.

All three of these factors create the perfect atmosphere for a disease to thrive. All the three parameters are interdependent. The host, affects the seed of development of the disease. It is independent whether the host is absent or involved partially or totally. A canopy structure best explains the role of the host. This influences the spread of disease over the plants. The quantity of damage suffered by the crop depends on the density of the crop and the effects of wind in that area. The amount of wind over an area and the density of crops are the most sensitive areas for the entry of pathogens.

Verma et al. [8] proposed the image processing and IoT-based techniques and prediction models for identifying, detecting, and classifying the diseases on a tomato plant are surveyed. In Ref. [9], a novel approach for preventing the crop disease (Groundnut Crop) is based on IoT and Machine Learning is proposed. Humidity and Temperature sensor is deployed to verify the humidity and the atmospheric temperature of the plant whereas soil moisture sensor is deployed to get status of the soil. Sensors, webcam, GSM and controllers are used for receiving the data from the groundnut farm. The received data is analyzed and predicted for any possible disease occurrence using machine learning models (XG boost). The prediction is intimated to farmers through SMS/E-mail. In Ref. [10], the SISALERT forecasting system is designed. It is a generic web-based model. The model monitors the weather hour-by-hour and gathers the station data with the help of risk assessment models. The weather data is interpreted based on past/recent dataset reports and the risk of predicted disease. This model is run depending on the coupling of crop and disease models. This is the unique feature provided by the SISALERT system.

Traditional disease classification methods are done using machine learning only consider areas having a single crop for cultivation. For example, a model to extract features and classify the tomato powdery mildew disease on the leaves of tomato plants is developed by Raza et al. [11] considering the thermal and stereo images of the plants; RGB images are used to detect the same disease [12]; the detection of apple scab

using RGBD is carried out by Chéné et al. [13] and the same is achieved using sensors based on the aircraft [14]. A prediction model for detecting the yellow leaf curl virus in tomato plants is presented. SVM pipeline classifier is used for classification in this model.

Automatic selection of useful data from bigger data repositories is done using data mining. Nowadays almost every field be it medical, agricultural, environmental, technology, all implement data mining techniques. The authors Shobha and Asha [15] have proposed clustering approach for monitoring weather-based meteorological data. Naseri and Hemmati [16] developed a forecasting model based on fuzzy logic structure. Fuzzy logic defines the linguistic variables. These variables make it easier to predict accurate results as per expectations. In Ref. [17], a technique to detect the chemical composition present in rice samples is proposed. Data mining techniques such as MLP, RBF, ANN, CNN are used as classifiers [18].

The use of the Convolution Neural Network has improved accuracy in image classification in several fields, including agriculture and specifically plant disease detection. As a result of their high accuracy, CNNs are commonly used to identify and categorize images [19]. CNN is used to identify weed, diseases, and pests, supports irrigation management, yield prediction. It is also used as backbone architecture in finding out the exact location of the infection.

The common methods of disease forecasting [20, 21] are as follows:

1) Forecasting based on primary inoculums: The density, viability, and presence of primary inoculum can be detected in soil, air, or material of plantation. Using different air trapping devices (spore traps) propagules present in the air can be checked. Monoculture method is used for determining the primary inoculum in soilborne diseases.

2) Forecasting based on weather conditions: Different parameters such as the velocity of wind, amount of rainfall, light available, temperature and humidity are measured during crop and inter-crop season. The conditions of weather above and beneath the soil are recorded.

3) Forecasting depending on correlative information: Dataset of weather report gathered over several years is collected and correlated according to the intensity of the diseases. Comparing the data in the dataset, forecasting of diseases is carried out. Certain diseases such as barley powdery mildew, fire blight of apple, etc. have certain fixed criteria set by the meteorological departments. These criteria are set by comparing observed disease information along with standard data available with the meteorological department.

4) Forecasting with the help of computers: Certain countries prefer using computers for forecasting because of their higher accuracy and precision. 'Blitecast' is one example of a computer program-based model used for the blight of potato in the USA.

Most of the researchers have worked on identifying the diseases on the crops, which is easier than classifying the disease category. The agricultural experts sometimes fail in classifying the infection. Also, it has been identified that very few researchers have focused on grading the severity level of the disease. Predicting the disease on the crop on an early basis is a difficult task, and hence it has become one less researched area in the agricultural domain. Also, there is a scarcity of real-world applications that farmers can use. The applications that are already available lack in meeting the farmers' expectations. Also, the hardware and computational complexity issues make

the system more complex to be used by the farmers. So, taking into consideration the design and implementation issues in the agricultural domain becomes motivation to explore the Artificial Intelligence in the agricultural field.

The comparative analysis of CNN, RNN, ANN, SVM and KNN algorithms is performed in the paper. Various evaluation parameters like F1 score, Accuracy, Recall, and Precision are used to gauge the performance of the models.

Out of the above-mentioned methods, this paper focuses on forecasting based on weather conditions. Section 2 presents the theoretical background of algorithms used for disease prediction. Section 3 evaluates the algorithms discussed in section 2 and finds the algorithm that outperforms for crop disease prediction. Section 4 discusses the observations, and finally, section 5 concludes the paper.

2. MATERIALS AND METHODS

2.1 Convolutional Neural Network (CNN)

Convolutional neural network (CNN) is extensively preferred in deep learning architecture. It utilizes perceptrons for breaking down the gathered information. CNN is a progression of layers. Each volume transforms one volume into another by a differentiable limit. Pooling layers as well as convolution layers are the key layers present in CNN. Additionally, there exist few layers such as normalization layers and so on. Encodings are learned efficiently in ANN by using autoencoders. Auto encoders have one encoding and decoding phase and reconstruct their inputs on their own [22]. Figure 2 depicts the CNN architecture.

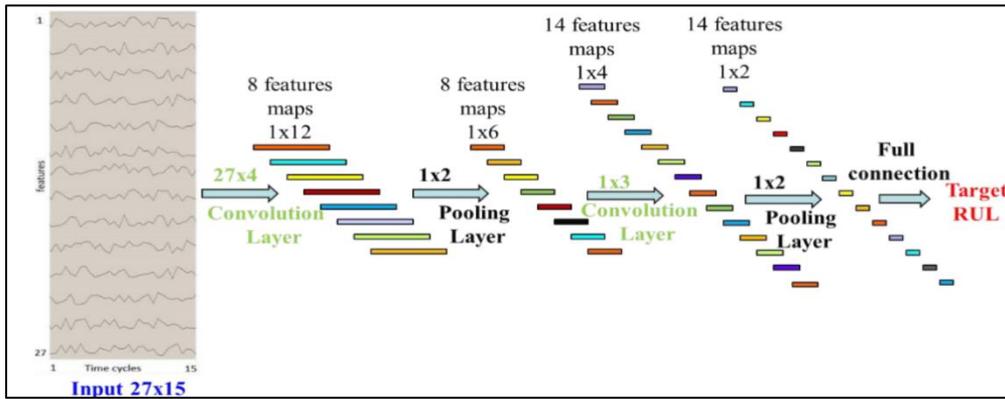


Figure 2. CNN architecture

Convolution layer: This layer constitutes the convolution of feature maps of the previous layer with the kernel. Hence named as convolution layer. Activation function constitutes the computation of bias added convolution layer operators and next layer feature map. The computation of convolution layer output feature map of is given in Eq. (1).

$$x_j^l = \text{sigm}(z_j^l), z_j^l = \sum_i x_j^{l-1} * k_{ij}^l + b_j^l \quad (1)$$

where, * denotes the convolution operator, x_j^{l-1} and x_j^l represents the convolution filter input as well as output, $\text{sigm}()$ is the sigmoid function, and z_j^l resembles input of non-linear sigmoid function. The Sigmoid function is preferred cause of its simplicity. D x 4 sized convolution filter is applied in the first layer whereas 1 x 3 sized filter is applied in the second layer.

Pooling Layer: Here sub-sampling of input features takes place. This is done to reduce the resolution of feature maps and therefore increase the invariance of those features. Partitioning of feature maps towards the input is carried out by the average pooling process. This produces sets of non-overlapping regions. Each sub-region has average value output. The output feature map of the pooling layer is computed in Eq. (2).

$$x_j^{l+1} = \text{down}(x_j^l) \quad (2)$$

where, x_j^l is the input and x_j^{l+1} is the output of the pooling layer, and $\text{down}()$ represents the sub-sampling function for average pooling. The filter size for the pooling layer is 1 x 2 for both the first and second layer.

2.2 Training Process

The squared error loss function is used for training and is represented in Eq. (3).

$$E = \frac{1}{2} (y(t) - y^*(t))^2 \quad (3)$$

where, $y^*(t)$ is the predicted RUL value and $y(t)$ is the target RUL of the training sample. Parameters of the network are estimated using the optimization method named, stochastic gradient descent. The loss function is minimized using a back propagation algorithm. Forward propagation, backward propagation, and application of gradients are cascaded processes in CNN training.

Forward Propagation: The output is predicted using segmented multi-variate time series input. Output feature maps of each layer are computed.

Each CNN layer comprises convolution and pooling layers. The output of these two layers is calculated by using the Eqns. (1) and (2).

Backward Propagation: Error value and squared error loss function are obtained on completion of one forward propagation iteration. This error propagates back from the last layer to the first layer. Generally, derivative chains are used for this. For the backward propagation of errors in the second stage pooling layer, the derivative is calculated by the sub-sampling function $\text{up}()$, it is an inverse operation of the sub-sampling function $\text{down}()$ and can be represented by Eq. (4).

$$\frac{\partial E}{\partial x_j^{l-1}} = \text{up} \frac{\partial E}{\partial x_j^l} \quad (4)$$

In the second stage of the feature extraction layer, z_j^l 's derivative is calculated by Eq. (5).

$$\partial_j^l = \frac{\partial E}{\partial z_j^l} = \frac{\partial E}{\partial x_j^l} \frac{\partial x_j^l}{\partial z_j^l} = \text{sigm}'(z_j^l) \odot \text{up} \frac{\partial E}{\partial x_j^{l+1}} \quad (5)$$

In Eq. (5), \odot the symbol denotes an element wise product. Bias derivative is calculated by adding all values ∂_j^l as in Eq. (6).

$$\frac{\partial E}{\partial b_j^l} = \sum_u \partial_j^l \quad (6)$$

The kernel weight k_{ij}^l 's derivative is calculated by adding all kernel values. With convolution operation, it is calculated using Eq. (7).

$$\frac{\partial E}{\partial x_j^{l-1}} = \sum_j \frac{\partial E}{\partial z_j^l} \frac{\partial z_j^l}{\partial x_j^{l-1}} = \sum_j \text{pad}(\partial_j^l) * \text{reverse}(k_{ij}^l) \quad (7)$$

In the above equation $\text{pad}(\cdot)$ denotes the padding function, zeros are padded to ∂_j^l at both ends. Generally, $\text{pad}(\cdot)$ function will pad at each end of ∂_j^l with $n_{\frac{l}{2}} - 1$ zero, where $n_{\frac{l}{2}}$ is the size of k_{ij}^l .

Apply Gradients: Once the derivatives of various parameters are calculated, they are further applied to update those parameters. Assume $E(w)$ represents the cost function to be minimized. The weights w is therefore modified towards steepest descent in Eq. (8).

$$w_{ij}^l = w_{ij}^l \eta \frac{\partial E}{\partial w_{ij}^l} \quad (8)$$

where, η is the learning rate. The variation between the current weight value and updated value is determined by this learning rate. The higher the learning rate, the larger will be the modification of weights w_{ij}^l .

The sequence of past observations from the environmental parameters is fed to the CNN model. The model learns that sequence and trains the model which knows how the sequence works and then it predicts the future values. To do so CNN uses all the equations mentioned from Eqns. (1) to (8). These equations are the modelling of CNN to train and test it and give results accordingly.

2.3 Recurrent Neural Network (RNN)

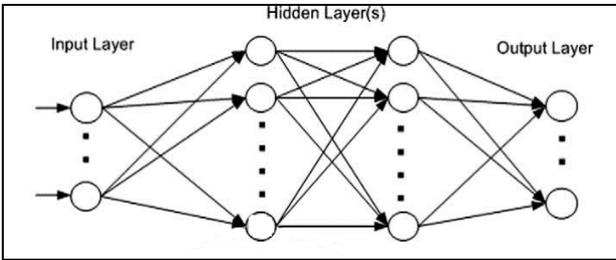


Figure 3. General architecture of RNN

The RNN can not only predict the very next time step but also generate a sequence of predictions and utilizes multiple driving time series together with a set of static features as its input. Time series of past observed environmental values have

been fed to the RNN model to train itself. Then trained model is used to predict the result. In RNN, the previous step output is the current state input. Generally, in all other ML algorithms, the input is independent of the outputs. To predict the next value of climatic parameters, previous values of climatic parameters are needed to be remembered. RNN helps solve this issue by introducing the hidden layer concept. The main feature of the hidden layer is remembering the previous sequence of information. Hence, RNN has a memory element to remember all the previously computed data. All input-output states use the same parameters, as similar tasks are performed on input-output and hidden layers [23]. This, in turn, helps reduce the complexity of parameters and RNN structure. Figure 3 represents the general architecture of RNN.

To illustrate RNN when applied to plant disease prediction, consider the environmental parameter sequence {30, 31, 32, 32, 33} as input to RNN structure. This is a five values sequence. Hence first four parameters are provided as input and RNN is expected to predict the last one. Hence the task has four parameters {30, 31, 32, 33} as 32 is repeated it will be considered as one value. It applies the recurrence formula on the input, considering the previous output state too. "30" being the first parameter, has no previous data. Taking parameter "31", the formula is applied on current state "31" and previous state "30". i.e., at time t , input is "31", at time $t-1$, it is "30". The formula thus applied on 31 and 30, gives a new output state to the structure.

The formula for the current state is calculated using Eq. (9).

$$h_t = f(h_{t-1}, x_t) \quad (9)$$

Here, h_t is the new state, h_{t-1} is the previous state while x_t is the current input. Hence, the input neuron applies transformations on previous inputs to obtain the previous input state of the input being considered.

Consider RNN network having \tanh as activation function, the recurrent neuron weight being W_{hh} , and the input neuron weight as W_{xh} , the equation for the state at time t is expressed using Eq. (10).

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \quad (10)$$

Each time immediate previous state output is considered by the RNN network thus used. Equation of longer sequences has multiple such states. The output stage is evaluated using Eq. (11).

$$y_t = W_{hy}h_t \quad (11)$$

Eq. (11) gives the result which is dependent on the weights and value of neurons of all the previous layer's values.

2.4 Artificial Neural Network (ANN)

ANN uses Logistics Regression (LR) for binary classification whereas Softmax Regression (SR) for multiclass classification. The neural network in multi-class classification has the same number of output nodes as the number of classes. Each output node is associated with a class and provides a score for that class. A softmax layer is used to pass the scores from the previous layer. The score is converted into probability values via the softmax layer. Finally, the data is categorized into the class with the highest probability value [24].

To design a SR model, the following parameters need to be focused on:

Number of nodes in dense layer: It is equal to the number of classes, which means that each class has its node.

Activation function: Softmax activation function is used while implementing plant disease forecasting. Softmax scales the values of the output nodes such that they represent probabilities and sum up to 1.

Loss function: The Loss function applied is categorical cross entropy. Categorical cross entropy is the generalization of binary cross entropy to more than 2 classes as this is multiclass classification.

2.5 Support Vector Machine

SVM is the abbreviation for Support Vector Machine. SVM falls under the supervised learning category of ML algorithms mainly used in classification and regression analysis. It is generally referred to as discriminant classifier. It has less computational complexity. Data is separated into various classification classes with the help formation of hyperplane. A line or hyperplane separating the classes is formed depending on the input data. Recognizing handwriting, classifying news articles, web pages, emails, face detection are some of the applications of SVM. Applying SVM on the input data, support vectors i.e. the closest points to the line/hyper plane are calculated from both classes. Distance between all these vectors and the line is computed to obtain the margin. The main goal is to maximize the margin [25]. The hyperplane having maximum margin is considered as optimal hyperplane as shown in Figure 4.

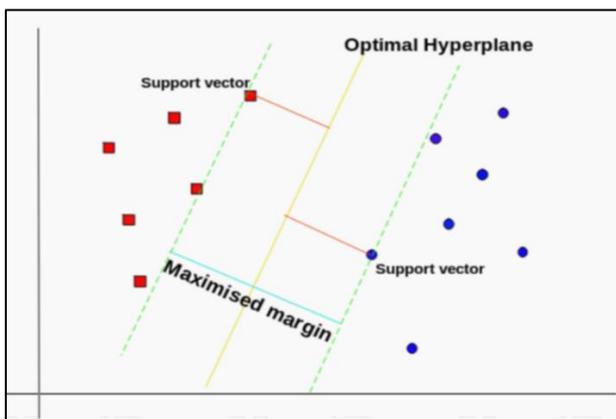


Figure 4. Optimal hyperplanes using the SVM algorithm

SVM is widely used in ML mainly due to its excellent ability to classify the input along with high accuracy rates. Classification is based on ‘widest street approach. The decision boundary of a linear classifier is represented using Eq. (12).

$$q \cdot x + b = 0 \quad (12)$$

where, ‘q’ is the weight vector, ‘x’ is the training/test pattern and ‘b’ is the bias term. Positive and negative samples with maximum interclass distance i.e. margin, are separated by the hyperplane. Margin (d) is calculated using Eq. (13).

$$\text{Margin } (d) = \frac{2}{\|q\|} \quad (13)$$

The minimum the value of “||q||”, maximum will be the distance i.e., the margin. This value is calculated using Eq. (14).

$$\text{Minimise } = \frac{1}{2} \|q\|^2 \quad (14)$$

At a new instance say x, the classifier function can be represented as given in Eq. (15).

$$f(x) = \text{sign}(q \cdot x + b) \quad (15)$$

By determining the best hyper-plane for separating the two classes, linearly separable data may be simply classified. In the case of non-separable data, ‘Kernel Functions’ are used for non-linear mapping. Gaussian Radial Basis Kernel Function (GRBKF), Polynomial Kernel Function (PKF), Exponential Radial Basis Kernel Function (ERBKF), Sigmoid Kernel Function (SKF), and Linear Kernel Function (LKF) are some of the existing kernel functions. The key challenge is to obtain the best values of “cost,” “epsilon,” and “gamma” parameters. The penalty of support vectors is resembled by “cost”, whereas effects of training examples are given by “gamma”. “Epsilon” resembles the smoothness in the resultant output of SVM.

2.6 K-Nearest Neighbor (KNN)

K-nearest Neighbor (KNN) is the easiest implemented ML algorithm under the supervised learning category. Also known as sample-based learning technique. KNN is an algorithm that assumes that similar objects are close together. As a result, if one data point is close to another, it is assumed that they both belong to the same class. Prediction and classification of the target value are performed based on stored data and distance function. Makowski, Manhattan, Euclidean, etc. are some of the generally used distance functions. Distance between input or the sample to be predicted and training points are evaluated using a distance function. Points having the smallest distance (as the name suggests, k- nearest neighbors) are considered. The target value is thus obtained by adding all these selected k neighbors. The only drawback of KNN is having high time and space complexity. This is mainly due to the use of all dataset samples every time while predicting [26]. Figure 5 illustrates the pictorial representation of KNN implementation.

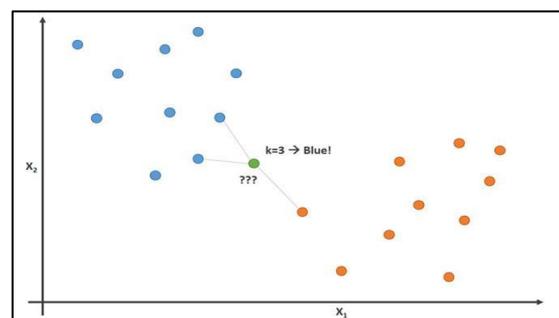


Figure 5. KNN implementation

For example, consider a labeled dataset having (x, y) as its training observations. Depending on this a KNN module can be designed such that a function h(x) can be defined as,

$$h: X \Rightarrow Y$$

This implies that, for a given unknown observation “x”, function $h(x)$ can accurately predict the corresponding output “y”.

KNN algorithmic steps:

Load the data.

Set ‘K’ to the number of neighbors to be considered in a cluster.

For each data point in the data set,

From the data, calculate the distance between the query example and the current example.

Add the distance and the index of the example to an ordered collection.

Using the distances, sort the ordered collection of distances and indices from smallest to greatest (in ascending order).

From the sorted collection, select the first K elements.

Get the labels for the K entries selected.

If classification, return the mode of the K labels.

3. RESULTS

A confusion matrix is preferred in evaluating the performance of classification models used in machine learning. The confusion matrix reveals how many classes have been correctly categorized and how many have been misclassified. The numbers on the diagonal axis represent the number of correctly identified points, whereas the remaining points are incorrectly classified. This matrix is a two row-two column matrix, obtained by applying the respective classifier model on a set of test data, for which true values are known. True positive (TP), true negative (TN), false positive (FP), and false negative (FN) values are calculated. Depending on these parameters such as accuracy, precision, f1 score, sensitivity, specificity, etc. are evaluated further [27, 28]. The formulae for all the evaluating parameters considered in this paper are stated below.

Sensitivity is also commonly referred to as True Positive Rate (TPR), Recall, or hit rate. It is denoted as:

$$TPR = \frac{TP}{P} = \frac{TP}{TP+FN} = 1 - FNR \quad (16)$$

Specificity is also known as True Negative Rate (TNR) or selectivity. It is denoted as:

$$TNR = \frac{TN}{N} = \frac{TN}{TN+FP} = 1 - FPR \quad (17)$$

Precision also known as Positive Predictive Value (PPV):

$$PPV = \frac{TP}{P} = 1 - FDR \quad (18)$$

Negative prediction value (NPV)

$$NPV = \frac{TN}{N} = 1 - FOR \quad (19)$$

Miss rate also known as false negative rate (FNR):

$$FNR = \frac{FN}{P} = \frac{FN}{FN+TP} = 1 - TPR \quad (20)$$

Fall-out or false positive rate (FPR):

$$FPR = \frac{FP}{N} = \frac{FP}{FP+TN} = 1 - TNR \quad (21)$$

False discovery rate (FDR)

$$FDR = \frac{FP}{FP+TP} = 1 - PPV \quad (22)$$

Accuracy (ACC):

$$ACC = \frac{TP+TN}{P+N} = \frac{TP+TN}{TP+TN+FP+FN} \quad (23)$$

F_1 score: This is given by the harmonic mean of precision and sensitivity.

$$F_1 = 2 \times \frac{PPV \times TPR}{PPV+TPR} = \frac{2TP}{2TP+FP+FN} \quad (24)$$

Matthew’s correlation coefficient (MCC) is also known as phi coefficient:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (25)$$

Table 1. Evaluation parameters obtained using CNN classifier

Evaluation parameters	Values obtained
Sensitivity (TPR)	0.8997
Specificity (TNR)	0.4412
Precision (PPV)	0.9847
Negative Predictive Value (NPV)	0.0993
Fall out (FPR)	0.5588
False Discovery Rate (FDR)	0.0153
Miss Rate (FNR)	0.1003
Accuracy (ACC)	0.8885
F1 Score	0.9403
phi coefficient (MCC)	0.1692

Table 2. Evaluation parameters obtained using RNN classifier

Evaluation parameters	Values obtained
Sensitivity (TPR)	0.8867
Specificity (TNR)	0.6364
Precision (PPV)	0.9969
Negative Predictive Value (NPV)	0.0412
Fall out (FPR)	0.3636
False Discovery Rate (FDR)	0.0031
Miss Rate (FNR)	0.1133
Accuracy (ACC)	0.8848
F1 Score	0.9386
phi coefficient (MCC)	0.1411

Table 3. Evaluation parameters obtained using ANN classifier

Evaluation parameters	Values obtained
Sensitivity (TPR)	0.915
Specificity (TNR)	0.4
Precision (PPV)	0.9909
Negative Predictive Value (NPV)	0.062
Fall out (FPR)	0.6
False Discovery Rate (FDR)	0.0091
Miss Rate (FNR)	0.085
Accuracy (ACC)	0.9079
F1 Score	0.9514
phi coefficient (MCC)	0.1291

Table 4. Evaluation parameters obtained using SVM classifier

Evaluation parameters	Values obtained
Sensitivity (TPR)	0.8723
Specificity (TNR)	0.4412
Precision (PPV)	0.9848
Negative Predictive Value (NPV)	0.0769
Fall out (FPR)	0.5588
False Discovery Rate (FDR)	0.0152
Miss Rate (FNR)	0.1277
Accuracy (ACC)	0.8622
F1 Score	0.9252
phi coefficient (MCC)	0.1391

Table 5. Evaluation parameters obtained using KNN classifier

Evaluation parameters	Values obtained
Sensitivity (TPR)	0.8603
Specificity (TNR)	0.4932
Precision (PPV)	0.9703
Negative Predictive Value (NPV)	0.1552
Fall out (FPR)	0.5068
False Discovery Rate (FDR)	0.0297
Miss Rate (FNR)	0.1397
Accuracy (ACC)	0.8421
F1 Score	0.912
phi coefficient (MCC)	0.2106

Table 6. Comparison of classifier models considering Evaluation Parameters

Classifier models	Evaluation parameters		
	Accuracy	Precision	F ₁ score
CNN	0.8885	0.9847	0.9403
RNN	0.8848	0.9969	0.9386
ANN	0.9079	0.9909	0.9514
SVM	0.8622	0.9848	0.9252
KNN	0.8421	0.9703	0.912

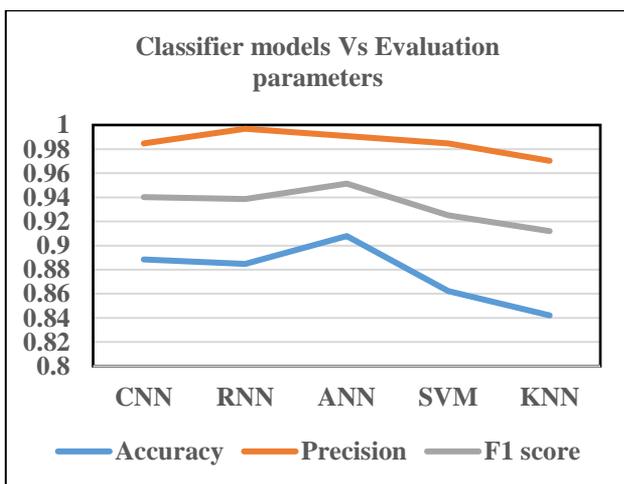


Figure 6. Classifier models Vs Evaluation parameters

This paper focuses on forecasting plant diseases based on weather conditions. Temperature, soil moisture, and humidity are the main parameters considered for experimentation. A prediction model considering certain weather condition

combinations is modeled using ANN. Classification of diseases is achieved using three machine learning algorithms namely ANN, SVM, and KNN. The output of the prediction model is the input for the classifiers. The data is split into training and testing sets as mentioned above. The five ML/DL algorithms are observed and judged concerning accuracy. Table 1 depicts the evaluation parameters obtained by using the CNN classifier model. Table 2 depicts the evaluation parameters obtained by using the RNN classifier model. Table 3 depicts the evaluation parameters obtained by using the ANN classifier model. Table 4 depicts the evaluation parameters obtained by using the SVM classifier model. Table 5 depicts the evaluation parameters obtained by using the KNN classifier model. Table 6 shows the comparison of the five ML/DL algorithms used concerning the accuracy, precision and F₁ parameters observed.

A graph is a plot of various classifier models against the evaluation parameters and shown in Figure 6. Considering three major parameters namely, accuracy, precision, and F₁ score, it is observed that ANN classifier model is the best among all the five classifier models experimented in this paper.

4. DISCUSSION

Before developing a forecasting model, one should decide the specific purpose of designing it and the various parameters to be considered such as particular area, type of climate, specific crop, disease, etc. Every plant disease forecasting model must be thoroughly tested and validated after being developed. Disease forecasting model development is a multi-disciplinary task. It depends on various factors such as (i) host factors (ii) pathogen factors (iii) weather factors (iv) soil factors (v) genetic factors (vi) Plant physiological factors etc. Synoptic weather stations help collect the required data on a macro-scale. This is a network of weather stations measuring various climatic conditions such as direction, speed, and atmospheric pressure of winds, amount of clouds, their type and base, temperatures, etc.

Five classifier models namely CNN, ANN, RNN, SVM, and KNN are used for disease prediction. The three main kinds of weathers conditions to be monitored are temperature, soil moisture, and humidity. For better accuracy and precision near about eleven evaluation parameters are evaluated. Sensitivity, specificity, precision value, accuracy, F₁ score, miss rate, negative prediction value, fall-out or false positive rate, false discovery rate, false omission rate, and Matthew's Correlation Coefficient are the evaluation parameters. Out of these, accuracy, precision, and F₁ score is considered while deciding the best suited classifier model.

Some recent online Disease Forecasting Models/DSS used in different countries:

EPIDEM- forecasting model for predicting alternaria solani disease on tomatoes & potatoes.

FAST- another forecasting model to predict Alternaria solani disease on tomatoes.

TOMCAST- predicts the presence of Alternaria, (Septoria, anthracnose).

WISDOM (BLITECAST)- predicts the presence of Late blight on tomato & potato plants.

MELCAST- predicts anthracnose, gummy stem blight on watermelons, and Alternaria on muskmelons.

Mary Blight- predicts the presence of fireblight on apple plants.

EPIVEN- predicts the presence of apple scab.

Blue mold warning system developed by North American to predict diseases on the tobacco plant.

There are various advantages of plant disease prediction models like increasing the annual income of farmers, Minimizing the losses of crops due to disease attacks. It serves as an alternative to pesticide spray scheduled over regular time intervals. It also guides in deciding the amount of spraying time of fungicide sprays to avoid the growth of diseases.

5. CONCLUSION

Technology nowadays has made immense advancements in the development of agro-based industries. It has also made it possible to grow crops in desert regions. Many researchers and companies have developed the latest automation techniques and solutions with the help of ML. In short, agriculture is now entering a digital era. To detect the presence of pathogens causing diseases and classifying them, deep learning techniques can be applied. This can help accurate prediction of required preventive measures such as spraying pesticides, using fertilizers, etc. to eliminate the possible present host from growing. Using ML/DL we can also predict the amount of spray to be sprayed at the targeted areas and also schedule the repetition of the same at specific time intervals. As Prevention is better than cure, this project aims in developing a forecasting model for predicting plant diseases/pathogens. Five types of ML/DL algorithms are used in this paper for disease forecasting based on weather conditions. Temperature, soil moisture, humidity are the parameters considered for experimentation. Comparing the five algorithms from the obtained results, it is observed that ANN has higher accuracy of 90.79% which is higher than that of the other algorithms used in plant disease prediction.

REFERENCES

- [1] Patil, R.R., Kumar, S. (2021). Predicting rice diseases across diverse agro-meteorological conditions using an artificial intelligence approach. *PeerJ Computer Science*, 7: e687. <https://doi.org/10.7717/peerj-cs.687>
- [2] Eli-Chukwu, N.C. (2019). Applications of artificial intelligence in agriculture: A review. *Engineering, Technology & Applied Science Research*, 9(4): 4377-4383. <https://doi.org/10.48084/etasr.2756>
- [3] Patil, R.R., Kumar, S. (2022). Priority selection of agro-meteorological parameters for integrated plant diseases management through analytical hierarchy process. *International Journal of Electrical & Computer Engineering* (2088-8708), 12(1): 649-659. <http://doi.org/10.11591/ijece.v12i1.pp649-659>
- [4] Shrestha, G., Das, M., Dey, N. (2020). Plant disease detection using CNN. In 2020 IEEE Applied Signal Processing Conference (ASPCON), pp. 109-113. <https://doi.org/10.1109/ASPCON49795.2020.9276722>
- [5] Boulent, J., Foucher, S., Théau, J., St-Charles, P.L. (2019). Convolutional neural networks for the automatic identification of plant diseases. *Frontiers in Plant Science*, 10: 941. <https://doi.org/10.3389/fpls.2019.00941>
- [6] Loey, M., ElSawy, A., Afify, M. (2020). Deep learning in plant diseases detection for agricultural crops: A survey. *International Journal of Service Science, Management, Engineering, and Technology (IJSSMET)*, 11(2): 41-58. <https://doi.org/10.4018/IJSSMET.2020040103>
- [7] Patil, R., Kumar, S. (2020). Bibliometric survey on diagnosis of plant leaf diseases using artificial intelligence. *International Journal of Modern Agriculture*, 9(3): 1111-1131.
- [8] Verma, S., Chug, A., Singh, A.P. (2018). Prediction models for identification and diagnosis of tomato plant diseases. In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1557-1563. <https://doi.org/10.1109/ICACCI.2018.8554842>
- [9] Yoganand, S., Narasingaperumal, P.R.P., Rahul, S. (2020). Prevention of crop disease in plants (Groundnut) using IoT and machine learning models. *International Research Journal of Engineering and Technology*, 7(3): 1164-1169.
- [10] Fernandes, J.M.C., Pavan, W., Sanhueza, R.M. (2011). SISALERT-A generic web-based plant disease forecasting system. In HAICTA, pp. 225-233.
- [11] Raza, S.E.A., Prince, G., Clarkson, J.P., Rajpoot, N.M. (2015). Automatic detection of diseased tomato plants using thermal and stereo visible light images. *PLoS ONE*, 10(4): 1-20. <https://doi.org/10.1371/journal.pone.0123262>
- [12] Hernández-Rabadán, D.L., Ramos-Quintana, F., Guerrero Juk, J. (2014). Integrating SOMs and a Bayesian classifier for segmenting diseased plants in uncontrolled environments. *Scientific World Journal*, 2014: 214674. <https://doi.org/10.1155/2014/214674>
- [13] Chéné, Y., Rousseau, D., Lucidarme, P., Bertheloot, J., Caffier, V., Morel, P., Belin, É., Chapeau Blondeau, F. (2012). On the use of depth camera for 3D phenotyping of entire plants. *Computers and Electronics in Agriculture*, 82: 122-127. <https://doi.org/10.1016/j.compag.2011.12.007>
- [14] Garcia-Ruiz, F., Sankaran, S., Maja, J.M., Lee, W.S., Rasmussen, J., Ehsani, R. (2013). Comparison of two aerial imaging platforms for identification of Huanglongbing-infected citrus trees. *Computers and Electronics in Agriculture*, 91: 106-115. <https://doi.org/10.1016/j.compag.2012.12.002>
- [15] Shobha, N., Asha, T. (2017). Monitoring weather based meteorological data: Clustering approach for analysis. *IEEE International Conference on Innovative Mechanisms for Industry Applications, ICIMIA 2017- Proceedings*, pp. 75-81. <https://doi.org/10.1109/ICIMIA.2017.7975575>
- [16] Naseri, B., Hemmati, R. (2017). Bean root rot management: Recommendations based on an integrated approach for plant disease control. *Rhizosphere*, 4: 48-53. <https://doi.org/10.1016/j.rhisph.2017.07.001>
- [17] Maione, C., Batista, B.L., Campiglia, A.D., Barbosa, F., Barbosa, R.M. (2016). Classification of geographic origin of rice by data mining and inductively coupled plasma mass spectrometry. *Computers and Electronics in Agriculture*, 121: 101-107. <https://doi.org/10.1016/j.compag.2015.11.009>
- [18] Sumathi, K., Depshikha, G., Dhivya, M., Karthika, P., Priyanka, B. (2021). Insect detection in rice crop using google code lab. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(2): 2328-2333. <https://doi.org/10.17762/turcomat.v12i2.1977>

- [19] Kumar, S., Patil, R.R., Kumawat, V., Rai, Y., Krishnan, N., Singh, S.K. (2021). A bibliometric analysis of plant disease classification with artificial intelligence using convolutional neural network. *Library Philosophy and Practice*, 2021: 1-14.
- [20] Orchi, H., Sadik, M., Khaldoun, M. (2021). On using artificial intelligence and the internet of things for crop disease detection: A contemporary survey. *Agriculture*, 12(1): 9. <https://doi.org/10.3390/agriculture12010009>
- [21] Fenu, G., Mallocci, F.M. (2021). Forecasting plant and crop disease: An explorative study on current algorithms. *Big Data and Cognitive Computing*, 5(1): 1-24. <https://doi.org/10.3390/bdcc5010002>
- [22] Khamparia, A., Saini, G., Gupta, D., Khanna, A., Tiwari, S., de Albuquerque, V.H.C. (2020). Seasonal crops disease prediction and classification using deep convolutional encoder network. *Circuits, Systems, and Signal Processing*, 39(2): 818-836. <https://doi.org/10.1007/s00034-019-01041-0>
- [23] Khaki, S., Wang, L., Archontoulis, S.V. (2020). A CNN-RNN framework for crop yield prediction. *Frontiers in Plant Science*, 10: 1750. <https://doi.org/10.3389/fpls.2019.01750>
- [24] Sharma, P., Singh, B.K., Singh, R.P. (2018). Prediction of potato late blight disease based upon weather parameters using artificial neural network approach. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-13. <https://doi.org/10.1109/ICCCNT.2018.8494024>
- [25] Ardila, C.E.C., Ramirez, L.A., Ortiz, F.A.P. (2020). Spectral analysis for the early detection of anthracnose in fruits of Sugar Mango (*Mangifera indica*). *Computers and Electronics in Agriculture*, 173: 105357. <https://doi.org/10.1016/j.compag.2020.105357>
- [26] Vaishnav, M.P., Devi, K.S., Srinivasan, P., Jothi, G.A.P. (2019). Detection and classification of groundnut leaf diseases using KNN classifier. In 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), pp. 1-5. <https://doi.org/10.1109/ICSCAN.2019.8878733>
- [27] Sambasivam, G., Opiyo, G.D. (2021). A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks. *Egyptian Informatics Journal*, 22(1): 27-34. <https://doi.org/10.1016/j.eij.2020.02.007>
- [28] Patil, R.R., Kumar, S. (2022). Rice-Fusion: A multimodality data fusion framework for rice disease diagnosis. In *IEEE Access*, 10: 5207-5222. <https://doi.org/10.1109/ACCESS.2022.3140815>