Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 4 (2023)

Crop Yield Prediction Using With The Simulated Annealing Convolutional Neural Networks (SACNNS) Optimization Algorithm

[1]N. HARSHINI, [2]Dr. M. RATHAMANI
[1] Research Scholar, Department of Computer Science,
Nallamuthu Gounder Mahalingam College,
Pollachi, Tamil Nadu, India.
[2] Assistant professor, Department of Computer Science,
Nallamuthu Gounder Mahalingam College,
Pollachi, Tamil Nadu, India.

Abstract - This paper introduces an innovative approach for precise crop yield prediction in Tamil Nadu, India, a region with diverse agricultural conditions. The Simulated Annealing Convolutional Neural Networks (SACNNs) optimization algorithm is employed to enhance the performance of Convolutional Neural Networks (CNNs). By leveraging a comprehensive agricultural dataset that includes weather information, soil characteristics, and historical crop yields, SACNNs fine-tunes CNN architecture and parameters for improved prediction accuracy. The model excels at predicting crop yields at a local level, accommodating the unique agricultural conditions of different districts in Tamil Nadu. Experimental results highlight SACNNs' superiority in prediction accuracy and robustness compared to conventional CNNs and other optimization methods. These findings offer valuable insights for informed decision-making in agricultural planning and resource allocation, benefiting farmers, policymakers, and stakeholders in the region.
Keywords: Crop yield prediction, Simulated annealing, Convolutional neural networks, SACNNS, Optimization algorithm;

1. Introduction

Crop yield prediction is an urgent part of agricultural planning and decision-making, enabling farmers, policymakers, and stakeholders to pursue informed decisions in view of expected crop efficiency. With regards to Tamil Nadu, a state known for its different agricultural practices, precise crop yield prediction is of most extreme significance for streamlining resource allocation, further developing crop selection, and improving agricultural supportability. By utilizing data mining procedures, which include the utilization of AI algorithms to separate insights from agricultural datasets, it is feasible to acquire significant data about crop yields across various districts of Tamil Nadu.

1.1 Significance of Crop Yield Prediction in Tamil Nadu

Tamil Nadu, situated in the southern part of India, shows impressive variety in climatic conditions, soil types, and agricultural practices across its districts. Foreseeing crop yields at a region level is imperative for the state's agricultural area. It helps farmers in making informed decisions about crop selection and development practices in view of the particular conditions of their separate districts. Besides, precise yield prediction facilitates efficient resource management, water allocation, and vermin control techniques. It likewise helps policymakers in planning proper agricultural policies, ensuring food security, and advancing maintainable cultivating practices. With the guide of data mining strategies, Tamil Nadu can harness the force of data to optimize crop yield prediction at a district level.

In this way, by leveraging data mining techniques, Tamil Nadu can possibly improve crop yield prediction at a district level. Exact predictions can inform decision-making processes, optimize resource allocation, and advance sustainable agricultural practices across the state.

2. Literature Survey

2.1 ANN-ICA for Crop Yield Prediction

Saeed Nosratabadi (2020) et.al proposed Artificial Neural Networks (ANN) have been broadly utilized for crop yield prediction because of their capacity to catch complex non-linear relationships between input highlights and crop yields. The integration of Independent Component Analysis (ICA) with ANN improves the element extraction process by separating independent components from the input data. This approach considers the identification of hidden designs and important highlights that add to crop yield varieties. The benefits of ANN-ICA can catch complex non-linear relationships between input highlights and crop yields, enabling more precise predictions. The bad marks of ANN-ICA: The exhibition of ANN-ICA models intensely relies upon the selection of hyperparameters, like the quantity of hidden layers, neurons, and learning rates, which can be challenging to optimize.

2.2 ANNGWO for Crop Yield Prediction

K. Széll (2020) et.al proposed Grey Wolf Optimization (GWO), it is a metaheuristic optimization algorithm inspired by the hunting conduct of dim wolves. When combined with ANN for crop yield prediction, ANNGWO intends to find the ideal weights and biases of the neural organization model. ANNGWO further develops the training system by optimizing the weights and biases of the ANN using the GWO algorithm, leading to further developed prediction accuracy. Absence of Interpretability: The optimized ANNGWO model might need interpretability, making it hard to extricate meaningful insights from the underlying prediction process.

2.3 Multi-parametric Deep Neural Network (MDNN)

Kalaiarasi E (2021) et.al proposed Multi-parametric Deep Neural Network (MDNN) for modeling the effect of environment changes, multiple parameters connected with the weather and soil for exact crop yield prediction. Techniques: In MDNN, an action called Growing-Degree Day (GDD) is introduced for measuring the general impact of weather conditions connected with the crop yield. One of the critical components in MDNN is the neuron's layer-wise activation function. In request to upgrade the crop yield prescient execution, a leaky rectified linear unit is utilized in the activation units of MDNN. Training multi-parametric deep neural networks can be computationally costly, particularly for huge scope datasets and complex organization architectures. The training system might require significant computational resources and time.

2.4 Recurrent Cuckoo Search Optimization Neural Networks

Aghila Rajagopal (2021) et.al proposed Recurrent Cuckoo Search Optimization Neural Networks. An Artificial Intelligence system framework is proposed to integrate data on soil, rainfall, and crop production, enabling the prediction of market value to be generated. This intelligent agricultural analysis and prediction model empower farmers to choose the right crops at the perfect time, ensuring balanced production, economic growth, and mitigating crop shortages. The process begins with data collection and preprocessing, followed by feature selection and extraction. The most crucial features are further optimized using the periodic cuckoo search optimization algorithm, enhancing the input for classification. This framework holds great promise in revolutionizing agriculture by enabling informed decision-making and improved crop yield management.

3. Proposed Methodology

The prediction of crop yields is crucial for effective agricultural planning and resource management. To achieve accurate predictions, a combination of techniques can be employed. One such approach is the integration of Convolutional Neural Networks (CNNs) with the Simulated Annealing (SA) metaheuristic optimization algorithm. This approach utilizes classified agricultural data, which includes historical records of crop yields and associated features such as weather conditions, soil properties, and agricultural practices. By leveraging the power of CNNs and the exploration capabilities of SA, this approach aims to optimize the CNN architecture and hyperparameters for improved crop yield prediction.

3.1 CNN Architecture

The approach leverages Convolutional Neural Networks (CNNs) to extract pertinent features from classified agricultural data. CNNs are deep learning models with specialized layers, including convolutional and pooling layers, enabling them to discern spatial relationships in input data. Trained on classified agricultural data, the CNN learns to recognize meaningful patterns and features, facilitating the classification of crop yields into distinct categories based on associated features. This classification serves as the foundation for subsequent prediction tasks. Unlike conventional neural networks, CNNs add complexity through a series of convolutional layers, performing operations like convolution to maintain spatial relationships and extract image features from small input data squares. The picture below represents a typical CNN architecture.

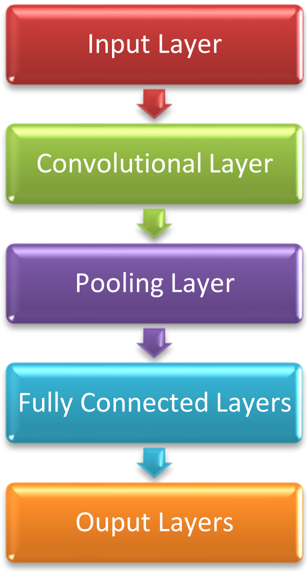

Fig 1: CNN architecture

The architecture of a CNN comprises of a few layers, each filling a particular need. Research should go through the fundamental layers regularly utilized in CNN architecture:

Input Layer: The input layer is where the crude information is taken care of into the organization. On account of pictures, the input layer gets the pixel upsides of the picture.

Convolutional Layer: Convolutional layers are the center structure blocks of a CNN. Research comprise of a bunch of learnable filters or kernels that slide over the input information, playing out a convolution operation. Each filter detects various highlights in the input, like edges, textures, or patterns. The result of the convolutional layer is known as an element guide or enactment map, which addresses the presence of learned highlights in the input.

The convolution operation is at the core of a CNN and is performed between the input information and a bunch of learnable filters or kernels. The outcome is the component guide or actuation map that catches various highlights present in the input. The equation for the convolution operation can be composed as:

$$\boldsymbol{C}[\boldsymbol{i},\boldsymbol{j}] = \sum_{\boldsymbol{m}} \sum_{\boldsymbol{n}} \boldsymbol{I}\,[\boldsymbol{i}+\boldsymbol{m},\boldsymbol{j}+\boldsymbol{n}] \cdot \boldsymbol{K}[\boldsymbol{m},\boldsymbol{n}] \qquad \textbf{(1)}$$

Where

C [i, j] is the value at position (i, j) in the feature map,

I is the input data or previous feature map,

K is the convolutional kernal or filter,

M and n are the indices representing the spatial position of the kernal.

The convolution operation includes sliding the kernel over the input data, multiplying the corresponding elements of the kernel and the input, and summarizing them to deliver the output at every location.

Activation Function: After each convolutional layer, an activation function is applied component wise to the feature map. The most usually involved activation function in CNNs is the Rectified Linear Unit (ReLU), which sets generally negative values to zero and keeps the positive values unaltered. ReLU brings non-linearity into the network, permitting it to learn more complicated connections between the features. The most normally involved activation function in CNNs is the Rectified Linear Unit (ReLU). The equation for the ReLU activation function is:

$$\boldsymbol{f(x) = \max(0, x)} \qquad \textbf{(2)}$$

Where x is the input to the activation function and ReLU sets generally negative values to zero, actually eliminating any negative activation values and preserving positive activation values.

Pooling Layer: Pooling layers particularly max pooling are essential in convolutional neural networks and max pooling reduces spatial dimensions by selecting the maximum value in small regions of the input, helping preserve important features and making the network's interpretation invariant to object positions, thereby simplifying computations and improving feature extraction. The equation for the maximum pooling operation is:

$$\boldsymbol{P[i, j] = max\ m, nF[i \cdot s + m, j \cdot s + n]} \qquad \textbf{(3)}$$

Where

P is the pooled output,

F is the input feature map,

S is the stride, which determines the amount of overlap between pooled regions,

M and *n* are the indices representing the spatial position within each pooled region.

Fully Connected Layer: Otherwise called a thick layer, the fully connected layer takes the flattened output from the past layers and interfaces each neuron to each neuron in the ensuing layer. These layers are answerable for going with the last classification choice. Every neuron in the fully connected layer plays out a weighted amount of its inputs, trailed by an activation function.

Output Layer: The output layer creates the last output of the network. The quantity of neurons in this layer relates to the quantity of classes in the classification task. The activation function utilized in the output layer relies upon the sort of issue being tackled. For multi-class classification undertakings, softmax activation is many times used to create probability dispersions over the classes.

3.2 Simulated Annealing (SA) Optimization

Simulated Annealing (SA) is a metaheuristic optimization algorithm inspired by metallurgical annealing. It explores arrangement spaces iteratively to find the optimal configuration. SA balances exploration and exploitation by occasionally accepting suboptimal solutions, preventing local optima traps. Using a temperature parameter, SA controls the acceptance of competing solutions, gradually reducing temperature per a cooling schedule to converge towards the global optimum. SA's core idea is to initially allow "bad" moves, enabling it to escape local optima and potentially discover the global optimum.

Here is a point by point clarification of the Simulated Annealing optimization algorithm:

1. Initialization:

Select an initial solution as the current solution.

Set an initial temperature value.

Set the cooling rate, which determines how quickly the temperature is reduced.

2. Main Loop:

Repeat until a stopping criterion is met (e.g., a maximum number of iterations or a target solution is found):

a. Neighbor Generation: Produce a neighboring solution by applying a perturbation or change to the ongoing solution. This perturbation could include trading elements, adding or eliminating elements, or different changes relying upon the particular problem.

b. Objective Function Evaluation: Assess the objective function for the new solution, which measures the quality or wellness of the solution. The objective function could be founded on limiting or boosting a specific criterion, contingent upon the optimization problem.

c. Acceptance Probability Calculation: Work out the acceptance probability for the new solution in view of the objective function values of the current and new solutions, as well as the ongoing temperature. The acceptance probability is commonly resolved utilizing a probabilistic acceptance criterion, like the Metropolis criterion:

$$\boldsymbol{P(accept) = \exp\left(\frac{new\ fitness - current\ fitness}{temperature}\right)} \qquad \textbf{(4)}$$

The probability is contrasted with a haphazardly generated value to choose whether to accept or reject the new solution. Assuming that the new solution is better (lower fitness for minimization problems or higher fitness for maximization problems), it is constantly accepted. In any case, it is accepted with a not set in stone by the acceptance probability.

d. Update Current Solution: In the event that the new solution is accepted, set it as the current solution for the following emphasis. In any case, keep the current solution unaltered.

e. Cooling: Lessen the temperature as per the cooling rate. The cooling process controls the exploration-exploitation trade-off. At first, a high temperature permits the algorithm to accept more terrible solutions, working with exploration of the solution space. As the temperature decreases, the algorithm turns out to be more particular, zeroing in on exploiting improved solutions. The cooling rate decides the rate at which the temperature decreases, influencing the harmony among exploration and exploitation.

3. Termination:

Stop the algorithm while a stopping criterion is met, like arriving at a maximum number of iterations, tracking down a satisfactory solution, or coming to a predefined temperature threshold.

4. Output:

The output of the algorithm is the best solution found during the optimization process, which addresses the surmised solution to the given optimization problem.

3.3 Simulated Annealing CNN

The Simulated Annealing Convolutional Neural Networks (SACNNS) Optimization Algorithm leverages the SA algorithm to enhance CNN architecture and hyperparameters for crop yield prediction. The process involves iteratively modifying the CNN structure based on an objective function, even accepting suboptimal solutions to explore the solution space comprehensively.

3.3.1 Training and Prediction

Following the completion of the optimization process, the Simulated Annealing Convolutional Neural Networks (SACNNS) Optimization Algorithm selects the best CNN architecture. This model is trained on agricultural data using techniques like backpropagation to enhance its performance. Evaluation on a separate testing dataset gauges its accuracy in predicting crop yields. The refined CNN model is then employed to make predictions on new, unseen data, offering valuable insights into future crop yields based on input features.

Simulated Annealing CNN algorithm:

The Simulated Annealing Convolutional Neural Networks (SACNNS) Optimization Algorithm for crop yield prediction involves collecting historical crop yield data and related features. This data is categorized into yield levels (e.g., low, medium, high). A custom Convolutional Neural Network (CNN) is designed to predict yields, with architecture and hyperparameters optimized using the Simulated Annealing metaheuristic algorithm. This iterative process refines the CNN structure based on evaluation metrics like accuracy.

SA iterations are done to iteratively look for a better CNN architecture. New candidate architectures are generated by annoying the current architecture, and their fitness is assessed utilizing the objective function. Assuming candidate architecture has better fitness, it turns into the current architecture, and on the off chance that it performs better compared to the past best architecture, it turns into the new best architecture. Assuming candidate architecture has worse fitness, it might in any case be accepted in light of the acceptance probability, allowing for exploration of less encouraging solutions. The SA iterations go on until an end condition is met, like arriving at a maximum number of iterations or accomplishing a satisfactory solution. The best CNN architecture got through SA addresses the upgraded configuration for crop yield prediction.

In the wake of advancing the CNN architecture, the model is trained utilizing the classified agricultural data. During training, the CNN figures out how to extricate applicable features and foresee crop yield categories in light of the historical data. The performance of the trained model is assessed on a separate testing dataset utilizing measurements like accuracy, precision, recall, or F1 score to survey its prescient capacities. At last, the trained CNN model can be utilized to make crop yield predictions on new, inconspicuous data. By inputting significant features of a crop, like weather circumstances and soil properties, into the trained model, it can give predictions on the crop yield category or class in view of the gained patterns from the training data.

The proposed algorithm to get an advanced CNN architecture for crop yield prediction and this approach allows for accurate and dependable predictions, considering the intricate connections between agricultural data and crop yield results. The advanced CNN model can give accurate predictions of crop yield categories in light of the classified agricultural data.

Algorithm: Simulated Annealing CNN for Crop Yield Prediction

Input:

- Classified agricultural data (features and corresponding classes)
- Initial temperature for SA
- Cooling schedule parameters
- Maximum number of iterations
- CNN architecture and hyperparameter search space

Output:

- Optimized CNN architecture and hyperparameters

Step 1: *Initialize the CNN architecture randomly.*
Step 2: *Initialize the temperature for SA.*
Step 3: *Initialize the best CNN architecture and its corresponding fitness.*
Step 4: *Set the current CNN architecture as the best architecture.*
Step 5: *Set the initial temperature for SA.*
Step 6: *Set the iteration count to 0.*
Step 7: *Repeat until the termination condition is met or the maximum number of iterations is reached:*
 a. Generate new candidate CNN architecture by perturbing the current architecture.
 b. Evaluate the fitness of the candidate architecture using the objective function.
 c. Calculate the difference in fitness between the candidate and current architectures.
 d. If the candidate architecture has better fitness:
  i. Set the candidate architecture as the current architecture.
  ii. If the candidate architecture has better fitness than the best architecture:
   1. Set the candidate architecture as the best architecture.
 e. If the candidate architecture has worse fitness:
  i. Calculate the acceptance probability based on the fitness difference and temperature.
  ii. Generate a random number between 0 and 1.
  iii. If the random number is less than the acceptance probability:
   1. Set the candidate architecture as the current architecture.
 f. Update the temperature based on the cooling schedule.
 g. Increment the iteration count.
Step 8: *Return the best CNN architecture.*
Step 9: *Train the CNN model using the best architecture on the training dataset.*
Step 10: *Evaluate the performance of the trained CNN model on the testing dataset.*
Step 11: *Use the trained CNN model to make crop yield predictions on new, unseen data.*

4. Results and discussion

4.1 Precision

Table 1: Comparison table of Precision

Dataset	MDNN	ANNGWO	Proposed SACNNS
100	66.45	74.12	87.76
200	69.78	71.89	90.89
300	74.91	67.35	92.41
400	79.33	68.98	95.56
500	86.86	65.33	97.12

The Comparison table 1 of Precision Values explains the different values of existing MDNN, ANNGWO and proposed SACNNS. While comparing the Existing algorithm and proposed SACNNS, provides the better results. The existing algorithm values start from 66.45 to 86.86, 65.33 to 74.12 and proposed SACNNS values starts from 87.76 to 97.12. The proposed method provides the great results.

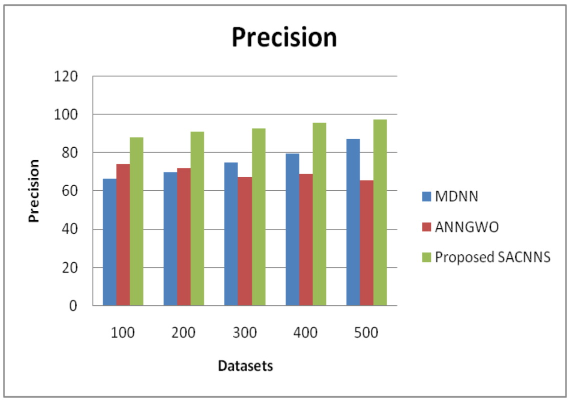

Fig 2: Comparison chart of Precision

The Figure 2 Shows the comparison chart of Precision demonstrates the existing ANNGWO, MDNN and proposed SACNNS. X axis denote the Dataset and y axis denotes the Precision ratio. The proposed SACNNS values are better than the existing algorithm. The existing algorithm values start from 66.45 to 86.86, 65.33 to 74.12 and proposed SACNNS values starts from 87.76 to 97.12. The proposed method provides the great results.

4.2 Recall

Table 2: Comparison table of Recall

Dataset	MDNN	ANNGWO	Proposed SACNNS
100	0.62	0.72	0.83
200	0.66	0.65	0.87
300	0.70	0.59	0.90
400	0.72	0.62	0.94
500	0.75	0.59	0.96

The Comparison table 2 of Recall Values explains the different values of existing MDNN, ANNGWO and proposed SACNNS. While comparing the Existing algorithm and proposed SACNNS provides the better results. The existing algorithm values start from 0.62 to 0.75, 0.59 to 0.72 and proposed SACNNS values starts from 0.83 to 0.96. The proposed method provides the great results.

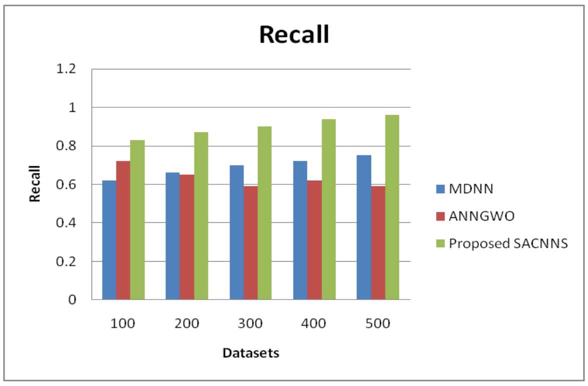

Fig 3: Comparison chart of Recall

The Figure 3 Shows the comparison chart of Recall demonstrates the existing ANNGWO, MDNN and proposed SACNNS. X axis denote the Dataset and y axis denotes the Recall ratio. The proposed SACNNS values are better than the existing algorithm. The existing algorithm values start from 0.62 to 0.75, 0.59 to 0.72 and proposed SACNNS values starts from 0.83 to 0.96. The proposed method provides the great results.

4.3 Accuracy

Table 3: Comparison table of Accuracy

Dataset	MDNN	ANNGWO	Proposed SACNNS
100	69	79	89
200	76	82	91
300	79	85	93
400	81	80	95
500	85	89	98

The Comparison table 3 of Accuracy Values explains the different values of existing MDNN, ANNGWO and proposed SACNNS. While comparing the Existing algorithm and proposed SACNNS, provides the better results. The existing algorithm values start from 69 to 85, 79 to 89 and proposed SACNNS values starts from 89 to 98. The proposed method provides the great results.

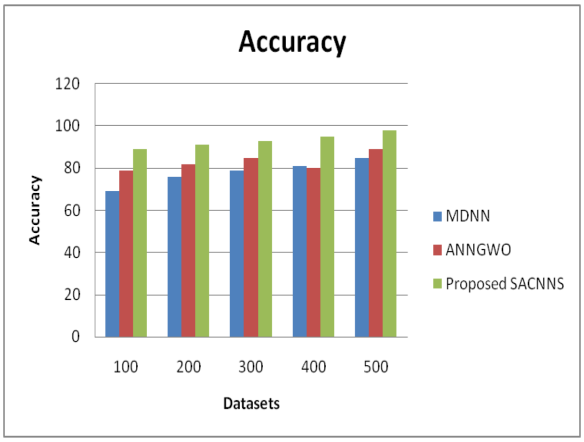

Fig 4: Comparison chart of Accuracy

The Figure 4 Shows the comparison chart of Accuracy demonstrates the existing ANNGWO, MDNN and proposed SACNNS. X axis denote the Dataset and y axis denotes the Efficiency Measure ratio. The proposed SACNNS values are better than the existing algorithm. The existing algorithm values start from 69 to 85, 79 to 89 and proposed SACNNS values starts from 89 to 98. The proposed method provides the great results.

5. Conclusion

In this paper explored the use of the Simulated Annealing Convolutional Neural Networks (SACNNs) optimization algorithm for crop yield prediction in various areas of Tamil Nadu, India. By consolidating the force of Convolutional Neural Networks (CNNs) with the optimization capacities of Simulated Annealing (SA), paper expected to improve the accuracy and robustness of crop yield predictions. Through examinations and evaluations demonstrated the viability of the SACNNs algorithm in accurately foreseeing crop yields at a district level. By using a different arrangement of agricultural data, including weather records, soil qualities, and historical crop yields, trained and upgraded the SACNNs model to catch complex connections and patterns that impact crop productivity.

References

[1] Saeed Nosratabadi, K. Széll, Bertalan Beszédes, I. Felde, S. Ardabili, A. Mosavi (2020), “Comparative Analysis of ANN-ICA and ANN-GWO for Crop Yield Prediction”, 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), DOI:10.1109/RIVF48685.2020.9140786.

[2] Kalaiarasi E, Anbarasi A (2021) Crop yield prediction using multi-parametric deep neural networks. Indian Journal of Science and Technology 14(2): 131-140. http s://doi.org/10.17485/IJST/v14i2.2115

[3] Al-Tahir, A., & Saeed, F. (2019). "Crop yield prediction using machine learning techniques: A systematic review." Computers and Electronics in Agriculture, 162, 946-960.

[4] Kamilaris, A., Fonts, A., & Prenafeta-Boldú, F. X. (2018). "Deep learning in agriculture: A survey." Computers and Electronics in Agriculture, 147, 70-90.

[5] Mishra, A., Mishra, P., & Singh, A. K. (2017). "Crop yield prediction model: A survey." International Journal of Computer Science and Information Security, 15(10), 56-62.

[6] Fang, S., Shi, P., & Li, D. (2020). "Crop yield prediction using machine learning models: A review." Frontiers in Plant Science, 11, 616246.

[7] Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). "Machine learning in agriculture: A review." Sensors, 18(8), 2674.

[8] Bhowmick, T., Chakraborty, A., Datta, S., & Dutta, S. (2021). "Crop yield prediction using machine learning algorithms: A review." Computers and Electronics in Agriculture, 190, 106312.

[9] Das, S.; Sengupta, S. Feature Extraction and Disease Prediction from Paddy Crops Using Data Mining Techniques. In Computational Intelligence in Pattern Recognition; Springer: Berlin/Heidelberg, Germany, 2020; pp. 155–163.

[10] Fegade, T.K.; Pawar, B. Crop Prediction Using Artificial Neural Network and Support Vector Machine. In Data Management, Analytics and Innovation; Springer: Berlin/Heidelberg, Germany, 2020; pp. 311–324.