



OPEN Smart crop disease monitoring system in IoT using optimization enabled deep residual network

Ashish Saini¹, Nasib Singh Gill¹, Preeti Gulia¹, Anoop Kumar Tiwari², Priti Maratha^{2,7}✉ & Mohd Asif Shah^{3,4,5,6,7}✉

Abstract The Internet of Things (IoT) has recently attracted substantial interest because of its diverse applications. In the agriculture sector, automated methods for detecting plant diseases offer numerous advantages over traditional methods. In the current study, a new model is developed to categorize plant diseases within an IoT network. The IoT network is simulated for monitoring crop diseases. Routing is performed with Henry Gas Chicken Swarm Optimization (HGCSO), which is designed by integrating Henry Gas Solubility Optimization (HGSO) and Chicken Swarm Optimization (CSO). The fitness parameters of the model include delay, energy, distance, and link lifetime (LLT). At the Base Station (BS), plant disease categorization is performed by collecting plant leaf images. Preprocessing is done on the input images using median filtering. Various features, such as Histogram of Oriented Gradient (HoG), statistical features, Spider Local Image Features (SLIF), and Local Ternary Patterns (LTP) are extracted. Plant disease categorization is carried out using a Deep Residual Network (DRN), which is trained using the developed Caviar Henry Gas Chicken Swarm Optimization (CHGCSO) that combines the CAViaR model with HGCSO. Comparative results show an accuracy of 94.3%, a maximum sensitivity of 93.3%, a maximum specificity of 92%, and an F1-score of 93%, indicating that the CHGCSO-based DRN outperforms existing methods. **Graphic Abstract**

Keywords Internet of Things, Smart crop disease monitoring, Deep residual network, Spider local image features

The IoT is an emerging technology that is rapidly developing in today's world. To fully leverage IoT for specific applications, an effective sensing and monitoring model is essential. IoT enables the creation of a smarter, more interconnected world^{25,36}. IoT in farming has recently inspired various innovative designs. It enables remote access, allowing for energy savings and more efficient resource management^{20,22}. An IoT platform for disease detection can significantly boost crop production by alerting farmers early, allowing them to take action before the disease causes damage^{17,32}. The type of disease detected by an IoT device is often not immediately apparent to the farmer. Additionally, modelling plant disease infection can be time-consuming and costly. Automating plant disease detection improves accuracy while reducing costs. Integrating image processing with IoT is crucial for precise assessment and timely intervention³⁴.

The detection and classification of diseases in rice plants^{15,16} are based on digital images and have gained significant attention in the research community. The automated model can assist farmers by guiding them in selecting the right pesticides at the right time³⁷. Some researchers have focused on optimization algorithms^{19,30,38,43,44} and deep learning, particularly Convolutional Neural Networks (CNNs)⁵, to address crop disease detection. These approaches have demonstrated the ability to outperform previous methods in various automated identification tasks, such as natural language processing and visual recognition. Deep learning is a well-known technique for recognition tasks²¹. Despite the efficiency of CNNs in general object recognition, deep learning offers an effective model for identifying diseases in crops⁴⁵. Automated plant disease detection

¹Department of Computer Science & Applications, Maharshi Dayanand University, Rohtak 124001, India.

²Department of Computer Science & Information Technology, Central University of Haryana, Mahendragarh 123031, India. ³Department of Economics, Kardan University, Kabul, Afghanistan. ⁴Centre for Research Impact

& Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, 140401, Rajpura, Punjab, India. ⁵ Chitkara Centre for Research and Development, Chitkara University, Baddi, Himachal Pradesh 174103, India. ⁶Division of Research and Development, Lovely Professional University, Phagwara, Punjab, India. ⁷P. Maratha, M. A. Shah: these authors contributed equally to this work. ✉email: drpriti@cuh.ac.in;

m.asif@kardan.edu.af

technology offers advantages for plant monitoring and reduces the reliance on manual inspections^{27,40}. It also aids in early disease identification, decreases pesticide use, and supports precision agriculture^{7,26,33,35}.

The main contributions are highlighted as follows:

- a) **HGCSO for routing:** The HGCSO is utilized for routing, which is obtained by combining HGSO and CSO.
- b) **CHGCSO-based DRN:** The CHGCSO is adapted for classifying plant disease. Here, the CHGCSO is adapted for training DRN. The CHGCSO is obtained by combining HGSO and CSO and the CAViaR model. The upcoming sections are organized as follows: Section 2 provides an overview of traditional plant disease classification methods. Section 3 explains the IoT framework. Section 4 presents the CHGCSO-based DRN for classifying plant diseases. Section 5 assesses the efficiency of classical classification approaches, followed by the conclusion in Section 6.

Literature survey

In 2018, S. Ramesh and Bharghava Rajaram³¹ developed a model for earlier discovery of disease in rice crops from visual symptoms. The rice crops were utilized for the processing. It provided on-field disease discovery and prediction. However, this scheme did not perform earlier discovery of crop disease. Artzai Picon *et al.*²⁹ utilized a CNN model for classifying plant disease in 2019. Here, the information based on images were incorporated. It combined the benefits of concurrently learning while minimizing the complication of classifying the disease. However, misclassification occurs whenever the quality of the image is low. In 2019, Artzai Picon *et al.*²⁸ utilized a Deep convolution Neural Network (DCNN)-based algorithm for dealing with plant disease. Here, the discovery of their relevant disease was done including Septoria, Tan Spot and Rust. However, this method acquired huge generalization power. Monalisa Mishra *et al.*²⁴ developed an automatic technique for detecting plant disease in an IoT. The method placed the nodes with a simulation platform to capture plant leaves. The model maintained the sink node that accumulated data and assisted in monitoring IoT in 2020. Here, the Sine Cosine Algorithm-based Rider Neural Network (SCA-based RideNN) was introduced for classifying diseases in plants. However, the method consumed more time for processing. In 2020, Yushan Zhao *et al.*⁴⁵ devised a Multi-Context Fusion Network (MCFN) for classifying the plant disease. Here, it adapted with CNN for the visual features extraction from crop samples. The contextual features accumulated from sensors help to the classification of crop disease. Here, the fusion of contextual features and visual features were done by a deep fully connected network, but this model poses unbalanced data, which affects the performance. Kumari Shibani *et al.*³⁶ developed a technique, namely Random Forest and Support Vector Machine and Logistic Regression (RF+SVM+LR) for classifying plant disease in 2020. Here, the first part aimed to model a system wherein the tracking of plant needs was done with sensors. Here, the automated water controller tasks with moisture content fall below a specific threshold. However, this method suffered from high computational costs. In 2020, Juan Wen *et al.*⁴² developed Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) for classifying plant diseases. Here, the ESRGAN was utilized for recovering the super-resolution from low-resolution images. In addition, the Transfer learning was adapted for compensating training samples. It was unable to process other databases. David Argüeso *et al.*³ established a plant disease identification model using distance metric Few-Shot Learning (FSL) using Triplet loss in 2020. Here, the CNN was utilized for extracting the general plant leaf features. This approach shortened the time taken for production and minimized the cost of plant disease. However, this method generated an error value, which affected the classification performance. In 2021, Olusola Oluwakemi Abayomi-Alli *et al.*¹ devised cassava disease recognition. Here, an image colour histogram alteration scheme was implemented for the creation of synthetic images to perform data augmentation. It detected and recognized the disease in cassava leaf effectively with less images. This system failed to identify a variety of diseases in plants. Ahmad Almadhor *et al.*² devised an Artificial Intelligence (AI)-driven approach for Disease Recognition of Guava Plant in 2021. It helped the farmers to avoid the production loss due to the early and effective finding of plant disease. However, this model gained less accuracy for some specific classes of disease. In 2021, Nidhi Kundu *et al.*¹⁷ devised an Automatic and Intelligent Data Collector and Classifier. The data used in this analysis was gathered from the pearl millet farmland. It was located in ICAR, Mysore, India and the parametric and farmland data were collected. These gathered information were processed in Raspberry Pi and cloud server with the help of 'Custom-Net' approach in the disease identification of pearl millet. It was an automated scheme with low-cost and user-friendly tool for farmers. However, the safety of data was the major drawback. Zia ur Rehman *et al.*³² devised novel parallel real-time processing in 2021. Here for improving the visual image impact a hybrid contrast stretching scheme was utilized and the infected areas detection was done by MASK RCNN. At the same time, CNN was trained for extracting the features. The Ensemble Subspace Discriminant Analysis (ESDA) was utilized to obtain good accuracy. However, it was trained by a small number of ground truth images. Table 1 shows the literature survey of the previous works.

System model

IoT comprises several kinds of objects that include smart devices, and are linked amongst each other to exchange collected data over the network. Various smartest devices are devised on the basis of resources that comprise communication and processing ability for exchanging data. Figure 1 displays the IoT model in which various nodes are splitted for sending data to BS with effective paths. The smartest objects are energy-constrained, and hence gateway must be interested for supervision. Consider IoT network poses n nodes given as, $H = \{H_1, H_2, \dots, H_l, \dots, H_k\}$ that relies in coverage area D . Each IoT node k attains data based on the plant leaves. IoT tends to be imperative in enhancing production in agriculture and is utilized to monitor soil temperature. Furthermore, IoT strengthens efficiency and minimizes physical work and time and helps to make farming more effective.

Authors	Methods	Dataset	Disadvantages
Mishra <i>et al.</i> ²⁴	An automatic technique for detecting plant disease in IoT platform.	Plant village dataset	The method consumed more time for processing.
Yushan Zhao <i>et al.</i> ⁴⁵	MCFN for classifying the plant disease.	50,000 in-field crop disease samples	This method poses unbalanced data, which affects the performance.
Kumari Shibani <i>et al.</i> ³⁶	RF+SVM+LR for classifying the plant disease.	Textile dataset	This method suffered from high computational costs.
ArtzaiPicon <i>et al.</i> ²⁹	CNN model	Field conditions were used to validate across five crops, 17 diseases, and a dataset of 121,955 images.	The misclassification occurs whenever the quality of the image is low.
S. Ramesh and Bharghava Rajaram ³¹	A model from visual symptoms.	For consideration, two samples were included in the work: the database image and the acquired image.	This technique did not perform earlier discovery of crop disease.
Juan Wen <i>et al.</i> ⁴²	ESRGAN for classifying the plant disease.	Plant Village dataset -	This technique was unable to handle other databases.
David Argüeso <i>et al.</i> ³	Distance metric FSL using Triplet loss for plant disease detection.	Plant Village dataset	This method generated an error value, which affects the classification performance.
ArtzaiPiconet <i>et al.</i> ²⁸	DCNN-based algorithm for dealing with plant disease.	Wheat 2015 (W-2015) Wheat 2014 (W-2014) Wheat 2016 (W-2016)	This method acquired huge generalization power.
Olusola Oluwakemi Abayomi-Alli <i>et al.</i> ¹	Cassava disease recognition.	Cassava leaf disease dataset	This system failed to identify a variety of plant diseases.
Ahmad Almadhor <i>et al.</i> ²	AI-Driven Framework	Image dataset of guava leaves and fruit.	This model gained less accuracy for some specific classes of disease
Nidhi Kundu <i>et al.</i> ¹⁷	Intelligent and Automatic Data Classifier and Collector	The dataset includes images of pearl millet leaves, with 1,964 infected by blast and 1,336 infected by rust	The safety of data was the major drawback
Zia ur Rehman <i>et al.</i> ³²	MASK RCNN	Plant Village dataset	It was trained by a small number of ground truth images.

Table 1. Literature Review.

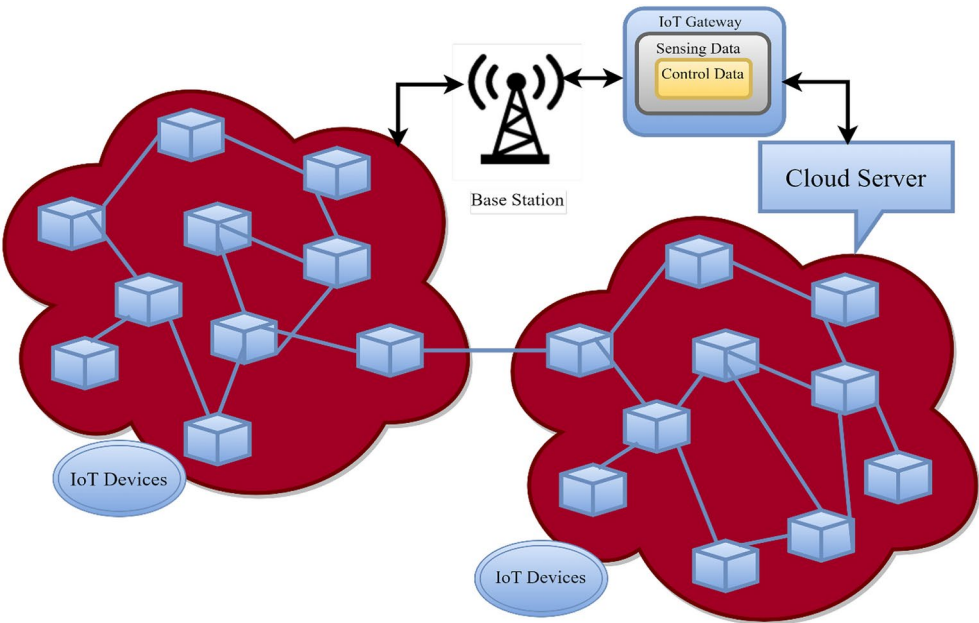


Fig. 1. IoT model.

Energy model

The energy model⁹ in IoT is examined using data transmission using various nodes. Whenever a normal sensor node sends d bits data then the energy of the IoT is expressed as,

$$E_a(d) = E_b(d) + E_c(d, f) \tag{1}$$

where, $E_a(d)$ expresses transmitted energy, E_c symbolizes consumed energy when 1-bit data send by the the node, E_b signifies utilized energy whenever a node transmits and receives 1-bit data, f symbolizes communication area, and d signifies data bits.

$$E_a(d) = \begin{cases} d \times E_b + d \times E_g \times f^2; & \text{if } f \leq f_0 \\ d \times E_b + d \times E_g \times f^4; & \text{if } f > f_0 \end{cases} \quad (2)$$

where, E_b symbolizes consumed energy in the free space approach, E_g signifies energy consumed in the multipath fading scheme.

$$f_0 = \sqrt{\frac{E_c}{E_b}} \quad (3)$$

The energy helps to receive d bits of data and formulated as,

$$E_m(d) = d \times E_b \quad (4)$$

where, $E_m(d)$ express received energy

$$\epsilon(d) = d \times E_m \quad (5)$$

where, E_m denotes the energy utilization when sending and receiving 1-bit data, $\epsilon(d)$ implies energy utilization of sensor nodes, d denotes the data bits.

Link lifetime model

LLT⁴ signifies network lifetime, which associates two sensor nodes in IoT. Consider C_1 and C_2 signifies two mobile nodes located at A_{C1}, B_{C1} and A_{C2}, B_{C2} for which the LLT is formulated as,

$$LLT = \frac{-(rs + pq) + \sqrt{(r^2 + q^2)l^2 - (rq - ps)^2}}{(r^2 + q^2)} \quad (6)$$

such that,

$$\begin{aligned} r &= G_{C1} \cos \theta_{C1} - G_{C2} \cos \theta_{C2} \\ s &= A_{C1} - A_{C2} \\ p &= G_{C1} \sin \theta_{C1} - G_{C2} \sin \theta_{C2} \\ q &= B_{C1} - B_{C2} \end{aligned} \quad (7)$$

where, l signifies range of transmission, G_{C1} symbolizes speed of mobility in the sensor node C_1 , G_{C2} signifies the sensor node's mobility speed C_2 , A_{C2}, B_{C2} signifies node coordinate, C_2 , A_{C1}, B_{C1} symbolize node coordinate C_1 , θ_{C1} refers motion direction of node C_1 , and θ_{C2} signifies direction of motion in node C_2 .

CHGCSO-based DRN for plant disease classification

The goal is to devise a CHGCSO-based DRN for classifying plant disease in IoT. At first, the IoT nodes are simulated HGCSO performs routing based on fitness function. The HGCSO is devised by combining HGSO¹³ and CSO²³. After the routing, the rice plant disease classification is performed at the BS. In the BS, the input image is fed into the preprocessing. Then, the features like HoG¹⁸, statistical features, SLIF¹¹ and LTP³⁹ are extracted. Finally, the plant disease classification is done with DRN⁸, which will be trained using CHGCSO. The CHGCSO is designed by combining the CAViaR model¹⁰, HGSO and CSO. Figure 2 depicts the architecture of the HGCSO-based DRN.

Proposed HGCSO-based routing

The routing is carried out with HGCSO using fitness function.

Solution encrypting

Determining the best solution to optimization problems requires solution encoding. Here, the HGCSO is adapted for choosing a suitable solution wherein the optimum node is chosen for routing. On the basis of fitness, the solution set attains optimum routing. This routing is performed using fitness function. Figure 3 presents solution modelling for optimum routing with HGCSO. Here, e signifies the node index.

Fitness function

The evaluation of fitness for optimized routing comprises certain parameters that include energy, distance, LLT and delay.

$$Fitness = \frac{1}{4} [E_i + F_{i,j} + (1 - L_{ij} + t)] \quad (8)$$

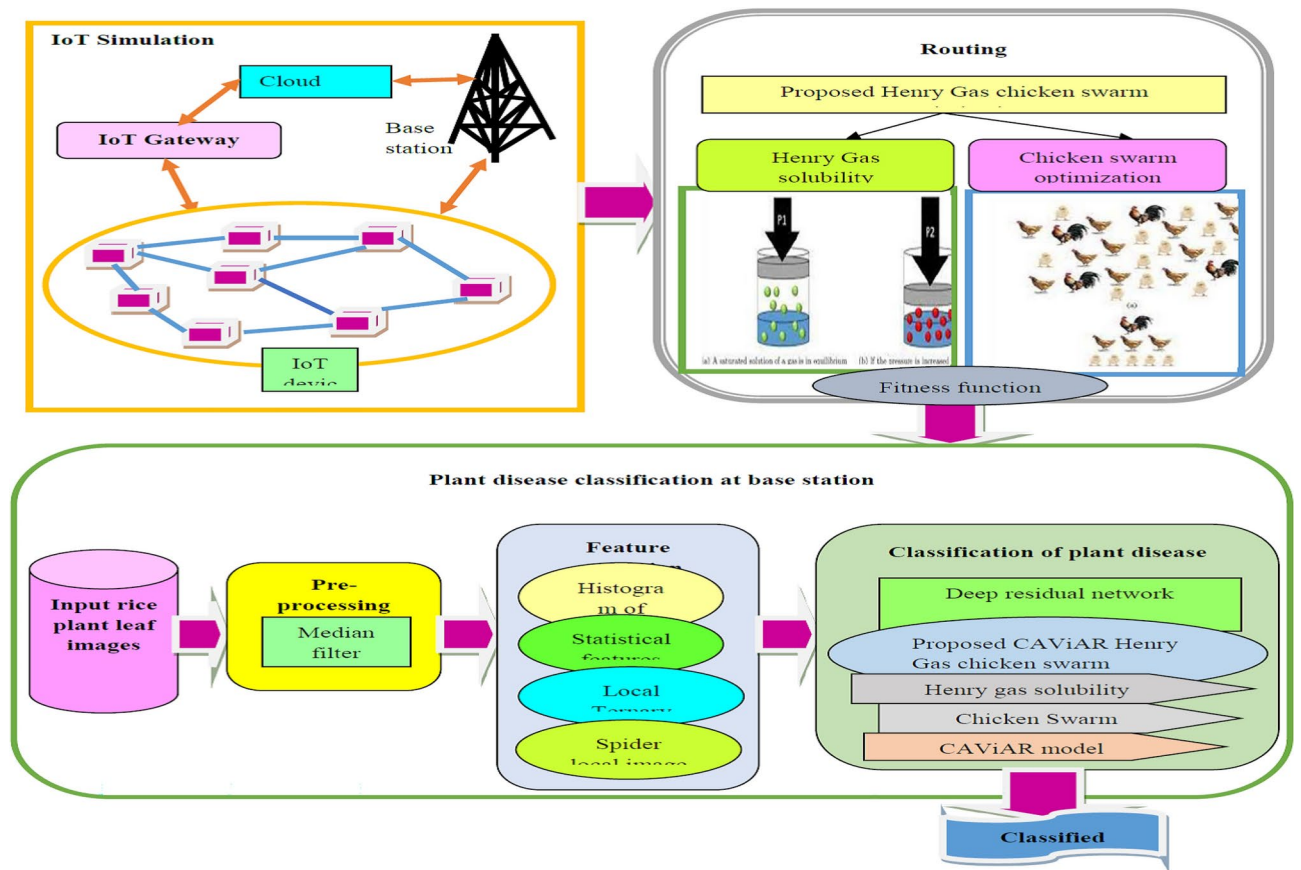


Fig. 2. Block diagram of HGCSO-based Deep Residual Network for Plant Disease Classification in IoT.

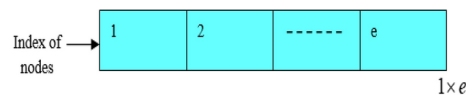


Fig. 3. Solution representation for routing with HGCSO.

where, E_i signifies energy consumption in i^{th} node, $F_{i,j}$ is Euclidean distance between i^{th} and j^{th} node, L_{ij} symbolizes LLT, and t symbolizes delay.

Here, the distance is given as,

$$F_{i,j} = ||n_i - n_j|| \quad (9)$$

where, n_i and n_j signifies distance between i^{th} and j^{th} node.

The delay is formulated as,

$$t = \frac{l_p}{h} \quad (10)$$

where, l symbolizes the entire nodes in a path, and h signifies the entire nodes in a network. The LLT is described in section 3.2.

Proposed HGCSO algorithm

The routing is done using the proposed HGCSO, and is produced by integrating HGSO¹³ and CSO²³. Algorithm 1 displays the pseudo-code of HGCSO and the flowchart of the HGCSO is given in Figure 4. The steps of HGCSO are specified below:

Step 1: Initiation The count and spot of gases are initialized as,

$$H_f(e+1) = H(H_{min_{max}})_{min} \quad (11)$$

where, u signifies random numeral in $[0, 1]$, H_{max} and H_{min} symbolize bounds of the problem, and e refers iteration time.

Step 2: Discovery of Fitness The fitness is already described in section 4.1.2.

Step 3: Clustering Divided into equal parts by gas type, each cluster of agents consists of the same gases and shares the same Henry's constant value.

Step 4: Evaluation Evaluation in each cluster seeks out the gas that reaches the highest equilibrium state, after which a ranking process identifies the optimal gas from the entire swarm.

Step 5: Update Henry coefficient The Henry coefficient of cluster w at redundancy $u+1$ is given as,

$$\lambda_w(u+1) = \lambda_w(u) + \exp(-\vartheta_w \times (1/\alpha(u) - 1/\alpha^\theta)) \quad (12)$$

where, $\lambda_w(u)$ signifies Henry coefficient for cluster w at iteration u , α symbolize temperature, α^θ refers constant in such a way that $\alpha^\theta = 298.15$.

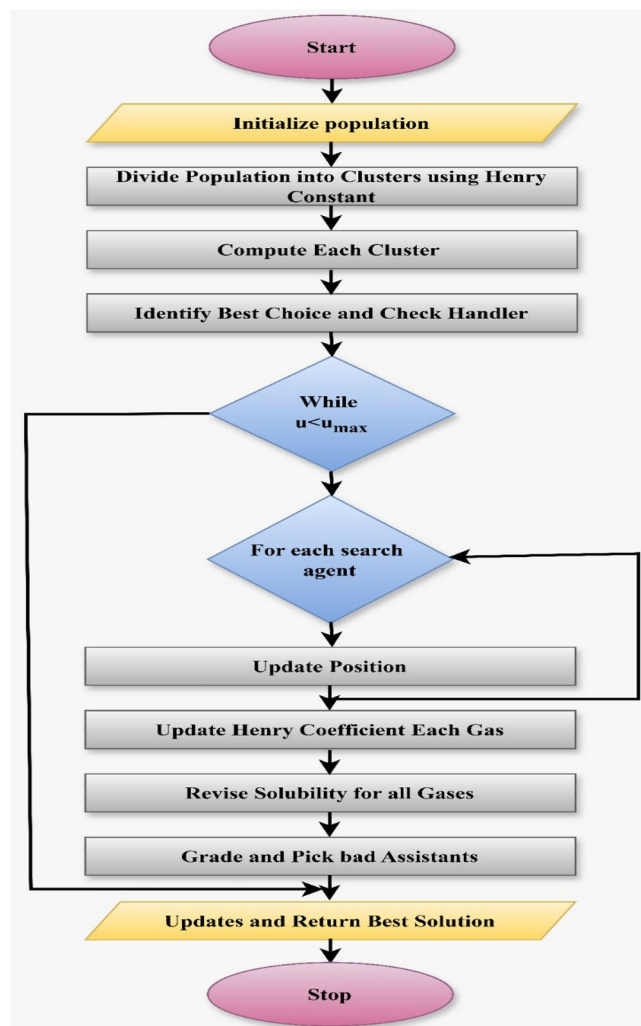


Fig. 4. Flowchart of HGCSO.

Step 6: Update Solubility

The solubility is specified as,

$$N_{o,w}(u) = x \times \lambda_w(u+1) \times K_{o,w}(u) \quad (13)$$

where, $K_{o,w}(u)$ signifies biased force on gas o in cluster w , and x is constant, and $\lambda_w(u+1)$ refers Henry coefficient to the cluster w at redundancy $u+1$.

where, $K_{o,w}(u)$ denotes limited pressure on gas o in cluster w , and x is constant, and $\lambda_w(u+1)$ denotes to the cluster w Henry coefficient at iteration $u+1$.

Step 7: Update Position

The HGSO is adapted to strengthen algorithm performance and resolve optimization issues. According to HGSO¹³, the update equation is,

$$M_{o,w}(u+1) = M_{o,w}(u) + X \times Y \times \beta (M_{o,best}(u) - M_{o,w}(u)) + X \times Y \times \gamma (R_{o,w}(u) \times M_{best}(u) - M_{ow}(u)) \quad (14)$$

where, $M_{o,w}(u)$ refers position of gas o in cluster w , Y refers constant, u indicates iteration time, $M_{o,best}(u)$ indicates best gas o in cluster w , $M_{best}(u)$ signifies optimum gas in swarm, β implies the gas capability o , and γ denotes influence in gas o by other gases in cluster w , and X denotes flag that alters the direction of the search agent and $R_{o,w}(u)$ refers solubility of gas o in cluster w .

Here, the gas o ability is,

$$\beta = \ell \times \exp\left(-\frac{X_{best}(u) + \mu}{X_{o,w}(u) + \mu}\right) \quad (15)$$

where, $X_{best}(u)$ signifies fitness of best gas in the whole system, and μ is constant where $\mu = 0.05$.

Here, the temperature is given as,

$$\alpha(u) = \exp(-u/u_{max}) \quad (16)$$

where, u_{max} refers total number of iterations.

The aforementioned equation can be rewritten as,

$$M_{o,w}(u+1) = M_{o,w}(u) (1 - XY\beta - XY\gamma) + XY\beta M_{o,best}(u) + XY\gamma R_{o,w}(u) \times M_{best}(u) \quad (17)$$

According to CSO²³, the group-mate roosters are followed for searching food. In addition, they also arbitrary steal the good food discovered by other chickens and this process is given by,

$$M_{o,w}(u+1) = M_{o,w}(u) + K_1 * Rand * (M_{r_1,w}(u) - M_{o,w}(u)) + K_2 * Rand * (M_{r_2,w}(u) - M_{o,w}(u)) \quad (18)$$

where, $Rand$ signifies a uniform arbitrary number amongst 0 to 1, r_1 refers rooster index, r_2 symbolizes chicken index, $M_{o,w}(u)$ indicates current position of hen, K_1 and K_2 represent the random numbers, $M_{r_1,w}(u)$ represent the position of rooster and $M_{r_2,w}(u)$ indicate the position of chicken.

$$\begin{aligned} M_{o,w}(u+1) &= M_{o,w}(u) + K_1 * Rand * M_{r_1,w}(u) - K_1 * Rand * M_{o,w}(u) + K_2 \\ &\quad * Rand * M_{r_2,w}(u) - K_2 * Rand * M_{o,w}(u) \\ M_{o,w}(u+1) &= M_{o,w}(u) (1 - K_1 * Rand - K_2 * Rand) + K_1 * Rand * M_{r_1,w}(u) + K_2 * Rand * M_{r_2,w}(u) \quad (19) \\ M_{o,w}(u) &= \frac{M_{o,w}(u+1) - K_1 * Rand * M_{r_1,w}(u) - K_2 * Rand * M_{r_2,w}(u)}{1 - K_1 * Rand - K_2 * Rand} \end{aligned}$$

Substituting equation (19) in equation (17),

$$M_{o,w}(u+1) = \frac{-[K_1 Rand M_{r_1,w}(u) + K_2 Rand M_{r_2,w}(u)] [1 - XY\beta - XY\gamma] + XY [\beta M_{o,best}(u) + \gamma R_{o,w}(u) M_{best}(u)] [1 - K_1 Rand - K_2 Rand]}{XY(\beta + \gamma) - Rand(K_1 + K_2)} \quad (20)$$

The final update of HGCSO is shown as,

$$M_{o,w}(u+1) = \frac{XY [\beta M_{o,best}(u) + \gamma R_{o,w}(u) M_{best}(u)] [1 - K_1 Rand - K_2 Rand] - [K_1 Rand M_{r_1,w}(u) + K_2 Rand M_{r_2,w}(u)] [1 - XY\beta - XY\gamma]}{XY(\beta + \gamma) - Rand(K_1 + K_2)} \quad (21)$$

Step 8: Local optima of escape:

Here, the step is used to escape from local optima. The rank and choose count of worst agents as,

$$\ell_o = \ell \times (\text{rand}(\omega_2 - \omega_1) + \omega_1), \omega_1 = 0.1 \text{ and } \omega_2 = 0.2 \quad (22)$$

where, ℓ symbolize count of search agents.

Step 9: Update worst search agents position

$$\rho_{(o,w)} = \rho_{\min(o,w)} + u \times (\rho_{\max(o,w)} - \rho_{\min(o,w)}) \quad (23)$$

where, $\rho_{(o,w)}$ refers position of gas oin cluster w , e signifies random number, and $\rho_{\min}$ and $\rho_{\max}$ are limits of difficulty.

Step 10: Fitness re-computation: The fitness is re-evaluated wherein the solution linked to the highest fitness is employed in routing.

Step 11: Terminate: The optimum weights are generated in a repeated manner until the highest iteration is accomplished.

Input: Search agent population M

Output: Best search agent M_{best}

1. Begin:

2. Initiation of different algorithmic limits
3. Dividing those populace towards clusters using Henry constant λ
4. Compute each cluster w
5. Acquire best choice gas $M_{o,best}$ and correct check handler M_{best}
6. **While** $u < u_{\max}$
7. **For search agent do**
8. Update position using equation

$$M_{o,w}(u+1) = \frac{XY [\beta M_{o,best}(u) + \gamma R_{o,w}(u) M_{best}(u)] [1 - K_1 \text{Rand} - K_2 \text{Rand}] - [K_1 \text{Rand} M_{r_1,w}(u) + K_2 \text{Rand} M_{r_2,w}(u)] [1 - XY\beta - XY\gamma]}{XY(\beta + \gamma) - \text{Rand}(K_1 + K_2)}$$

9. End for

10. Update Henry coefficient of each gas using equation

$$\lambda_w(u+1) = \lambda_w(u) + \exp(-\vartheta_w \times (1/\alpha(u) - 1/\alpha^\theta))$$

11. Revise solubility from all gas utilizing equation

$$N_{o,w}(u) = x \times \lambda_w(u+1) \times K_{o,w}(u)$$

12. Grade and pick bad assistants

13. Update best gas $M_{o,best}$ and best search agent M_{best}

14. End while

15. $u = u + 1$

16. **Return** M_{best}

17. End**Algorithm 1.** Pseudo code HGCSO**Plant disease classification using proposed CHGCSO-based DRN**

The classification of plant disease is done with CHGCSO-based DRN. Consider a dataset J having i rice plant images, and is expressed as,

$$J = \{I_1, I_2, \dots, I_\ell, \dots, I_i\} \quad (24)$$

where, I_ℓ signifies ℓ^{th} input image, and i symbolize total images.

Pre-Processing using median filter

Here, the group of input images J is adapted as input for the median filter. Median filter (<https://towardsdatascience.com/image-processing-class-egbe443-4-filters-aa1037676130>) is a smoothing method that reduces the edge blurring in which the notion is to replace the current point of the image with median. Furthermore, the median filter assists to discard the impulse noise. It is represented as,

$$I'(d, e) \leftarrow \text{median} \{I(d + k, e + j) | (k, j) \in K\} \quad (25)$$

where, $I(d + k, e + j)$ symbolize corresponding image element, (k, j) signifies position of image, and K refers moving region. The pre-processed output is expressed as P .

Extraction of features

Once the pre-processing is done, the extraction of features is done for effective plant disease classification. P is adapted as input for extracting features. Here, the features, like HoG, LTP, SLIF, and statistical features are adapted for further analysis. The feature vector is expressed as,

$$C = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8\} \quad (26)$$

where, C_1 symbolizes HoG feature, C_2 represents a mean feature, C_3 signifies variance feature, C_4 refers standard deviation feature, C_5 is skewness, C_6 is kurtosis feature, C_7 is SLIF, C_8 is LTP feature.

Classification of plants using proposed CHGCSO-based DRN

The classification of plant disease is done with CHGCSO-based DRN.

a) Training of DRN with CHGCSO algorithm

The CAViaR¹⁰ model indicates the evolution of quantile in contrast to time considering an autoregressive procedure, and evaluates the attributes with respect to regression quantiles. This statistical method helps compute the potential loss over a certain time period, based on historical records. The HGCSO is formulated by combining HGSO and CSO, leveraging the benefits of both algorithms.

Step 1: Initialization: The solution initialization is formulated as,

$$M = \{M_1, M_2, \dots, M_\mu, \dots, M_\eta\}; 1 \leq \mu \leq \eta \quad (27)$$

where, η expresses total solution and T_μ symbolizes μ^{th} solution.

Step 2 Discovery of Error: The optimal solution is found using the error function, which is formulated as a minimization problem, as shown:

$$MS_{err} = \frac{1}{g} \sum_{h=1}^g [K_h - O]^2 \quad (28)$$

where, K_h signifies expected output, and O denotes output of classifier, g symbolizes entire data samples, such that $1 < h \leq g$.

Step 3: Find update solutions:

To generate global optimum solution, the proposed HGCSO is adapted, which is given as,

$$M_{o,w}(u+1) = \frac{XY [\beta M_{o,best}(u) + \gamma R_{o,w}(u) M_{best}(u)] [1 - K_1 Rand - K_2 Rand] - [K_1 Rand M_{r1,w}(u) + K_2 Rand M_{r2,w}(u)] [1 - XY\beta - XY\gamma]}{XY(\beta + \gamma) - Rand(K_1 + K_2)} \quad (29)$$

Subtract $M_{o,w}(u)$ on both sides,

$$M_{o,w}(u+1) - M_{o,w}(u) = \frac{XY [\beta M_{o,best}(u) + \gamma R_{o,w}(u) M_{best}(u)] [1 - K_1 Rand - K_2 Rand] - [K_1 Rand M_{r1,w}(u) + K_2 Rand M_{r2,w}(u)] [1 - XY\beta - XY\gamma]}{XY(\beta + \gamma) - Rand(K_1 + K_2)} - M_{o,w}(u) \quad (30)$$

As per CaViaR¹⁰, the update equation is given by the current and past observation and is expressed as,

$$M_{o,w}(u) = \delta_0 + \sum_{o=1}^x \delta_1 * M_{o,w}(u-1) + \sum_{w=1}^p \delta_w f(M_{o,w}(u-w)) \quad (31)$$

where, δ represents a vector of unknown parameter, $M_{o,w}(u)$ represents current autoregressive quantile, and $M_{o,w}(u-1)$ is the previous autoregressive quantile, $f[\cdot]$ represents fitness of solution, and w is constant.

Assume $\chi = \rho = 2$,

$$M_{o,w}(u) = \delta_0 + \delta_1 M_{o,w}(u-1) + \delta_2 M_{o,w}(u-1) + \delta_1 f(M_{o,w}(u-1)) + \delta_2 f(M_{o,w}(u-2)) \quad (32)$$

$$M_{o,w}(u) = \delta_0 + \delta_1 (M_{o,w}(u-1) + f(M_{o,w}(u-1))) + \delta_2 (M_{o,w}(u-1) + f(M_{o,w}(u-2))) \quad (33)$$

After substituting equation (33) in equation (30) in RHS, the final update of CHGCSO is given as,

$$M_{o,w}(u+1) - M_{o,w}(u) = \frac{A}{B} \quad (34)$$

where,

$$A = XY [\beta M_{o,best}(u) + \gamma R_{o,w}(u) M_{best}(u)] [1 - K_1 Rand - K_2 Rand] - [K_1 Rand M_{r_1,w}(u) + K_2 Rand M_{r_2,w}(u)] [1 - XY\beta - XY\gamma] \quad (35)$$

$$B = XY(\beta + \gamma) - Rand(K_1 + K_2) - \delta_0 - \delta_1 (M_{o,w}(u-1) + f(M_{o,w}(u-1))) - \delta_2 (M_{o,w}(u-1) + f(M_{o,w}(u-2))) \quad (36)$$

where *Rand* signifies an arbitrary number between 0 to 1, K_1 and K_2 represent the random numbers, Y refers constant, u indicates iteration time, $M_{o,best}(u)$ indicates best gas o in cluster w , $M_{r_1,w}(u)$ represent the position of the rooster and $M_{r_2,w}(u)$ indicate the position of chicken, $M_{best}(u)$ denotes optimum gas in swarm, β implies the capability of gas o , and γ denote impact of other gases on gas o in cluster w , and X denote flag that alters search agent direction and $R_{o,w}(u)$ refers solubility of gas o in cluster w , δ implies an unknown parameter vector, $M_{o,w}(u)$ denotes present autoregressive quantile, and $M_{o,w}(u-1)$ is the preceding autoregressive quantile, $f[\cdot]$ represents solution fitness.

Step 4 Evaluation of Best Solution: After recomputing the error for each solution, the one showing the least error is selected as the optimal solution.

Step 5 Termination: The procedure is repeated until the optimal disease detection solution is obtained, and the process is terminated after reaching the maximal iteration u_{max} .

Results and discussion

The efficiency of CHGCSO+DRN is evaluated by varying training data using 100 and 200 nodes.

Experimental setup

The CHGCSO+DRN are implemented on Matlab and Windows 10 OS having 2GB RAM and Intel Core processor.

Dataset

The rice leaf diseases dataset (<https://archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases>) comprises three classes of diseases that include Brown spot, bacterial leaf blight, and Leaf smut wherein each class contains 40 images. The dataset is multivariate, featuring integer attributes, with 120 instances and a total of 57,566 web hits. The dataset was created by manually dividing infected leaves into distinct classes based on disease type.

Experimental outcomes

Figure 5 displays the experimental outcomes of CHGCSO+DRN. Here, the input rice leaf image is revealed in Figure 5a). The pre-processed image is revealed in Figure 5b). The HoG feature extracted image is depicted in Figure 5c). The LTP feature extracted image is displayed in Figure 5d).

Evaluation measures

Sensitivity, energy, accuracy, F1-score, specificity and ROC are the considered evaluation metrics in this model.

Comparative assessment

The techniques taken for the assessment include SCA based RideNN²⁴, RF+SVM+LR³⁶, DCNN²⁸, Residual network (ResNet)¹⁴, MCFN⁴⁵, ESRGAN⁴², and CHGCSO+DRN.

a) Evaluation utilizing 100 nodes

Figure 6 exposes comparative analysis with 100 nodes. The accuracy results are shown in Figure 6(a). For 90% of data, the testing accuracy measured by SCA based RideNN, RF+SVM+LR, DCNN, ResNet, MCFN, ESRGAN, and CHGCSO+DRN are 0.784, 0.767, 0.776, 0.826, 0.784, 0.767, and 0.868. The sensitivity result is presented in Figure 6(b). For 90% of training data, the sensitivity analyzed by SCA based RideNN, RF+SVM+LR, DCNN, ResNet, MCFN, ESRGAN, and CHGCSO+DRN are 0.790, 0.775, 0.735, 0.769, 0.741, 0.756, and 0.846. The specificity result is shown in Figure 6(c). For 90% of training data, the specificity analyzed by SCA based RideNN, RF+SVM+LR, DCNN, ResNet, MCFN, ESRGAN, and CHGCSO+DRN is 0.796, 0.786, 0.814, 0.819, 0.776, 0.826, and 0.888. The review with f1-score is displayed in Figure 6(d). For 90% of training data, the f1-score evaluated by SCA based RideNN, RF+SVM+LR, DCNN, ResNet, MCFN, ESRGAN, and CHGCSO+DRN are 0.773, 0.757, 0.765, 0.815, 0.773, 0.757, and 0.855. The ROC analysis is displayed in Figure 6(e). When FPR is 10, the TPR of SCA based RideNN, RF+SVM+LR, DCNN, ResNet, MCFN, ESRGAN, and CHGCSO+DRN are 1, 1, 1, 1, 1, 1, and 1. The values of AUC are 0.5098, 0.5207, 0.5274, 0.5396, 0.541, and 0.5575.

b) Evaluation utilizing 150 nodes

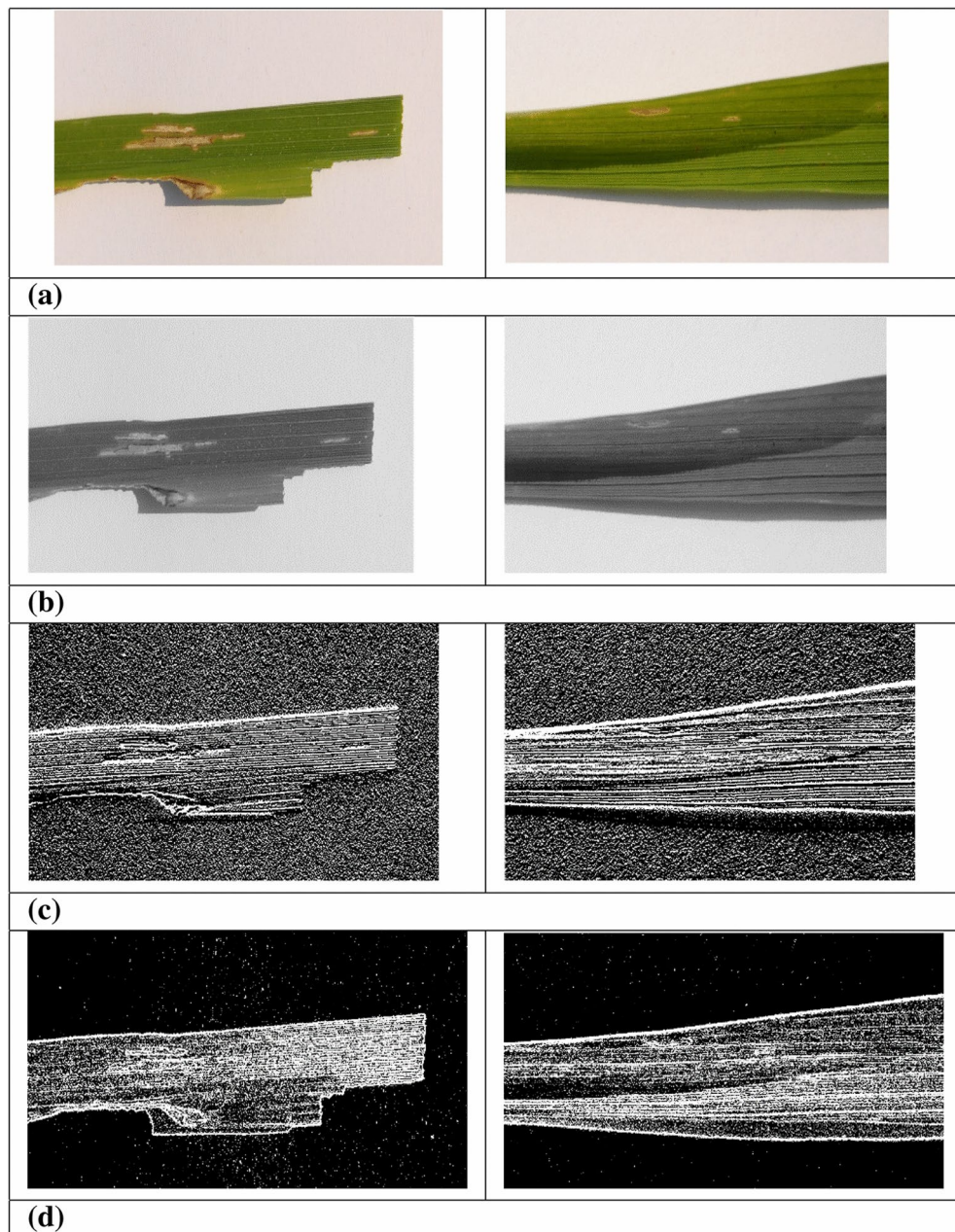


Fig. 5. Experimental outcomes of developed CHGCSO+DRN with (a) Input Image (b) Pre-processed Image (c) HoG feature Extracted Image (d) LTP Feature Extracted Image.

The estimation of techniques with 150 nodes is exposed in Figure 7. The accuracy results are provided in Figure 7(a). When training data is 50%, the accuracy measured by SCA based RideNN, RF+SVM+LR, DCNN, ResNet, MCFN, ESRGAN, are 0.598, 0.606, 0.649, 0.694, 0.774, 0.796, and the CHGCSO+DRN is 0.774. The sensitivity result is given in Figure 7(b). For 50% of training data, the sensitivity created by SCA based RideNN, RF+SVM+LR, DCNN, ResNetMCFN, ESRGAN, are 0.600, 0.612, 0.633, 0.665, 0.796, 0.815, and the CHGCSO+DRN is 0.750. The specificity result is shown in Figure 7(c). For 50% of training data, specificity obtained by SCA based RideNN, RF+SVM+LR, DCNN, ResNet, MCFN, ESRGAN, are 0.655, 0.660, 0.686, 0.704, 0.816, 0.825, whereas the CHGCSO+DRN is 0.752. The f1-score analysis is shown in Figure 7(d). For 90% of training data, the f1-score evaluated by SCA based RideNN, RF+SVM+LR, DCNN, ResNet, MCFN, ESRGAN, and CHGCSO+DRN are 0.815, 0.798, 0.806, 0.859, 0.763, 0.785, and 0.902. The ROC analysis is displayed in Figure 7(e). When FPR is 10, the TPR of SCA based RideNN, RF+SVM+LR, DCNN, ResNet, MCFN, ESRGAN, and CHGCSO+DRN are 1, 1, 1, 1, 1, and 1. The values of AUC are 0.5128, 0.5287, 0.5354, 0.5446, 0.5653, and 0.5885.

c) Evaluation utilizing 200 nodes

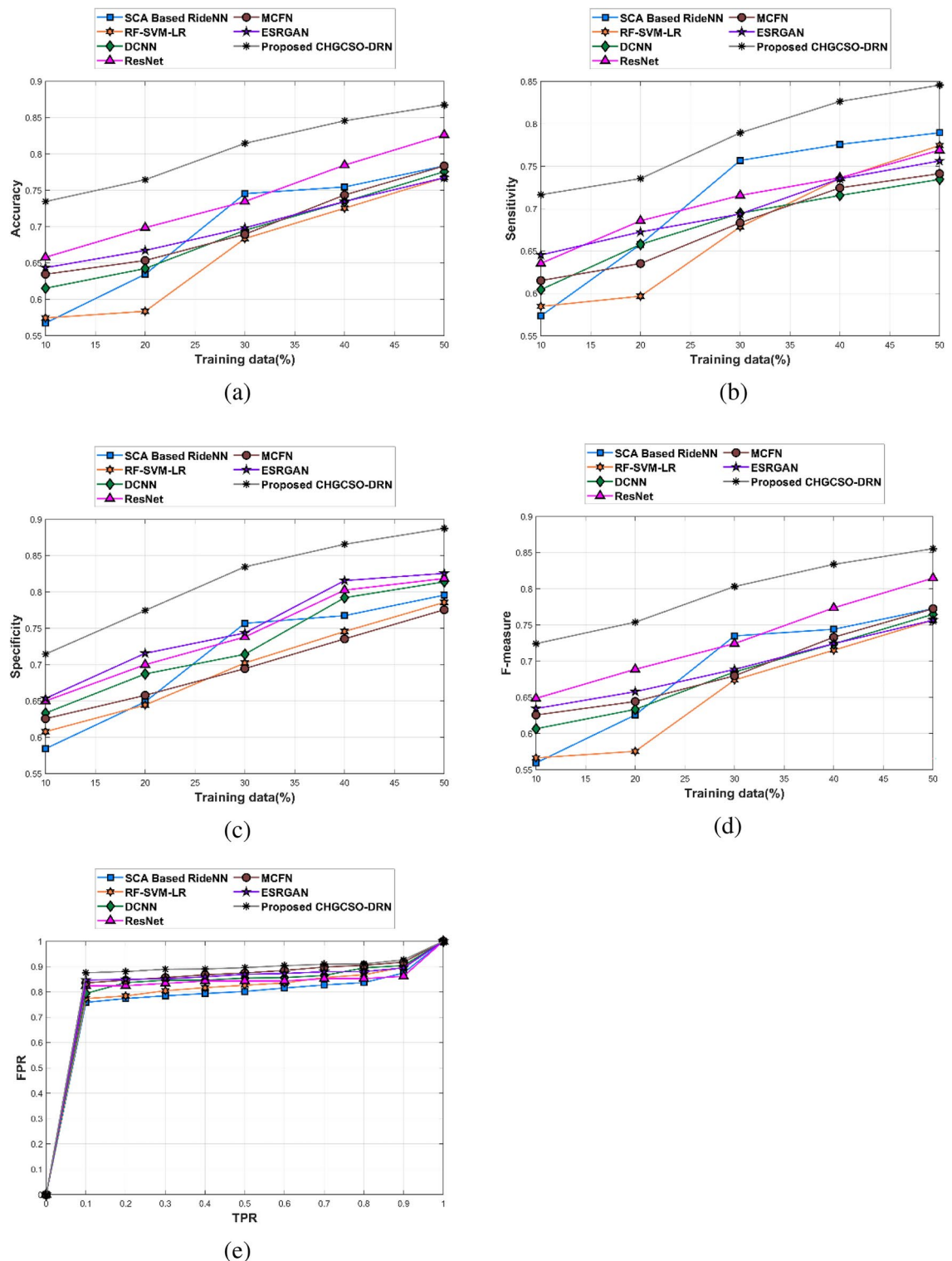


Fig. 6. Comparative of methods with 100 nodes using (a) Testing accuracy (b) Sensitivity (c) Specificity (d) F1-score (e) ROC.

Figure 8 shows estimate of methods accompanying 200 nodes. The accuracy result is revealed in Figure 8(a). For 90% of training data, the accuracy generated by SCA based RideNN is 0.933, RF+SVM+LR is 0.831, DCNN is 0.854, ResNet is 0.867, MCFN is 0.825, ESRGAN is 0.816, and CHGCSO+DRN is 0.943. The sensitivity result is conferred in Figure 8(b). For 90% of training data, the sensitivity obtained by SCA based RideNN is 0.889, RF+SVM+LR is 0.883, DCNN is 0.904, ResNet is 0.860, MCFN is 0.816, ESRGAN is 0.825, and CHGCSO+DRN is 0.933. The evaluation of specificity is presented in Figure 8(c). For 90% of training data, the specificity

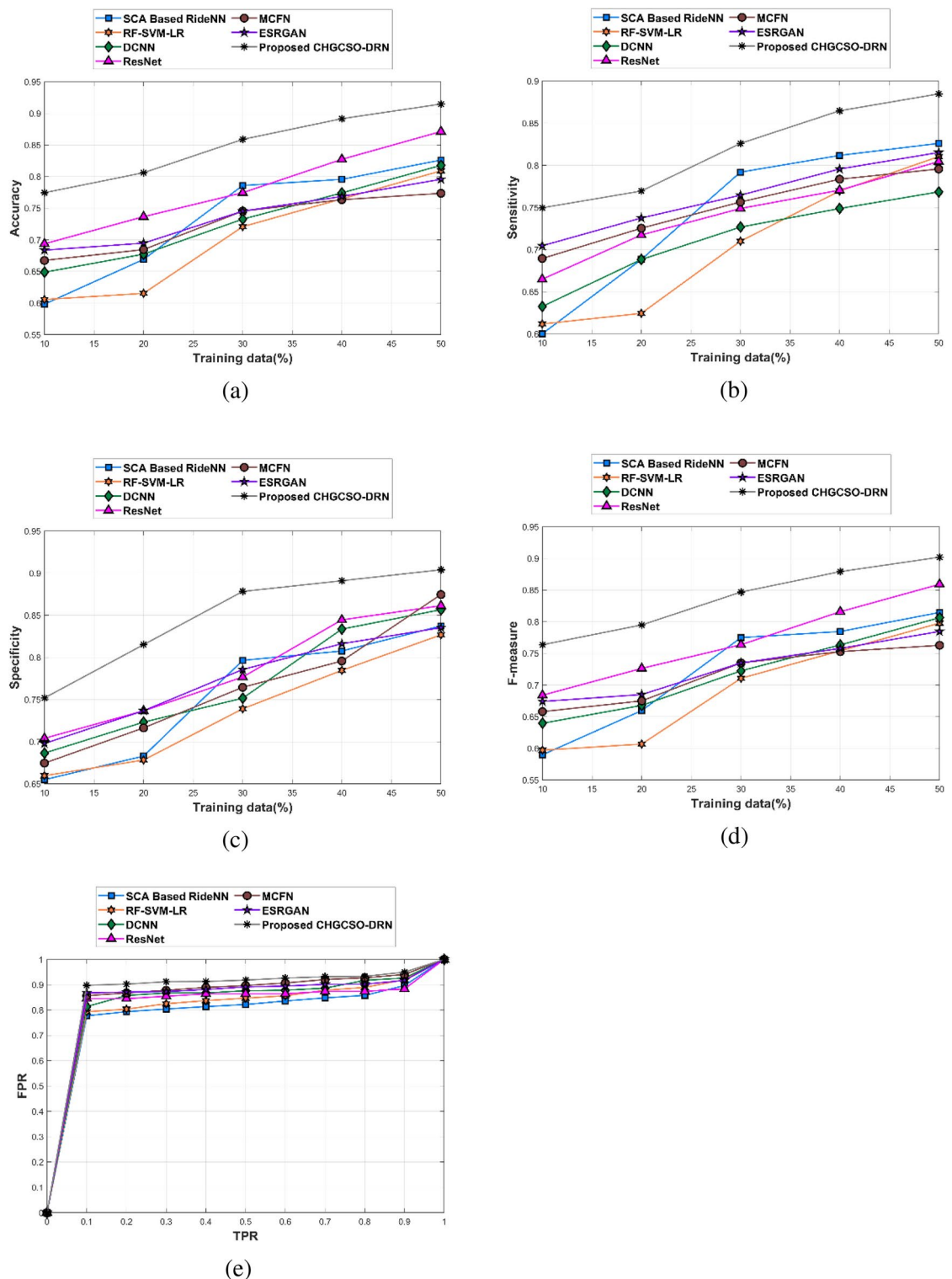


Fig. 7. Comparative with 150 nodes using (a) Accuracy (b) Sensitivity (c) Specificity (d) F1-score (e) ROC.

generated by SCA located RideNN is 0.831, RF+SVM+LR is 0.747, DCNN is 0.749, ResNet is 0.912, MCFN is 0.825, ESRGAN is 0.816, and CHGCSO+DRN is 0.920. The f1-score estimation is given in Figure 8(d). For 90% of training data, the f1-score judged by SCA located RideNN, RF+SVM+LR, DCNN, ResNet, MCFN, ESRGAN, and CHGCSO+DRN are 0.920, 0.820, 0.842, 0.855, 0.814, 0.805, and 0.930. The ROC analysis is displayed in Figure 8(e). When FPR is 10, the TPR of SCA based RideNN, RF+SVM+LR, DCNN, ResNet, MCFN, ESRGAN, and proposed CHGCSO+DRN are 1, 1, 1, 1, 1, and 1. The values of AUC are 0.5245, 0.5347, 0.5524, 0.5696, 0.5815, and 0.6035.

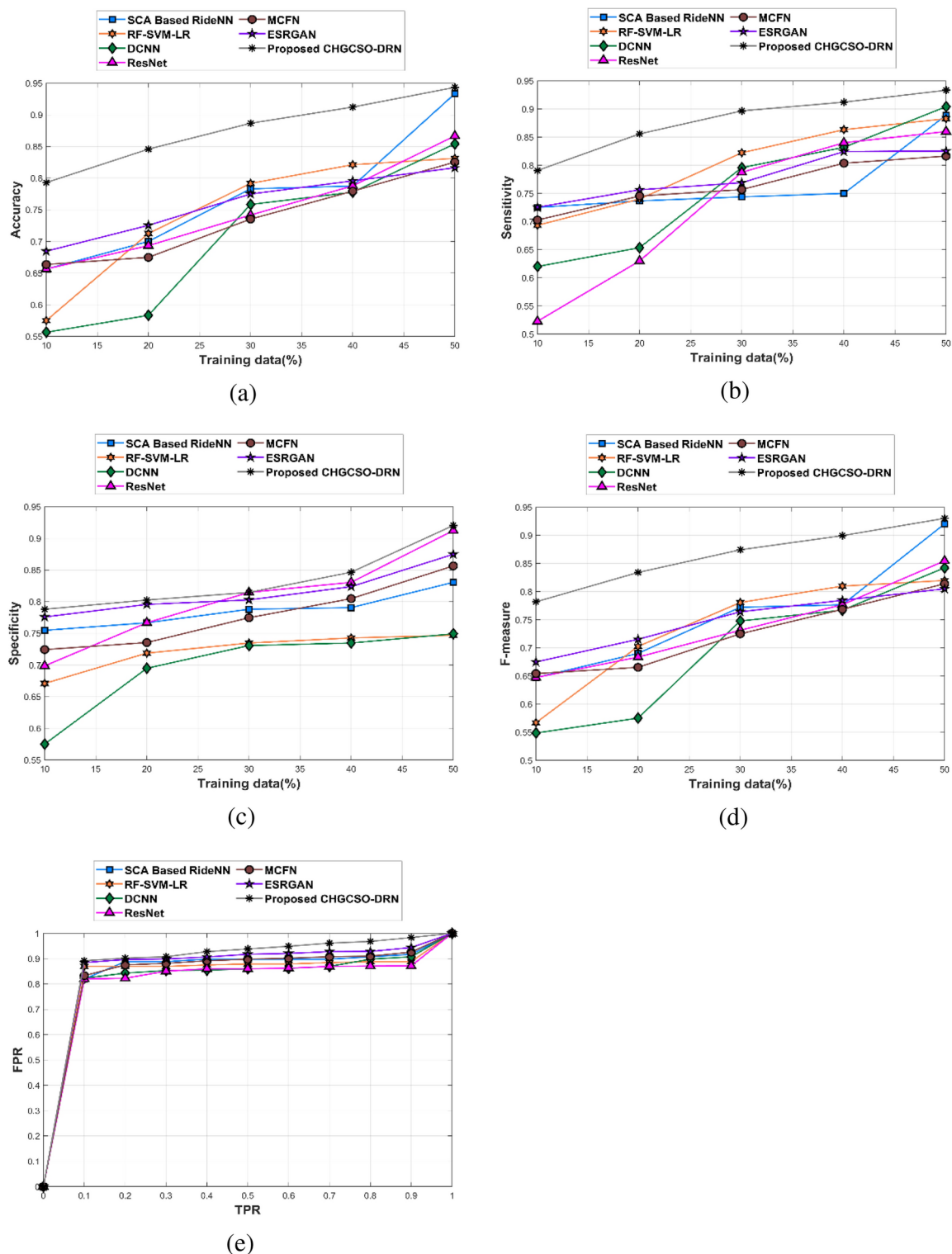


Fig. 8. Assessment with 200 nodes using (a) Accuracy (b) Sensitivity (c) Specificity (d) f1-score (e) ROC.

K-Fold cross validation

Table 2 represents the analysis based on k-fold cross validation by considering 100, 150, and 200 nodes based on all evaluation metrics. Here, the performance is increased when increasing the k-fold and the maximum results are obtained at k-fold=9.

Metrics	K-Fold/ Methods	5	6	7	8	9
	100 nodes					
Accuracy	SCA located RideNN	0.559	0.625	0.734	0.743	0.772
	RF+SVM+LR	0.566	0.575	0.673	0.714	0.756
	DCNN	0.606	0.633	0.684	0.723	0.764
	ResNet	0.648	0.688	0.724	0.773	0.814
	MCFN	0.625	0.644	0.679	0.732	0.772
	ESRGAN	0.634	0.657	0.688	0.724	0.756
	CHGCSO+DRN	0.724	0.753	0.802	0.833	0.855
Sensitivity	SCA located RideNN	0.565	0.648	0.746	0.764	0.778
	RF+SVM+LR	0.576	0.588	0.668	0.724	0.763
	DCNN	0.595	0.648	0.684	0.705	0.724
	ResNet	0.626	0.675	0.705	0.725	0.757
	MCFN	0.606	0.626	0.673	0.714	0.730
	ESRGAN	0.636	0.662	0.683	0.724	0.745
	CHGCSO+DRN	0.706	0.724	0.778	0.814	0.833
Specificity	SCA located RideNN	0.576	0.639	0.746	0.756	0.784
	RF+SVM+LR	0.599	0.635	0.692	0.734	0.774
	DCNN	0.624	0.677	0.704	0.780	0.802
	ResNet	0.640	0.690	0.727	0.790	0.806
	MCFN	0.616	0.648	0.684	0.724	0.764
	ESRGAN	0.644	0.705	0.732	0.803	0.813
	CHGCSO+DRN	0.704	0.763	0.822	0.853	0.874
F1-score	SCA located RideNN	0.551	0.616	0.724	0.733	0.761
	RF+SVM+LR	0.558	0.567	0.664	0.704	0.745
	DCNN	0.598	0.624	0.675	0.713	0.753
	ResNet	0.639	0.678	0.713	0.762	0.803
	MCFN	0.616	0.635	0.670	0.722	0.761
	ESRGAN	0.625	0.648	0.678	0.713	0.745
	CHGCSO+DRN	0.713	0.743	0.791	0.821	0.843
	150 nodes					
Accuracy	SCA located RideNN	0.589	0.659	0.774	0.784	0.814
	RF+SVM+LR	0.596	0.606	0.710	0.753	0.797
	DCNN	0.639	0.667	0.722	0.763	0.806
	ResNet	0.683	0.726	0.763	0.815	0.858
	MCFN	0.657	0.674	0.734	0.752	0.762
	ESRGAN	0.673	0.684	0.734	0.757	0.784
	CHGCSO+DRN	0.763	0.794	0.846	0.878	0.901
Sensitivity	SCA located RideNN	0.591	0.678	0.780	0.800	0.814
	RF+SVM+LR	0.603	0.615	0.699	0.758	0.798
	DCNN	0.623	0.678	0.716	0.737	0.757
	ResNet	0.655	0.707	0.738	0.759	0.792
	MCFN	0.679	0.714	0.745	0.772	0.784
	ESRGAN	0.694	0.726	0.753	0.784	0.803
	CHGCSO+DRN	0.738	0.758	0.814	0.852	0.871
Specificity	SCA located RideNN	0.645	0.673	0.785	0.795	0.825
	RF+SVM+LR	0.650	0.668	0.728	0.773	0.814
	DCNN	0.676	0.712	0.740	0.821	0.844
	ResNet	0.693	0.726	0.765	0.832	0.848
	MCFN	0.664	0.706	0.753	0.784	0.861
	ESRGAN	0.687	0.726	0.774	0.804	0.823
	CHGCSO+DRN	0.741	0.803	0.865	0.878	0.890
Continued						

Metrics	K-Fold/ Methods	5	6	7	8	9
F1-score	SCA located RideNN	0.581	0.650	0.763	0.773	0.802
	RF+SVM+LR	0.588	0.598	0.700	0.743	0.786
	DCNN	0.630	0.658	0.712	0.752	0.794
	ResNet	0.674	0.715	0.752	0.803	0.846
	MCFN	0.648	0.665	0.724	0.741	0.751
	ESRGAN	0.664	0.675	0.724	0.746	0.773
	CHGCSO+DRN	0.752	0.783	0.834	0.866	0.888
200 nodes						
Accuracy	SCA located RideNN	0.646	0.689	0.771	0.776	0.919
	RF+SVM+LR	0.566	0.702	0.780	0.809	0.819
	DCNN	0.548	0.575	0.747	0.766	0.841
	ResNet	0.647	0.683	0.731	0.776	0.854
	MCFN	0.654	0.665	0.724	0.768	0.813
	ESRGAN	0.674	0.714	0.763	0.784	0.804
	CHGCSO+DRN	0.781	0.833	0.873	0.899	0.929
Sensitivity	SCA located RideNN	0.714	0.726	0.733	0.739	0.876
	RF+SVM+LR	0.683	0.729	0.810	0.850	0.870
	DCNN	0.611	0.644	0.784	0.819	0.891
	ResNet	0.515	0.621	0.776	0.827	0.847
	MCFN	0.692	0.734	0.745	0.792	0.804
	ESRGAN	0.714	0.745	0.757	0.812	0.813
	CHGCSO+DRN	0.779	0.843	0.883	0.899	0.919
Specificity	SCA located RideNN	0.743	0.755	0.776	0.779	0.818
	RF+SVM+LR	0.661	0.708	0.724	0.732	0.736
	DCNN	0.566	0.684	0.720	0.724	0.738
	ResNet	0.688	0.755	0.802	0.818	0.899
	MCFN	0.713	0.724	0.763	0.793	0.844
	ESRGAN	0.764	0.784	0.791	0.811	0.862
	CHGCSO+DRN	0.776	0.791	0.802	0.834	0.907
F1-score	SCA located RideNN	0.647	0.690	0.772	0.776	0.920
	RF+SVM+LR	0.567	0.703	0.781	0.810	0.820
	DCNN	0.549	0.575	0.748	0.767	0.842
	ResNet	0.647	0.684	0.731	0.777	0.855
	MCFN	0.654	0.665	0.725	0.768	0.814
	ESRGAN	0.675	0.715	0.764	0.785	0.805
	CHGCSO+DRN	0.782	0.834	0.874	0.899	0.930

Table 2. Analysis on K-Fold cross validation.

Analysis based on routing

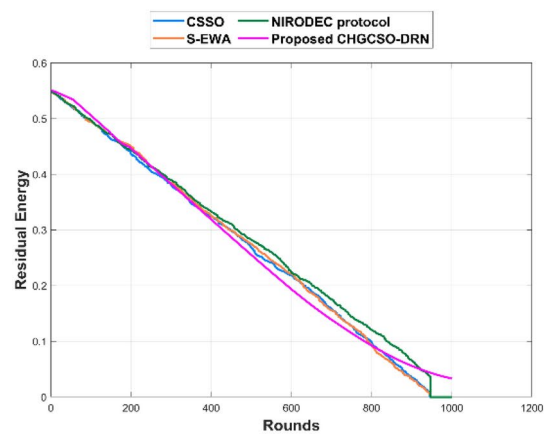
Figure 9 shows the analysis based on routing. In this research, the routing performance of the devised model is compared with other routing techniques, such as Competitive Shuffled Shepherd Optimization (CSSO)¹², Sunflower EarthWorm (S-EWA)⁴¹, and Novel Intelligent Robust Optimized Dynamic and Efficient Cluster (NIRODEC) protocol⁶. The residual energy metric is considered for the evaluation and Figures 9a), 9b), and 9c) denote the assessment based on 100, 150, and 200 nodes respectively. When the number of rounds 1000, the residual energy obtained by the CSSO, S-EWA, NIRODEC protocol, and the devised scheme is 0, 0, 0, and 0.033 for 100 nodes, 0.001, 0.001, 0.001, and 0.046 for 150 nodes, and 0.006, 0.009, 0.011, and 0.059 for 200 nodes, respectively.

Ablation study

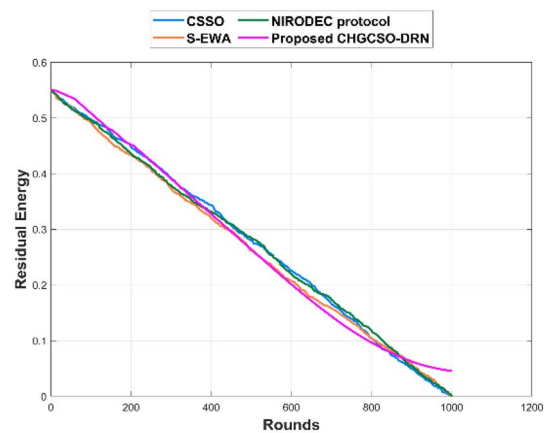
Table 3 shows the ablation study of the CHGCSO+DRN. Here, the performance of the CHGCSO+DRN is compared with each individual component and the performance is high for CHGCSO+DRN for all evaluation metrics. The evaluation is done by considering different nodes.

Comparative discussion

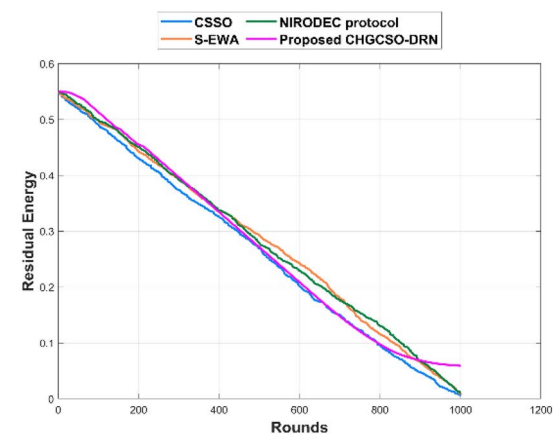
Table 4 depicts the evaluation accompanying 100, 150 and 200 nodes. Here, we discuss the best results. Utilizing 200 nodes, the very high evaluating accuracy of 0.943, specificity of 0.920, sensitivity of 0.933, and F1-score of 0.930 is calculated by the CHGCSO+DRN. The reasons for the better performance of the devised approach are discussed below: In the devised approach, the HGCSO is devised by combining HGSO and CSO, which inherit



(a)



(b)



(c)

Fig. 9. Assessment based on routing (a) 100 nodes, (b) 150 nodes, and (c) 200 nodes.

the advantages of both algorithms. Here, the HGSO is effective in addressing the engineering design issues whereas the CSO can acquire enhanced optimization outcomes in both robustness and accuracy. The proposed CHGCSO is developed by combining the CAViaR model and HGCSO is used to train DRN. The CAViaR model indicates the evolution of quantile in contrast to time considering an autoregressive procedure and evaluates the attributes with respect to regression quantiles. It is a statistical method, which is utilized to compute the quantity of potential loss that could occur over a particular time period using past records. Thus, the effectiveness of the devised approach is optimum than the conventional approaches.

100 nodes				
Methods/ Training data	DRN	HGSO+DRN	CSO+DRN	Proposed CHGCSO+DRN
Accuracy				
50	0.697	0.712	0.727	0.735
60	0.738	0.742	0.757	0.765
70	0.797	0.790	0.806	0.815
80	0.817	0.820	0.837	0.846
90	0.835	0.842	0.859	0.868
Sensitivity				
50	0.656	0.695	0.709	0.716
60	0.682	0.713	0.728	0.735
70	0.697	0.766	0.782	0.789
80	0.746	0.802	0.818	0.826
90	0.786	0.820	0.837	0.846
Specificity				
50	0.668	0.693	0.707	0.715
60	0.718	0.751	0.767	0.775
70	0.761	0.810	0.826	0.835
80	0.811	0.840	0.857	0.866
90	0.849	0.861	0.879	0.888
F1-score				
50	0.687	0.703	0.717	0.724
60	0.728	0.731	0.746	0.754
70	0.785	0.779	0.795	0.803
80	0.806	0.809	0.825	0.834
90	0.823	0.830	0.847	0.855
150 nodes				
Methods/ Training data	DRN	HGSO+DRN	CSO+DRN	Proposed CHGCSO+DRN
Accuracy				
50	0.699	0.751	0.767	0.774
60	0.709	0.782	0.798	0.806
70	0.718	0.833	0.850	0.859
80	0.777	0.865	0.883	0.892
90	0.797	0.887	0.906	0.915
Sensitivity				
50	0.718	0.727	0.742	0.750
60	0.738	0.746	0.762	0.769
70	0.787	0.801	0.818	0.826
80	0.797	0.839	0.856	0.865
90	0.837	0.858	0.876	0.885
Specificity				
50	0.708	0.729	0.744	0.752
60	0.738	0.791	0.807	0.815
70	0.788	0.852	0.869	0.878
80	0.828	0.864	0.882	0.891
90	0.856	0.877	0.895	0.904
F1-score				
50	0.689	0.741	0.756	0.764
60	0.699	0.771	0.787	0.795
70	0.708	0.821	0.838	0.847
80	0.766	0.853	0.870	0.879
90	0.785	0.875	0.893	0.902
200 nodes				
Methods/ Training data	DRN	HGSO+DRN	CSO+DRN	Proposed CHGCSO+DRN
Accuracy				
50	0.738	0.769	0.785	0.793
60	0.776	0.820	0.837	0.846
Continued				

200 nodes				
Methods/ Training data	DRN	HGSO+DRN	CSO+DRN	Proposed CHGCSO+DRN
70	0.818	0.860	0.878	0.887
80	0.838	0.885	0.903	0.912
90	0.856	0.915	0.934	0.943
Sensitivity				
50	0.738	0.767	0.783	0.791
60	0.806	0.830	0.847	0.856
70	0.857	0.870	0.888	0.897
80	0.876	0.885	0.903	0.912
90	0.907	0.905	0.924	0.933
Specificity				
50	0.771	0.764	0.780	0.788
60	0.791	0.779	0.795	0.803
70	0.796	0.790	0.806	0.815
80	0.822	0.821	0.838	0.847
90	0.881	0.893	0.911	0.920
F1-score				
50	0.728	0.758	0.774	0.782
60	0.765	0.809	0.826	0.834
70	0.807	0.848	0.866	0.874
80	0.826	0.872	0.890	0.899
90	0.844	0.902	0.921	0.930

Table 3. Ablation Study.

Nodes	Metrics	SCA based RideNN	RF+SVM+LR	DCNN	ResNet	MCFN	ESRGAN	Proposed CHGCSO+DRN
100 nodes	Accuracy	0.784	0.767	0.776	0.826	0.784	0.767	0.868
	Sensitivity	0.790	0.775	0.735	0.769	0.741	0.756	0.846
	Specificity	0.796	0.786	0.814	0.819			0.888
	F1-score	0.773	0.757	0.765	0.815	0.773	0.757	0.855
150 nodes	Accuracy	0.826	0.809	0.818	0.871	0.774	0.796	0.915
	Sensitivity	0.826	0.810	0.768	0.805	0.796	0.815	0.885
	Specificity	0.837	0.827	0.857	0.861	0.875	0.835	0.904
	F1-score	0.815	0.798	0.806	0.859	0.763	0.785	0.902
200 nodes	Accuracy	0.933	0.831	0.854	0.867	0.825	0.816	0.943
	Sensitivity	0.889	0.883	0.904	0.860	0.816	0.825	0.933
	Specificity	0.831	0.747	0.749	0.912	0.856	0.875	0.920
	F1-score	0.920	0.820	0.842	0.855	0.814	0.805	0.930

Table 4. Comparative Discussion.

Methods	Computation time (sec)
SCA based RideNN	44.562
RF+SVM+LR	43.675
DCNN,	48.367
MCFN	41.537
ESRGAN	40.966
DRN	47.895
ResNet	45.736
CHGCSO+DRN	39.671

Table 5. Analysis based on Computation Time.

Analysis based on computation time

Table 5 displays the analysis based on computation time. Here, when comparing the existing methods, like, SCA based RideNN, RF+SVM+LR, DCNN, MCFN, ESRGAN, DRN, ResNet, and the CHGCSO+DRN attains minimum computation time.

Conclusion and future work

A novel model has been developed for plant disease classification in an IoT network. The process begins with the simulation of the IoT network to monitor crop diseases. Routing is performed using the HGCSO algorithm, where the fitness function incorporates energy, delay, distance, and LLT. At the BS, plant disease classification is conducted by collecting images of rice crops. Each rice crop image undergoes preprocessing, with median filtering applied to eliminate noise. Then, feature extraction is performed to identify key characteristics. Finally, plant disease classification is achieved using a Deep Recurrent Network (DRN), where the DRN is trained with the CHGCSO algorithm. The CHGCSO is a hybrid model combining the CAViaR model and HGCSO. The CHGCSO-based DRN outperforms existing methods, achieving a maximum accuracy of 94.3%, sensitivity of 93.3%, specificity of 92%, and an F1-score of 93%. In future, the applicability of the model will be extended beyond rice crops to support disease detection in a variety of plant types, enabling broader usage in agriculture. Also, the test will be conducted under different environmental conditions and IoT network configurations to evaluate and enhance the reliability of the model and accuracy across diverse real-world scenarios.

Data availability

The data and code supporting this study's findings are available from the corresponding author upon reasonable request.

Received: 20 September 2024; Accepted: 3 January 2025

Published online: 09 January 2025

References

- Abayomi-Alli, O. O., Damaševičius, R., Misra, S. & Maskeliūnas, R. Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning. *Expert Systems* **38**, e12746 (2021).
- Almadhor, A. et al. AI-driven framework for recognition of guava plant diseases through machine learning from DSLR camera sensor based high resolution imagery. *Sensors* **21**, 3830 (2021).
- Argüeso, D. et al. Few-Shot Learning approach for plant disease classification using images taken in the field. *Computers and Electronics in Agriculture* **175**, 105542 (2020).
- Balachandra, M., Prema, K. & Makkithaya, K. Multiconstrained and multipath QoS aware routing protocol for MANETs. *Wireless networks* **20**, 2395–2408 (2014).
- Chandanapalli, S. B., Reddy, E. S. & Lakshmi, D. R. Convolutional neural network for water quality prediction in WSN. *Journal of Networking and Communication Systems* **2**, 40–47 (2019).
- Chavan, V. G., Patil, P. R., Potgantwar, A. D. & Bhaladhare, P. R. A Novel Intelligent Efficient and Dynamic Cluster Based Routing Protocol for IoT Assisted Precision Agriculture. *Journal of Algebraic Statistics* **13**, 4025–4037 (2022).
- Chen, M.-R., Yang, L.-Q., Zeng, G.-Q., Lu, K.-D. & Huang, Y.-Y. IFA-EO: An improved firefly algorithm hybridized with extremal optimization for continuous unconstrained optimization problems. *Soft Computing* **27**, 2943–2964 (2023).
- Chen, Z., Chen, Y., Wu, L., Cheng, S. & Lin, P. Deep residual network based fault detection and diagnosis of photovoltaic arrays using current-voltage curves and ambient conditions. *Energy Conversion and Management* **198**, 111793 (2019).
- Chen, Z., He, M., Liang, W. & Chen, K. Trust-aware and low energy consumption security topology protocol of wireless sensor network. *Journal of Sensors* **2015**, 716468 (2015).
- Engle, R. F. & Manganelli, S. CAViaR: Conditional autoregressive value at risk by regression quantiles. *Journal of business and economic statistics* **22**, 367–381 (2004).
- Fausto, F., Cuevas, E. & Gonzales, A. A new descriptor for image matching based on bionic principles. *Pattern Analysis and Applications* **20**, 1245–1259 (2017).
- Guniseti, L., Koduri, S. B. & Jagannathan, V. Optimized deep learning system for smart maize leaf disease detection in IoT platform via routing algorithm. *Multimedia Tools and Applications* **82**, 13533–13555 (2023).
- Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W. & Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Future Generation Computer Systems* **101**, 646–667 (2019).
- Hu, W.-J. et al. MDFC-ResNet: An agricultural IoT system to accurately recognize crop diseases. *IEEE Access* **8**, 115287–115298 (2020).
- Khirade, S. D., & Patil, A. B. In 2015 International conference on computing communication control and automation. (IEEE, 2015), pp. 768–771.
- Kulkarni, O. in 2018 fourth international conference on computing communication control and automation (ICCUBEA). (IEEE, 2018), pp. 1–4.
- Kundu, N. et al. IoT and interpretable machine learning based framework for disease prediction in pearl millet. *Sensors* **21**, 5386 (2021).
- Lakshmi, N. D., Latha, Y. M., & Damodaram, A. Silhouette extraction of a human body based on fusion of HOG and graph-cut segmentation in dynamic backgrounds. (2013).
- Lian, J. et al. The educational competition optimizer. *International Journal of Systems Science* **55**, 3185–3222 (2024).
- Maratha, P. & Gupta, K. HFLBSC: Heuristic and fuzzy based load balanced, scalable clustering algorithm for wireless sensor network. *Wireless Personal Communications* **125**, 281–304 (2022).
- Maratha, P. & Gupta, K. Linear optimization and fuzzy-based clustering for WSNs assisted internet of things. *Multimedia Tools and Applications* **82**, 5161–5185 (2023).
- Maratha, P., Gupta, K. & Luhach, A. K. Improved fault-tolerant optimal route reconstruction approach for energy consumed areas in wireless sensor networks. *IET Wireless Sensor Systems* **10**, 112–116 (2020).
- Meng, X., Liu, Y., Gao, X., & Zhang, H. In *Advances in Swarm Intelligence: 5th International Conference, ICSI 2014, Hefei, China, October 17–20, 2014, Proceedings, Part I* 5. (Springer, 2014), pp. 86–94.
- Mishra, M., Choudhury, P. & Pati, B. Modified ride-NN optimizer for the IoT based plant disease detection. *Journal of Ambient Intelligence and Humanized Computing* **12**, 691–703 (2021).
- Moraru, A., Pesko, M., Porcius, M., Fortuna, C. & Mladenec, D. Using machine learning on sensor data. *Journal of computing and information technology* **18**, 341–347 (2010).

26. Ni, L. et al. Pi puf: a processor-intrinsic puf for iot. *Computers and Electrical Engineering* **105**, 108540 (2023).
27. Peter, A. P. et al. Artificial intelligence model for monitoring biomass growth in semi-batch *Chlorella vulgaris* cultivation. *Fuel* **333**, 126438 (2023).
28. Picon, A. et al. Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. *Computers and Electronics in Agriculture* **161**, 280–290 (2019).
29. Picon, A. et al. Crop conditional Convolutional Neural Networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions. *Computers and Electronics in Agriculture* **167**, 105093 (2019).
30. Qi, A. et al. FATA: an efficient optimization method based on geophysics. *Neurocomputing* **607**, 128289 (2024).
31. Ramesh, S. & Rajaram, B. Iot based crop disease identification system using optimization techniques. *ARPN J. Eng. Appl. Sci* **13**, 1392–1395 (2018).
32. Rehman, Z. U. et al. Recognizing apple leaf diseases using a novel parallel real-time processing framework based on MASK RCNN and transfer learning: An application for smart agriculture. *IET Image Processing* **15**, 2157–2168 (2021).
33. Ruby, R. et al. Enhancing secrecy performance of cooperative NOMA-based IoT networks via multiantenna-aided artificial noise. *IEEE Internet of Things Journal* **9**, 5108–5127 (2021).
34. Sabrol, H., & Satish, K. In *2016 international conference on communication and signal processing (ICCSP)*. (IEEE, 2016), pp. 1242–1246.
35. Shi, B. et al. Prediction of recurrent spontaneous abortion using evolutionary machine learning with joint self-adaptive sine mould algorithm. *Computers in Biology and Medicine* **148**, 105885 (2022).
36. Shibani, K., Kumar, K. Sendhil, & Shanmugam, G. Siva. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. (Springer, 2020), pp. 143–154.
37. Shrivastava, V. K. & Pradhan, M. K. Rice plant disease classification using color features: a machine learning paradigm. *Journal of Plant Pathology* **103**, 17–26 (2021).
38. Su, H. et al. RIME: A physics-based optimization. *Neurocomputing* **532**, 183–214 (2023).
39. Tan, X. & Triggs, B. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE transactions on image processing* **19**, 1635–1650 (2010).
40. Unni, R. et al. Enhancement of mechanical and thermal properties of *Ixora coccinea* L. plant root derived nanocellulose using polyethylene glycol-glutaraldehyde system. *Chemosphere* **298**, 134324 (2022).
41. Vimala, S., Merlin, N. Gladiss, Ramanathan, L., & Cristin, R. Optimal routing and deep regression neural network for rice leaf disease prediction in IoT. *International Journal of Computational Methods* **18**, 2150014 (2021).
42. Wen, J., Shi, Y., Zhou, X. & Xue, Y. Crop disease classification on inadequate low-resolution target images. *Sensors* **20**, 4601 (2020).
43. Yuan, C. et al. Polar lights optimizer: Algorithm and applications in image segmentation and feature selection. *Neurocomputing* **607**, 128427 (2024).
44. Yuan, C. et al. Artemisinin optimization based on malaria therapy: Algorithm and applications to medical image segmentation. *Displays* **84**, 102740 (2024).
45. Zhao, Y. et al. An effective automatic system deployed in agricultural Internet of Things using Multi-Context Fusion Network towards crop disease recognition in the wild. *Applied Soft Computing* **89**, 106128 (2020).

Author contributions

Ashish Saini: Conceptualization, Problem formulation, Methodology, Original draft preparation, Reviewing and Editing, and Final drafting. **Nasib Singh Gill:** Data curation, Programming, Simulation, Validation. **Preeti Gulia:** Numerical analysis, Visualization, and System set-up. **Anoop Kumar Tiwari:** Mathematical Modelling, Visualization, and Investigation. **Priti Maratha:** Supervision, Problem formulation, Programming, Validation, Writing, Reviewing, and Editing. **Mohd Asif Shah:** Conceptualization, Supervision, Validation, Reviewing, Ensuring Ethical Standards.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to P.M. or M.A.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025