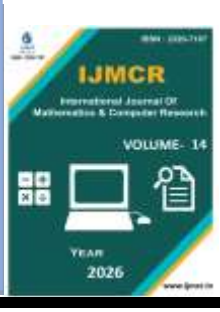
International Journal of Mathematics and Computer Research

ISSN: 2320-7167

Volume 14 Issue 02 February 2026, Page no. – 6168-6182

Index Copernicus ICV: 57.55, Impact Factor: 8.615

DOI: 10.47191/ijmcr/v14i2.04



Improved Cassava Leaf Disease Classification through Enhanced Support Vector Machine Learning Models

O.B. Ayoade* (PhD)[1], M.O. Raji (PhD)[2], A.A. Akindele[3], K.J. Yusuf-Mashopa[4], M.F. Abdulrauff[5], I. A. Raji[6], F.B. Musah[7]

[1,2,3,4,5,6,7]Department of Data Science, Informatics and Computer Science, Emmanuel Alayande University of Education, Oyo, Nigeria

ARTICLE INFO

Published Online:
09 February 2026

ABSTRACT

Cassava leaf diseases significantly reduce both the quantity and quality of cassava production. To address this challenge, the study developed two enhanced machine learning models—Enhanced Binary Particle Swarm Optimisation-Support Vector Machine (EBPSO-SVM) and Enhanced Reptile Search Algorithm-Support Vector Machine (ERSA-SVM)—for improved multiclass classification of cassava leaf diseases. The research utilised a dataset of 2,180 cassava leaf images from the Kaggle village repository, comprising 466 images each of cassava bacterial blight disease (CBBD), cassava brown streak disease (CBSD), cassava green mottle or mite disease (CGMD), cassava mosaic disease (CMD), and 316 images of healthy leaves. Preprocessing involved image resizing, RGB to grayscale conversion, and contrast enhancement using bi-histogram equalisation. The affected areas were segmented using the Sobel edge detection method, while Gray Level Spatial Dependence and colour moment techniques were employed for texture, shape, and colour feature extraction. Comparative experiments revealed that both the EBPSO-SVM and ERSA-SVM models achieved superior performance with an average classification accuracy of 96.42% and 95.14%, respectively, outperforming both the BPSO-SVM and RSA-SVM models, which attained an average accuracy of 95.40% and 93.86%, respectively. These findings demonstrate the effectiveness of the enhanced optimisation algorithms in overcoming dataset imbalance, premature convergence, and local optima challenges that often hinder model accuracy. In conclusion, the proposed EBPSO-SVM and ERSA-SVM models enhance classification precision and efficiency in cassava disease detection. Their adoption could facilitate early and automated identification of cassava leaf diseases, contributing to improved crop management, increased agricultural productivity, and enhanced food security in cassava-dependent regions.

Corresponding Author:
O.B. Ayoade

KEYWORDS: Cassava leaf Disease, Cassava Bacterial Blight Disease, Gray level Spatial Dependence, Optimised Support Vector Machine, Premature Convergence

I. INTRODUCTION

Cassava (manihot esculenta crantz), also known as yucca, cassava or tapioca, is one of the most important carbohydrate starch-rich human foods found especially in sub-Saharan Africa and South America. Common edible parts of the plant are the leaves and starchy roots [1]. Cassava provides food security and income opportunities for millions of people, particularly in rural areas where other crops may not grow well. Leaf disease is common in many plants [2]. Cassava crops can become infected with leaf diseases, reducing overall production and farmers' income [3]. Cassava leaf diseases, such as Cassava Green Mite Disease (CGMD), Cassava Bacteria Blight Disease (CBBD), Cassava Brown Streak Disease (CBSD), Cassava Mosaic Disease (CMD), Cassava American Latent Leaf Disease (CALD), Cassava Brown Streak Uganda Disease (CBSUD), and Cassava Colombian Symptomless Disease (CCSD), have severely impacted cassava production and resulted in significant economic losses [4].

Cassava Mosaic Disease (CMD), which makes the leaves wrinkled and yellowish, and Cassava Bacterial Blight Disease (CBBD) [5], which causes root rot, are the major cassava leaf diseases that serve as threats to food security in various parts of Africa. Other diseases of cassava include Cassava Green Mite Disease (CGMD), which causes milky spots on leaves [6], and Cassava Brown Streak Disease (CBSD), which has symptoms of red necrotic areas in oval roots [2]. The traditional style of detecting plant disease with human intervention is laborious and cannot detect cassava disease conveniently [7]. It becomes more cumbersome and creates a burden on farmers. The farmers can benefit from developing an automated system for early detection and prevention of disease in cassava leaves. In the modern era, Machine Learning (ML) algorithms based on Support Vector Machine play a predominant role in providing solutions to problems related to prediction and classification.

Support Vector Machine (SVM) can effectively find the best separating hyperplane between classes, maximising the margin between them, but it can be sensitive to noise and outliers, especially when classes overlap considerably [8]. Fine-tuning the hyperparameters of the SVM can be challenging [9], and SVM models are sensitive to the scaling of input features [10]. However, by using optimisation techniques such as Binary Particle Swarm Optimisation (BPSO) and Reptile Search Algorithm (RSA), SVM can be made more robust to noise, outliers, and overlapping classes, leading to improved performance and generalisation ability. Optimisation techniques such as BPSO and RSA can help SVM in scaling input features to ensure that all features contribute equally to the decision boundary by preventing features with larger values from dominating the model, leading to an improved model performance and stability. In addition, BPSO and RSA can make it possible to overcome the challenges associated with fine-tuning the SVM hyperparameters.

Therefore, several researchers have developed models using optimisation techniques to improve the performance of the classification algorithm. For example, Nivethithaa and Vijayalakshmi [11] created a classification model for maize and rice leaf disease detection by optimising the Support Vector Machine (SVM) with a Fish Swarm Optimiser (FSO). The FSO-SVM model developed had a higher performance accuracy of 98.3%. Kour and Arora [12] also created a classification model to detect diseases affecting seven plants: Guava, Jamun, Mango, Grapes, Apple, Tomato, and Arjun. They used Particle Swarm Optimisation (PSO) to optimise the Support Vector Machine (SVM). The PSO-SVM model developed had a higher performance accuracy of 95.25%.

Ghazalli and Roslan [13] developed a machine-learning model that detects Cassava Bacterial Blight Disease (CBBD) and Cassava Mosaic Disease (CMD) on cassava leaves. The authors obtained 200 images of cassava leaf disease, including 100 CBBD and 100 CMD, from the Kaggle Dataset. The dataset was divided into 80% training and 20% testing. The preprocessing stage was then carried out, which included resizing and increasing the brightness of the leaf image to improve image quality. ROI is used to separate the cassava leaf from the background and other unnecessary elements of the image. After the segmentation process, colour and texture features are extracted using the colour moment and gray level co-occurrence matrix (GLCM), respectively. To classify, the dataset is trained with the SVM classifier before the accuracy test. It was discovered that SVM achieves 87.5% accuracy.

However, BPSO is prone to premature convergence, where the algorithm gets stuck in a local optimum before reaching the global optimum, due to the way particles move and interact [14]. The sigmoid transfer function in BPSO may not accurately capture the characteristics of discrete optimisation, leading to suboptimal results [15]. BPSO, like other optimisation algorithms, is sensitive to parameter settings such as inertia weight, acceleration coefficients, and population size [16]. Finding the optimal parameter settings can be challenging and time-consuming.

While RSA is a promising optimisation technique inspired by crocodile hunting, it struggles with some issues which limit their performance in optimisation problems. These issues arise from the algorithm's inherent complexity and the computational resources required for its execution. RSA's complexity stems from its simulation of crocodile behaviour, which involves multiple phases such as encircling, hunting, and prey capture. Each of these phases adds to the overall computational burden, especially when handling complex or high-dimensional optimisation problems. RSA, like other algorithms inspired by biological patterns, can have a high computational overhead due to the need to track and manage the population of "reptile searchers" (simulating crocodiles) and their interactions with the environment [17]. These limitations lead to slow convergence speed, making it difficult for the algorithm to reach the global optimum promptly. Additionally, the algorithm can get trapped in local optima, failing to find the best solution [18].

Therefore, in this study, an enhanced binary particle swarm optimisation (EBPSO) model, which used three strategies, namely Chaotic Map, Arc Tangent Acceleration Coefficient, and Cosine Map Inertia Weight, was developed to improve the original binary particle swarm optimisation. Also, an enhanced reptile search algorithm (ERSA) model, which used three strategies, namely Dynamic Evolutionary Sense, Prey Approaching Strategy, and Cauchy Mutation Strategy, to improve the original reptile search algorithm. Our objective is to propose classification models that will achieve better classification results with reduced computational complexity using enhanced machine learning models.

II. MATERIALS AND METHODS

In this subsection, the methodology of the proposed method is presented. As shown in Figure 1, the process begins with dataset acquisition, followed by preprocessing, where different process was performed on the dataset to improve the quality for the classification models. The preprocessed dataset was segmented using the Sobel Edge Detection method. After texture, colour and shape features were extracted from the segmented dataset, it is now split into training and testing datasets using 10-fold cross-validation. Next, the model-building process involves leaf detection and classification models using the training dataset images. Finally, the testing images are deployed on the developed leaf detection and disease classification system to evaluate its performance.

A. Dataset Acquisition

Using the Kaggle Village dataset, 2,180 photos of cassava leaves were acquired. The same number of datasets from the original dataset was chosen at random for each class of the diseased datasets to avoid having an uneven dataset for class labelling. The dataset is divided into five categories: 316 photos showing healthy cassava leaves, 466 photos showing cassava brown streak disease (CBSD), 466 photos showing cassava bacterial blight disease (CBBD), 466 photos showing cassava green mite/mottle disease (CGMD), and 466 photos showing cassava mosaic disease (CMD). Figure 2 presents samples from the dataset used, and a 10-fold cross-validation approach was used to select the training and test datasets; therefore, for the cassava dataset, 218 datasets were used for testing, and the remaining 1,962 datasets were used for training, as shown in Table 1.

B. Dataset Pre-processing

Pre-processing was done on the gathered images to reduce noise, remove blur, and improve the images' quality for the subsequent processing stages. Initially, MATLAB's image resizer toolbox was used to resize the RGB images. The images of cassava leaves were resized from 512 X 512 pixels resolution to 256 X 256 pixels resolution to eliminate unnecessary pixel information and streamline the classification model. RGB images were converted to grayscale using the luminosity method, indicated by Equation 2.1, which was put forth by Padmavathi and Thangadurai [19]. This takes up less space and thus processes complex computations faster than RGB images. These can be used for fast object identification within an image. In this research, the bi-histogram equalisation technique was used for Contrast Enhancement. Image enhancement improves an image's characteristics for human visual perception and improves feature extraction for additional image processing analysis. Then, morphological filtering was used to sharpen the image; this process enhanced the image further. Lastly, adaptive median filtering was used to denoise the images before image segmentation processing.

Table 1: Distribution of the Cassava Dataset Acquired from the Kaggle Village

Class	Content	Number of Images per Class	Number of Training Datasets	Number of Testing Datasets
1	Cassava Bacterial Blight Disease (CBBD)	466	1,962	218
2	Cassava Brown Streak Disease (CBSD)	466		
3	Cassava Green Mite/ Mottle Disease (CGMD)	466		
4	Cassava Mosaic Disease (CMD)	466		
5	Healthy Cassava Leaves	316		

$$Grayscale = 0.2989 * Red + 0.587 * Green + 0.114 * Blue \quad (2.1)$$

C. Image Segmentation Procedure and Feature Extraction

The suggested models use the Sobel edge detection method to separate lesions from the uninfected part of a leaf. The process of using the Sobel edge detection method involved two stages. Firstly, the gradient of the image along the x-axis and y-axis is determined using the Sobel operator. After finding the image gradient, the next step is to find a threshold value automatically so that edges can be determined [20]. According to Aslam *et al*. [21], equations 2.2, 2.3, 2.4, and 2.5 were used to compute the image gradient along the x and y-axis and equations 2.6, 2.7 and 2.8 were used to calculate the threshold value. Algorithm 2.1 demonstrated an example of the image-dependent threshold computation and edge point Algorithm's pseudo-code.

$$g_x = \frac{\delta f}{\delta x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad (2.2)$$

$$g_y = \frac{\delta f}{\delta y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \tag{2.3}$$

Then the gradient of the image is defined as:

$$\nabla f(x, y) = \frac{\delta f}{\delta x}\hat{h} + \frac{\delta f}{\delta y} = g_x\hat{h} + g_y\hat{s} \tag{2.4}$$

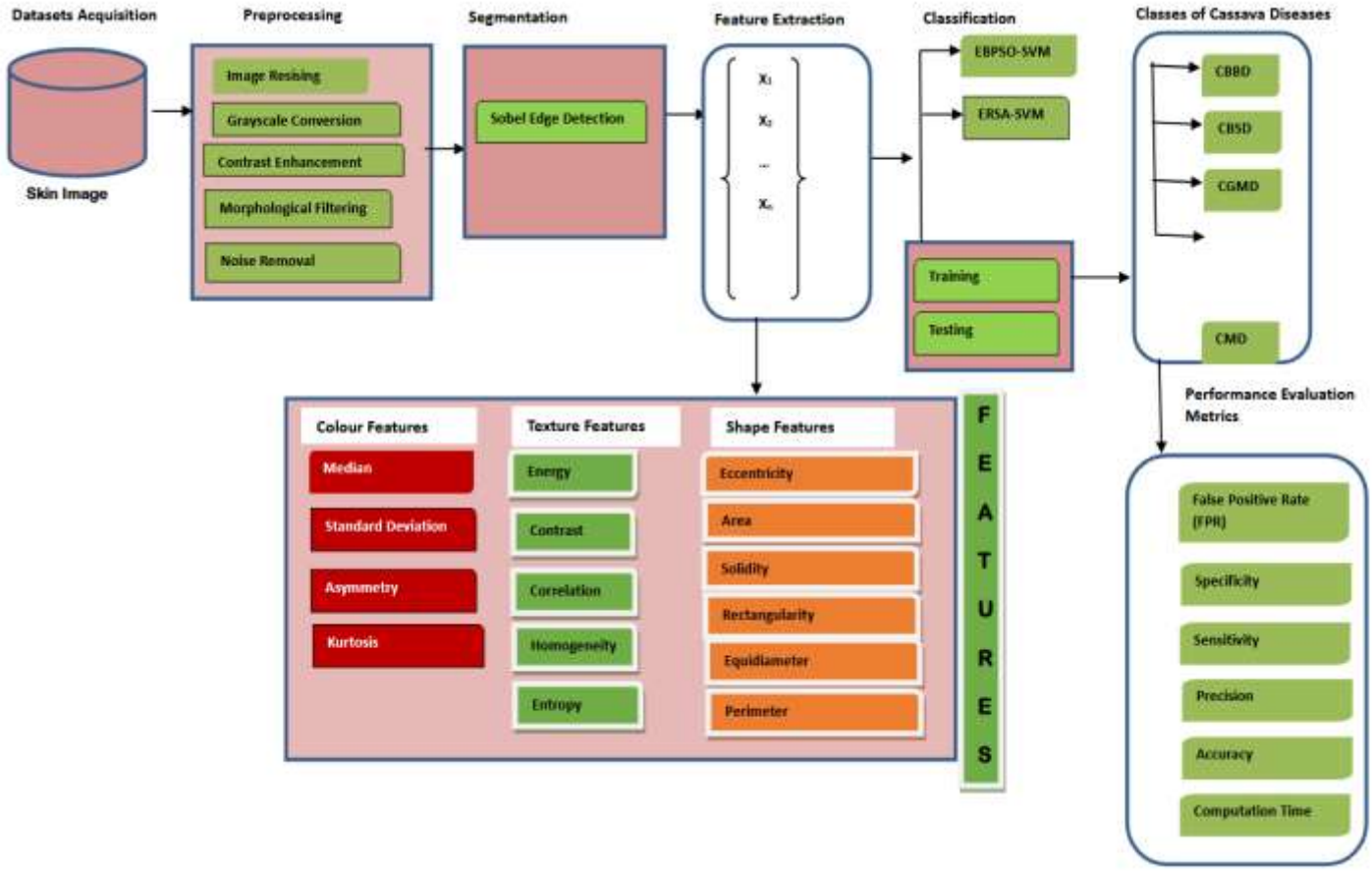


Fig. 1: A Block Diagram of the Developed Cassava Diseases Classification Models

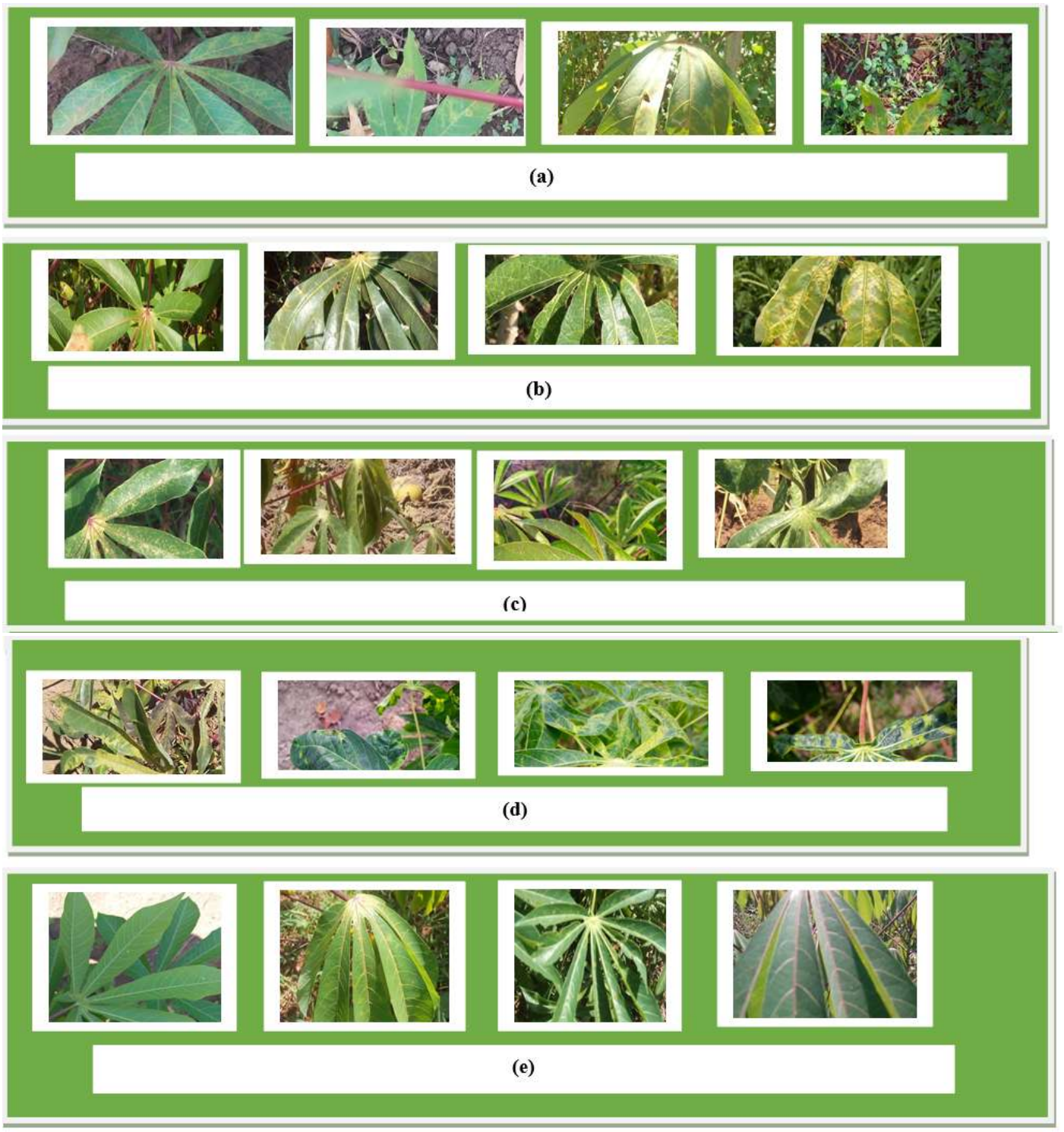


Fig. 2. Samples of the Cassava Dataset used for the Study (a) Cassava Brown Streak Disease (CBSD), (b) Cassava Bacterial Blight Disease (CBBD), (c) Cassava Green Mottle/Mite Disease (CGMD), (d) Cassava Mosaic Disease (CMD), (e) Healthy Cassava Leaf (Source: [20])

where f(x,y) is the image, g(x,y) is the gradient image and $\hat{h}$ & $\hat{s}$ are unit vectors along the *x* and *y* axes, respectively. The magnitude of the gradient is given by equation 3.5:

$$g(x,y) = |\nabla f(x,y)| = \sqrt{gx^2 + gy^2} \quad (2.5)$$

Algorithm 2.1: Computation of Threshold Value of the Image (Source: Adapted from [21])

1: Input: a, height of the image, b, width of the image, and g(x,y): gradient of the image

2: Utilising Equation 3.6, compute the average intensity of the gradient image g(x,y)
The initial threshold, Thd^0, is equal to the average intensity of the gradient image *g(x, y)*, as defined in equation 3.6

$$Thd^0 = \frac{\sum_{k=1}^{a} \sum_{l=1}^{b} g(x,y)}{\mathrm{a} \times \mathrm{b}} \quad (2.6)$$

3: Set iteration index $i = 0$, separate *g(x,y)* into two classes, where the lower class consists of those pixels of *g(x,y)* which have gradient values less than Thd^i, and the upper class contains the rest of the pixels.

4: Compute the average gradient values m_{Low} and m_{High} of the lower and upper classes, respectively.

5: Set iteration $i = i+1$ and update threshold value as:

$$Thd^i = \frac{m_{Low} + m_{High}}{2} \quad (2.7)$$

6: Repeat steps 2 to 4 until $|Thd^i - Thd^{i-1}| \leq \epsilon$, where $\epsilon \rightarrow 0$ and take Thd^i as the final threshold and denote it by *Thd.*

7: Compute the output utilising equation 3.8

$$E(x,y) = \begin{cases} 255 & g(x,y) \geq Thd, \\ 0 & otherwise \end{cases} \quad (2.8)$$

8: Comparing the value of the gradient image with the threshold value to determine the output:
If *g(x,y) > Thd*
Output: Return edge point (represented as white)
Endif
If *g(x,y) < Thd*
Output: Return background point (represented as black)
Endif

After segmentation, the shape and texture features of the image were extracted using the Gray Level Spatial Dependence, and colour features were extracted using colour moments. Colour, texture and shape are the three features extracted in this study.

D. Formulation of Enhanced Binary Particle Swarm Optimisation (EBPSO) Model and Enhanced Reptile Search Algorithm (ERSA)

The algorithms 2.2 and 2.3 depict both the enhanced binary particle swarm optimisation model and the enhanced reptile search model.

Algorithm 2.2: Enhanced Binary Particle Swarm Optimisation (EBPSO) Model

Input: Population Size (N), maximum number of iterations F_{max}

1: Initialise the particles' position ($Z_{i,j}^f$), the velocity constant (Vel_c), previous best position ($h_{i,j}^f$), acceleration coefficient (C_1: factor of constant cognitive, C_2: factor of social scaling), number of particles (N), set *Iteration*=0, and maximum number of iterations (F_{max})

Outputs: Solution to the problem

2: **Local Search**
According to Gou *et al*. [22], Chen *et al*. [23], and Lynn *et al*. [24] claim that while BPSO is reliable for handling challenging optimisation problems, its quick convergence can easily trap the swarm inside certain local optima. Its algorithm also lacks diversity in its search process ([25-28]). Therefore, chaotic, Arc Tangent Acceleration Coefficients (AT), and Cosine Map Inertial were used to enhance the BPSO.
Applied Cosine Map Inertial

3: Evaluate inertial weight ω using Equation 2.9 as in Ma *et al.* [29]

$$\omega = \delta \times \cos\left(\left(\frac{F_j}{F_{max}}\right) \times \pi\right) + \vartheta \quad (2.9)$$

Where δ and ϑ are constant (δ=1/3, ϑ= 0.6), F_j and F_{max} denote the number of present iterations and the maximum number of iterations defined by the user, respectively.

4. Evaluate maximum velocity (v_{max}) using

$$Vel_{\max} = \left(e^{1-f/F\max}\right) Vel_c$$

Applied Arc Tangent Acceleration Coefficients (AT)

5: Using Equations 2.10 and 2.11 to update the cognitive component C_1 and the social component C_2 as in Ma *et al.* [35]

$$C_1 = -\gamma \times arctan((\frac{F_j}{F_{max}}) \times \beta) + \rho_1 \quad (2.10)$$

$$C_2 = \gamma \times arctan((\frac{F_j}{F_{max}}) \times \beta) + \rho_2 \quad (2.11)$$

where γ, β, ρ_1, and ρ_2 are constant (γ = 1.5, β = 4, ρ_1 = 2.5, and ρ_2 = 0.5)

6: while (iteration < F*max*) do

7: For all particles (i) do

8: Using the current position $z_{i,j}^t$ of the i^{th} as a starting point, calculate each particle's performance L (L ($z_{i,j}^f$)) using Equation 2.12

$$Z_{i,j}^{(f+1)} = Z_{i,j}^{f} + vel_{i,j}^{(f+1)} \quad (2.12)$$

9: Evaluate every single individual's performance with their greatest hitherto:
If L ($z_{i,j}^f$) < L ($h_{i,j}^f$) then
$h_{i,j}^f = z_{i,j}^f$
End if

10: Each particle's performance is compared to the universal best particle.
If L ($z_{i,j}^f$) < L ($h_{g,j}^f$) then
$h_{g,j}^f = z_{i,j}^f$
End if

11: Update the new velocity of particle i^{th} ($vel_{i,j}^{(f+1)}$) using Equation 2.13

$$vel_{i,j}^{(f+1)} = \omega vel_{i,j}^{f} + C_1 \times r_1 \times (h_{i,j}^{f} - Z_{i,j}^{f}) + C_2 \times r_2 (h_{g,j}^{f} - Z_{i,j}^{f}) \quad (2.13)$$

where ω is the inertia weight; r_1 and r_2 are random numbers within [0,1]; C_1 and C_2 represent learning factors; $C_1 \times r_1 \times (h_{i,j}^{f} - Z_{i,j}^{f})$ is the individual cognition term, which represents the individual cognitive experience of the particles, and it causes the particles to move toward the best position they experience; and $C_2 \times r_2 (h_{g,j}^{f} - Z_{i,j}^{f})$is the social cognition term, which denotes the influence of the group experience n the flight path of particles. The social cognition term moves the particles toward the best position found by the group, representing the sharing of information between the particles.

12: Update the new position of particle ith ($z_{i,j}^{(t+1)}$) using
If rnd() < P($vel_{i,j}$), then $z_{i,j}$ = 1; else $z_{i,j}$ =0;
where

$$P(\upsilon) = \frac{1}{1 + e^{-vel(i,j)}}$$

$$Z_{i,j}^{(f+1)} = Z_{i,j}^{f} + vel_{i,j}^{(f+1)}$$

End for

11: If ($vel_{(i,j)}$ > vel_{max})
$vel_{(i,j)}$= vel_{max}
Endif

12: If ($vel_{(i,j)}$ < -vel_{max})
$vel_{(i,j)}$= -vel_{max}
Endif

13: **Global Search**
Applied Chaotic

14: Chaotic search K times near gbest

15: Use Equation 2.12 to generate a chaotic random sequence D between [0, 1] using the logistic equation, and then pass the carrier map in Equation 2.14, introducing chaos into gbest, a nearby area, to achieve local chaotic search as in Ma *et al.* [29]:

$$D \rightarrow A: Z = gbest + M \times \cos(D) \quad (2.14)$$

16: *E* chaotic search points near $pbest_f$ are obtained
17: for *j=1: E* do
18: if $f(Z) < f(gbest)$
19: Set $f(Z)$ to be *gbest*
20: End if
21: End for
22: Increase the number of iterations. *Iteration = Iteration +1*
23: End while
24: Return gbest

Algorithm 2.3: Enhanced Reptile Search Algorithm (ERSA) Model

Input: Number of Features (N), maximum number of iterations *T*

1. Initialise the RSA parameters, number of features (N), minimum and maximum feature boundaries (LWB, UPB), a sentient specification that governs the global optimum performance (δ), two parameter factors that contribute to the quality of the classification model (θ and μ), the number of population (Po), set iteration t=0, and the highest number of repetitions (T)

Outputs: Solution to the problem

2. Generate an initial population randomly using Equation 2.15

$$Z_{i,j} = random \times (UPB_j - LWB_j) + LWB_j, j = 1,2,\dots,M \quad (2.15)$$

where random denotes a random number between 0 and 1, $UPB_j \; and - LWB_j$ denote the upper bound and lower bound of the search space. M denotes the dimension of the optimisation problem. And $Z_{i,j}$ means the generated position of the i-th individual in the j-th dimension.

3: while t< T) do
4: For all populations (Po) do
5: Update RSA parameters using Equations 2.16, 2.17, 2.18, 2.19, and 2.20

$$\Psi_{(i,j)}(t) = B_j(t) \times P_{(i,j)}(t) \quad (2.16)$$

$$R_{(i,j)}(t) = \frac{B_j(t) - Z_{(r1,j)}}{B_j(t) + \epsilon} \quad (2.17)$$

According to Khan *et al.* [30], RSA has certain drawbacks, including highly complex computations, sluggish convergence, and neighbourhood minimum trapping. Also, the parameter ES(t) has a limited range of variation, which makes the RSA show poor global search ability and converge slowly [31]. To enhance the searching capabilities of RSA, an enhanced RSA is developed by using multiple strategies: dynamic evolutionary sense factor, prey approaching strategy and Cauchy mutation strategy.

Dynamic Evolutionary Sense (DES)

$$DES(t) = 2 \times r_2 \times \left(1 - \frac{t}{T}\right)^{\left(2 \times \frac{t}{T}\right)} \times randomn \quad (2.18)$$

where, $B_j(t)$ is the current best position within the population. t is the current iteration number, and T is the maximum iteration number; randomn means a random number satisfying the standard normal distribution. $\Psi_{(i,j)}(t)$ is the hunting operator, which is determined using Equation 2.16, $R_{(i,j)}(t)$ is a reduce function, which is calculated using Equation 2.17, r1 is a random integer between [1, N]. DES(t) is a factor used to improve the ES(t) to avoid early convergence, and it can be obtained by using Equation 2.18. ϵ is a very small positive number. r2 can be 1 or 1 randomly. $P_{(i,j)}(t)$ is the percentage difference between the current position and the best position, which is calculated using Equations 2.19 and 2.20.

$$P_{(i,j)}(t) = \delta + \frac{Z_{(i,j)} - N(z_1)}{B_j(t) \times (UPB_j - LWB_j) + \epsilon} \quad (2.19)$$

$$N(z_1) = \frac{1}{M} \sum_{j=1}^{M} Z_{(i,j)} \quad (2.20)$$

where represents a sensitive parameter controlling exploration performance, which is set to 0.1 in RSA as mentioned by Zhou *et al*. [31], and $N(z_1)$ signifies the average solution. To balance the exploration and exploitation performance of RSA, DES(t) is applied in the stages of high walking and hunting coordination.

6. **Phase 1: Encircling phase (Exploration Mechanisms, i.e., Global Search)**
Perform a high walking exploration mechanism using Equation 2.21

$$Z_{(i,j)}(t+1)=B_j(t)-(\Psi_{(i,j)}(t)\times\alpha+R_{(i,j)}(t)\times random)\times DES(t), if\ t\le T\times 0.5 \quad (2.21)$$

where, α is a constant value, indicating the exploration accuracy, which is set to 0.005 in RSA as mentioned by Zhou *et al*. [31], and r3 is a random integer between [1, N].

7. **Phase 2: Prey Approaching Strategy (PAS)**
A prey approaching strategy replaced the belly walking of RSA (PAS), which is based on the secretary bird optimisation algorithm (SBOA) proposed by Fu *et al.* [32]. The SBOA helps the RSA explore the search space more effectively. Equation 2.22 was used to mathematically express the SBOA model as in Zhou *et al*. [37]:

$$Z_{(i,j)}(t+1)=B_j(t)+\exp\left(\left(\frac{t}{T}\right)^4\right)\times(randomn-0.5)\times(B_j(t)-Z_{(i,j)}(t)) \quad (2.22)$$

8. **Phase 3: Hunting phase (Exploitation Mechanisms, i.e., Local Search)**
To balance the exploration and exploitation performance of RSA, DES(t) is applied in the stages of hunting coordination using Equation 2.23

$$Z_{(i,j)}(t+1)=B_j(t)-(\Psi_{(i,j)}(t)\times\epsilon+R_{(i,j)}(t)\times random)\times DES(t), if\ t>3\times T\times 0.25 \quad (2.23)$$

$$Z_{(i,j)}(t+1)=B_j(t)\times P_{(i,j)}\times random, if\ t\le 3\times T\times 0.25\ and\ t>T\times 0.5 \quad (2.24)$$

9. **Phase 4: Cauchy Mutation Strategy (CMS)**
According to Zhao *et al.* [33], the Cauchy Mutation Strategy (CMS) provide more potential candidates and enriches the population diversity of the metaheuristic algorithm. Therefore, the CMS was executed after the hunting phase of RSA. The new position after the mutation can be calculated using Equations 2.25 and 2.26

$$Z_{(i,j)}=Z_{(i,j)}\times(1+CauchyRandom) \quad (2.25)$$

$$CauchyRandom=a\times tan(\pi\times(random-0.5))+b \quad (2.26)$$

where the parameters a and b are set to 1 and 0 in ERSA as in Zhou *et al*. [31], respectively. CauchyRandom is a random number based on the standard Cauchy distribution.

8. End for
9. Increase the iteration t=t+1
10 If t>T, move to step 11; otherwise, step 4
11. End while
12. Return the best position

III. RESULTS AND DISCUSSION OF THE FINDINGS

The graphical user interface was developed with image processing, computer vision, and optimisation toolboxes. All algorithms are implemented in a Windows 11-64-bit environment using a 1.20 GHz frequency Intel(R) Core (TM) i3-1005G1 CPU, an x64 version with 8GB RAM, and programmed in MATLAB R2020a. The results of the experiments are described in subsections that follow this section.

A. Performance Evaluation Metrics of the Developed Multiclass Enhanced Support Vector Machine Learning Classification Models on Cassava Datasets

The EBPSO-SVM model's FPR of 3.48, 4.11, 4.43, and 3.79% outperformed the ERSA-SVM model's FPR of 5.58, 5.70, 5.06, and 6.01% on the diseased dataset for each class comprising CBBD, CBSD, CGMD, and CMD. On the diseased dataset, the EBPSO-SVM model's specificity of 96.52, 95.89, 95.57, and 96.20% outperformed the ERSA-SVM model's specificity of 94.62, 94.30, 94.94, and 93.99% for each class comprising CBBD, CBSD, CGMD, and CMD, respectively (Table 2). More so, the EBPSO-SVM model's sensitivity of 97.00, 96.57, 96.35, and 96.78% outperformed the ERSA-SVM model's sensitivity of 95.71, 95.49, 95.92, and 95.28% on the diseased dataset for each class comprising CBBD, CBSD, CGMD, and CMD, respectively (Table 2). Furthermore, the EBPSO-SVM model's precision of 97.62, 97.19, 96.98, and 97.41% outperformed the ERSA-SVM model's precision of 96.33, 96.11, 96.54, and 95.90% on the diseased dataset for each class comprising CBBD, CBSD, CGMD and CMD, respectively (Table 2). Likewise, the EBPSO-SVM model's accuracy of 96.80, 96.29, 96.04, and 96.55% outperformed the ERSA-SVM model's accuracy of 95.27, 95.01, 95.52, and 94.76% on the diseased dataset for each class comprising CBBD, CBSD, CGMD and CMD, respectively (Table 2). Finally, the EBPSO-SVM model's computation time of 48.77, 47.48, 48.72, and 47.03sec outperformed the ERSA-SVM model's computation time of 49.90, 47.80, 47.67, and 49.38sec on the diseased dataset for each class comprising CBBD, CBSD, CGMD and CMD, respectively (Table 2).

Table 2. Performance Evaluation Metrics of the Developed Multiclass Support Vector Machine Classification Models on Cassava Datasets

	CBBD	CBSD	CGMD	CMD	Average
FPR (%)					
EBPSO-SVM	3.48	4.11	4.43	3.79	3.95
ERSA-SVM	5.38	5.70	5.06	6.01	5.54
Specificity (%)					
EBPSO-SVM	96.52	95.89	95.57	96.20	96.05
ERSA-SVM	94.62	94.30	94.94	93.99	94.46
Sensitivity (%)					
EBPSO-SVM	97.00	96.57	96.35	96.78	96.68
ERSA-SVM	95.71	95.49	95.92	95.28	95.60
Precision (%)					
EBPSO-SVM	97.62	97.19	96.98	97.41	97.30
ERSA-SVM	96.33	96.11	96.54	95.90	96.22
Accuracy (%)					
EBPSO-SVM	96.80	96.29	96.04	96.55	96.42
ERSA-SVM	95.27	95.01	95.52	94.76	95.14

Note:

EBPSO-SVM (Enhanced Binary Particle Swarm Optimisation-Support Vector Machine), ERSA-SVM (Enhanced Reptile Search Algorithm-Support Vector Machine), FPR (False Positive Rate), CBBD (Cassava Bacterial Blight Disease), CBSD (Cassava Brown Streak Disease), CGMD (Cassava Green Mottle or Mite Disease), CMD (Cassava Mosaic Disease)

B. Performance Evaluation Metrics Comparison of Developed Enhanced Support Vector Machine Learning Models (EBPSO-SVM and ERSA-SVM) with the Existing Machine Learning Models in Plant Diseases Classification

Table 3 shows that the enhanced support vector machine learning models developed (EBPSO-SVM and ERSA-SVM) outperform several existing state-of-the-art machine learning models and deep learning models. For instance, the developed models outperform BPSO-SVM and RSA-SVM machine learning models developed by Ayoade [20]. Also, the developed models outperform the HC (CNN+LSTM), ResNet50V2+ResNet50, RIPEA+EfficientNet, and IRV-WSA+ETLNet deep learning models developed by Salini *et al.* [34], Kiruthika *et al.* [2], Srivathsan *et al.* [35], and Naveenkumar & Nandapogal [36], respectively, in terms of performance evaluation metrics used in their studies. Additionally, the developed models outperform the InceptionV3+DenseNet-BC-121-32+Xception and ResNet50V2+DenseNet-BC-121-32 deep learning models developed by Kiruthika *et al*. [2] in terms of sensitivity and precision performance metrics, except for specificity. Finally, the BPSO-RSA-SVM hybrid machine learning model developed by Ayoade [20] outperforms the developed model in terms of all the performance evaluation metrics used in this study. Similarly, the VGG-16+NAsNet, 2D CNN+SVM, and CNN-Stacking deep learning models developed by Pandiyaraju *et al*. [37], Prince *et al.* [38], and Khan *et al.* [39], respectively, were evaluated in terms of sensitivity, precision, and accuracy performance metrics.

Table 3. Performance Evaluation Metrics Comparison of Developed Enhanced Machine Learning Models (EBPSO-SVM and ERSA-SVM) with the Existing Machine Learning Models in Plant Diseases Classification

Author(s) and Models	False Positive Rate (%)	Specificity (%)	Sensitivity (%)	Precision (%)	Accuracy (%)
Pandiyaraju *et al.* [37]					
"VGG-16+NAsNet"	-	-	98.60	97.90	98.70
Prince *et al.* [38]					
"CNN-SVM"	-	-	99.00	99.00	99.09
Khan *et al.* [39]					
"CNN-Stacking"	-	-	98.53	98.53	98.53
Salini *et al.* [34]					
"HC (CNN+LSTM)"	6.97	93.03	93.30	93.59	93.17
Kiruthika *et al.* [2]					
"InceptionV3+DenseNet-BC-121-32+Xception"	-	97.14	88.56	88.53	-
"ResNet50V2+DenseNet-BC-121-32"	-	96.28	85.13	85.65	-
"ResNet50V2+ResNet50"	-	95.60	82.02	81.78	-
Srivathsan *et al.* [35]					
"RIPEA+EfficientNet"	-	-	87.18	88.18	93.06
Naveenkumar and Nandagopal [36]					
"IRV-WSA-ETLNet"	-	94.89	94.75	-	94.85
Ayoade [20]					
BPSO-SVM	5.22	94.78	95.82	96.44	95.40
RSA-SVM	7.12	92.88	94.53	95.14	93.86
BPSO-RSA-SVM	3.64	96.36	96.89	97.52	96.68
Developed Models					
EBPSO-SVM	**3.95**	**96.05**	**96.68**	**97.30**	**96.42**
ERSA-SVM	**5.54**	**94.46**	**95.60**	**96.22**	**95.14**

Note

VGG-16+NAsNet (Visual Geometry Group-16+Neural Architecture Search Network), CNN-SVM (Convolutional Neural Network-Support Vector Machine), HC (CNN+LSTM) (Hybrid Classifier (Convolutional Neural Network+Long Short-Term Memory), IRV-WSA-ETLNet (Improved Random Variable-Based Water Strider Algorithm-Efficient Transfer Learning Network), RIPEA (Residual Inception Positional Encoding Attention)

C. Discussion of the Findings

The reason why the Enhanced Binary Particle Swarm Optimisation-Support Vector Machine (EBPSO-SVM) outperform the Enhanced Reptile Search Algorithm - Support Vector Machine (ERSA-SVM) in terms of all the performance evaluation metrics used in this study as shown in Table 2 may be supported by the No Free Lunch (NFL) theorem which states that an algorithm's performance in one

problem category does not imply that it will perform well in other categories [40]. As a result, the efficacy and superiority of the (EBPSO-SVM) model over the (ERSA-SVM) in this study are problem-dependent.

Furthermore, it was observed that the developed models outperform the Binary Particle Swarm Optimisation-Support Vector Machine (BPSO-SVM) and Reptile Search Algorithm -Support Vector Machine (RSA-SVM) in terms of all the performance evaluation metrics used in this study, as shown in Table 3. This result clearly shows that enhancing the Binary Particle Swarm Optimisation (BPSO) with Chaotic Map, Arc Tangent Acceleration Coefficient, and Cosine Map Inertia Weight improves its performance. Similarly, enhancing the Reptile Search Algorithm with the Dynamic Evolutionary Sense, Prey Approaching Strategy, and Cauchy Mutation Strategy improves its performance. Enhancements to these algorithms generally improve their performance, resulting in more accurate predictions, better generalisation, and increased robustness.

According to Alzubaidi *et al*. [41], "Deep Learning models have been demonstrated to outperform popular machine learning techniques in several fields, including cybersecurity, natural language processing, bioinformatics, robotics and control, and medical information processing. This assertion is supported by the performance of VGG-16+NAsNet, 2D CNN+SVM, and CNN-Stacking developed by Pandiyaraju *et al.* [37], Prince *et al.* [38], and Khan *et al.* [39], which outperforms the proposed models in terms of sensitivity, precision, and accuracy, while InceptionV3+DenseNet-BC-121-32+Xception, and ResNet50V2+DenseNet-BC-121-32 developed by Kiruthika *et al.* [2] outperform the developed models in terms of specificity. This finding demonstrates that deep learning models outperform machine learning models, especially for complex tasks and large datasets, due to their ability to automatically learn hierarchical features and adapt to unstructured data.

However, in this study, it was discovered that the proposed models (EBPSO-SVM and ERSA-SVM) outperform some deep learning models such as RIPEA+EfficientNet, and IRV-WSA+ETLNet, developed by Srivathsan *et al.* [35], and Naveenkumar & Nandagopal [36] in terms of sensitivity, precision and accuracy, HC (CNN+LSTM) model developed by Salini *et al.* [34] in terms of all the performance evaluation metrics used in the developed models, ResNet50V2+ResNet50 developed by Kiruthika *et al.* [2], InceptionV3+DenseNet-BC-121-32+Xception, and ResNet50V2+DenseNet-BC-121-32 developed by Kiruthika *et al.* [2] in terms of the sensitivity and precision. This could be due to the quality of the dataset used in their studies [42], insufficient training data [43], improperly selected learning parameters [44], or an overly complex deep learning model setup [43-44].

Finally, it was discovered that the hybrid machine learning model (BPSO-RSA-SVM) developed by Ayoade [20] outperforms the developed models despite the improvement on both the BPSO and RSA with some strategies to resolve issues affecting their performance in the optimisation problem. This could be that individual optimisation techniques, such as Binary Particle Swarm Optimisation (BPSO) and Reptile Search Algorithm (RSA) or those improved by some strategies, such as Chaotic maps, Evolutionary sense, Cosine map inertia weight, and Cauchy mutation, may not fully address the challenges of SVM robustness in complex optimisation problems such as plant disease classification. However, Individual algorithms, or those improved by any of the aforementioned strategies, can increase the diversity of initial solutions and potentially improve exploration in algorithms such as RSA, but they do not provide the same level of exploitation as RSA and BPSO combined. Hybridisation of two or more optimisation techniques allows for a more comprehensive search of the parameter space, resulting in a balance of exploitation and exploration search behaviours, as well as a potentially more robust solution for scaling SVM input features, resolving SVM challenges of hyperparameter tuning, datasets with noise, outliers, and overlapping classes than using a single algorithm with or without these strategies.

IV. CONCLUSION AND FUTURE WORK

The study focuses on the application of image processing and machine learning techniques to detect and classify cassava leaf diseases. A support vector machine learning model based on enhanced Binary Particle Swarm Optimisation (EBPSO-SVM) and Enhanced Reptile Search Algorithm (ERSA-SVM) for cassava leaf diseases detection and classification is described. The digital image processing using EBPSO-SVM and ERSA-SVM follows predefined guidelines that include image dataset acquisition, preprocessing, segmentation, feature extraction, representation, interpretation, detection, and classification. The constructed enhanced machine learning models are evaluated for robustness using measures such as false positive rate, specificity, sensitivity, precision, accuracy, and computation time. The experimental results demonstrate that this methodology has a high potential for mitigating the economic losses caused by cassava disease outbreaks. The enhancement of the Binary and Harri Particle Swarm Optimisation and Reptile Search Algorithm a Chaotic Map, Arc Tangent Acceleration Coefficient, and Cosine Map Inertia Weight, Dynamic Evolutionary Sense, Prey Approaching Strategy, and Cauchy Mutation Strategy, respectively, presents robust frameworks for detecting and classifying cassava diseases, offering a viable solution to efficient resource allocation, higher yields, hunger eradication, healthier food production, and, ultimately, sustainable agricultural practices and food security. The

EBPSO-SVM and ERSA-SVM classification models achieved commendable accuracy rates of 96.42% and 95.14%, respectively. This high accuracy rate demonstrates how well the proposed models work to correctly detect and classify cassava disease, allowing for timely intervention and management. Future research should focus on a hybrid model that combines multiple architectures to optimise performance and automate cassava disease detection and classification. This approach has the potential to reduce economic losses in agriculture by allowing for early intervention.

CONFLICTS OF INTEREST STATEMENT

On behalf of the authors whose names appeared above, I certify that we have NO affiliations with or involvement in any organisation or entity with any financial interest (such as honoraria, educational grants, participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

REFERENCES

1. V. Singh, and A, K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," Information Processing in Agriculture, vol. 4, no. 1, pp. 41-49, 2017
2. V. Kiruthika, S. Shoba, M. Sendil, K. Nagarajan, and D. Punetha, "Hybrid ensemble - deep transfer model for early cassava leaf disease classification," Heliyon, vol. 10, no. 36097, pp. 1-16, 2024
3. U. K. Lilhore, A. L. Imoize, C. Lee, S. Simaiya, S.K. Pani, N. Goyal, A. Kumar, and C. Li, "Enhanced convolutional neural network model for cassava leaf diases identification and classification," Mathematics, vol. 10, no.4, pp. 1-20, 2022.
4. D. O. Oyewola, E. G. Dada, S. Misra, and R. Damaševi˘cius, "Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing," Peer J Computer Science, vol. 7, pp. 1-15, 2021. https://pmc.ncbi.nlm.nih.gov/articles/PMC7959600/pdf/peerj-cs-07-352.pdf
5. S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks-based recognition of plant diseases by leaf image classification,", Computational. Intelligence and Neuroscience, vol. 2016, no. 3289801, pp. 1-11, 2016. https://pmc.ncbi.nlm.nih.gov/articles/PMC4934169/pdf/CIN2016-3289801.pdf
6. D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification", In 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp. 3642–3649, 2012.
7. C. Bock, G. Poole, P. Parker, and T. Gottwald, "Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging," Critical Reviews in Plant Science, vol. 29, no 2, pp. 59-107, 2010.
8. H. Li, J. Yang, G. Zhang, and B. Fan, "Probabilistic support vector machines for classification of noise affected data," Information Sciences, vol. 221, pp. 60-71, 2013.
9. W. Pannakkong, K. Thiwa-Anont, K. Singthong, P. Parthanadee, and J. Buddhakulsomsiri, "Hyperparameter tuning of machine learning algorithms using response surface methodology: A case study of ANN, SVM, and DBN," Hindawi Mathematical Problems in Engineering, vol. 2022, no. 8513719, pp. 1-17, 2022.
10. N. Yang, S. Li, J. Liu, and F. Bian, "Sensitivity of support vector machine classification to various training features," TELKOMNIKA Indonesian Journal of Electrical Engineering, vol. 12, no. 1, pp. 286-291, 2014.
11. K. Nivethithaa, and S. Vijayalakshmi, "Optimized svm model for maize and rice leaf disease detection," Data Acquisition and Processing, vol. 38, no. 2, pp. 3146-3159, 2023. https://sjcjycl.cn/article/view-2023/pdf/02_3146.pdf
12. V. P. Kour, and S. Arora, "Particle swarm optimization based support vector machine (P-SVM) for the segmentation and classification of plants," IEEE Access, vol. 7, pp. 29374-29385, 2019. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8664152
13. H. I. Ghazalli, and N. A. Roslan, "Bacterial blight and mosaic disease detection on cassava leaf using support vector machine," Journal of Islamic, Social, Economics and Development (JISED), vol. 7, no. 46, pp. 467-475, 2022. https://jised.com/PDF/JISED-2022-46-07-51.pdf
14. A. G. Gad, "Particle swarm optimization algorithm and its applications: A systematic review," Archives of Computational Methods in Engineering, vol. 29, no. 12, pp. 2531–2561, 2022. https://www.researchgate.net/publication/360057862_Particle_Swarm_Optimization_Algorithm_and_Its_Applications_A_Systematic_Review
15. D. Zhang, J. Liu, L. Jiang, G. Bu, R. Hu, and Y. Luo, "The improvement of v-shaped transfer

function of binary particle swarm optimisation," Advances in Swarm Intelligence, vol. 12145, pp. 202-211, 2020.

16. M. Isiet, and M. Gadala, "Sensitivity analysis of control parameters in particle swarm optimization," Journal of Computational Science, vol. 41, no. 1, pp. 1-13, 2020. https://dspace.adu.ac.ae/bitstream/handle/1/1850/gadala%20Sensitivity%20analysis%20of%20control%20parameters%20in%20particle%20swarmoptimization.pdf?sequence=1&isAllowed=y
17. Q. Yuan, Y. Zhang, X. Dai, and S. Zhang, "A modified reptile search algorithm for numerical optimisation problems," Computational Intelligence and Neuroscience, vol. 2022, no. 975200, pp.1-20, 2022.
18. . K. Kailasam, R. Nalliah, S. N. Muthusamy, and P. Manoharan, "MLBRSA: Multi-learning-based reptile search algorithm for global optimisation and software requirements prioritisation problems," Biomimetics (Basel), vol. 8, no. 8, pp. 1-49, 2023.
19. K. Padmavathi and K. Thangadurai, "Implementation of RGB and gray scale images in plant leaves disease detection: Comparative study," Indian Journal of Science and Technology, vol. 9, no. 6, pp. 1-6. 2016. https://indjst.org/downloadarticle.php?Article_Unique_Id=INDJST5373&Full_TextPdf_Download=True
20. O. B. Ayoade, "Development of an optimised support vector machine for classification of cassava and maize diseases," A Ph.D. Thesis Submitted to the Department of Computer Sciences, Faculty of Natural Sciences, Ajayi Crowther University, Oyo, Nigeria. Unpublished Dissertation, 2025.
21. A. Aslam, E. Khan, and M. M. S. Beg, "Improved edge detection algorithm for brain tumour segmentation," Procedia Computer Science, vol. 58, pp. 430-437, 2015.
22. J. Gou, Y. X. Lei, W. P. Guo, C. Wang, Y. Q. Cai, and W. Luo, "A novel improved particle swarm optimization algorithm based on individual difference evolution," Applied Soft Computing Journal, vol. 57, pp. 468-481, 2017.
23. Y. Chen, L. Li, H. Peng, J. Xiao, and Q. Wu, "Dynamic multi-swarm differential learning particle swarm optimizer," Swarm Evolutionary Computation Journal, vol. 39, pp. 209-221, 2018.
24. N. Lynn, M. Z. Ali, and P. N. Suganthan, "Population topologies for particle swarm optimization and differential evolution," Swarm Evolutionary Computation Journal, vol. 39, pp. 24-35, 2018.
25. A. Khatami, S. Mirghasemi, A. Khosravi, C. P. Lim, and S. Nahavandi, "A new PSO-based approach to fire flame detection using K-Medoids clustering," Expert System Application. Vol. 68, pp. 69-80, 2017.
26. C. W. Lin, L. Yang, P. Fournier-Viger, T. P. Hong, and M. Voznak,"A binary PSO approach to mine high-utility itemsets." Soft computing, vol. 21, pp. 5103-5121, 2017.
27. Y. Zhou, N. Wang, and W. Xiang, "Clustering hierarchy protocol in wireless sensor networks using an improved PSO algorithm," IEEE Access, vol. 5, pp. 2241-2253, 2017.
28. N. Chouikhi, B. Ammar, N. Rokbani, and A. M. Alimi, "PSO-based analysis of Echo State Network parameters for time series forecasting," Applied Soft Computing, vol. 55, pp. 211-225, 2017.
29. Z. Ma, X. Yuan, S. Han, D. Sun, and Y. Ma, "Improved chaotic particle swarm optimisation algorithm with more symmetric distribution for numerical function optimisation," Symmetry, vol. 11, no. 876, pp, 1-19, 2019.
30. M. K. Khan, M. H. Zafar, S. Rashid, M. Mansoor, S. K. Raza Moosavi, and F. Sanfilippo, "Improved reptile search optimization algorithm: Application on Regression and classification problems," Applied Sciences, vol. 13, no. 945, pp. 1-29, 2023. https://www.mdpi.com/2076-3417/13/2/945
31. L. Zhou, X. Liu, R. Tian, W. Wang, and G. Jin, "A multi-strategy enhanced reptile search algorithm for global optimisation and engineering optimization design problems," Cluster Computing, vol. 28, no. 141, pp. 1-41, 2025.
32. Y. Fu, D. Liu, J. Chen, and L. He, "Secretary bird optimisation algorithm: a new metaheuristic for solving global optimisation problems," Artificial Intelligence Review, vol. 57, no. 123, pp. 1-102, 2024. https://link.springer.com/content/pdf/10.1007/s10462-024-10729-y.pdf
33. S. Zhao, T. Zhang, S. Ma, and M. Chen, "Dandelion optimizer: a nature-inspired metaheuristic algorithm for engineering applications," Engineering Application of Artificial Intelligence, vol. 114, no. 2, pp. 1-28, 2022.
34. . Salini, G. C. P. Latha, and R. Khilar, "Deep hybrid classification model for leaf disease classification of underground crops," Web Intelligence, vol. 22, no. 3, pp. 443-465, 2024.
35. M. S. Srivathsan, S. A. Jenish, K. Arvindhan, and R. Karthik, "An explainable hybrid feature aggregation network with residual inception positional encoding attention and EfficientNet for

cassava leaf disease classification," Scientific Reports, vol. 15, no. 11750, pp. 1-16, 2025.

36. M. Naveenkumar, and S. Nandagopal, "Adaptive hybrid segmentation,combined with meta heuristic optimization in transfer learning for plant leaf disease classification," Scientific Reports, vol. 15, no. 9838, pp. 1-30, 2025. https://www.nature.com/articles/s41598-025-93225-9.pdf
37. V. Pandiyaraju, A. M. S. Kumar, J. I. R. Praveen, S. Venkatraman, S. P. Kumar, S. A. Aravintakshan, A. Abeshek, and A. Kannan, "Improved tomato leaf disease classification through adaptive ensemble models with exponential moving average fusion and enhanced weighted gradient optimization," Frontiers in Plant Science, vol. 15, no. 1382416, pp. 1-26, 2024. https://pmc.ncbi.nlm.nih.gov/articles/PMC11140105/pdf/fpls-15-1382416.pdf
38. R. H. Prince, A. A. Mamun, H. I. Peyal, S. Miraz, M. Nahiduzzaman, A. Khandakar, and M. A. Ayari, "CSXAI: A lightweight 2D CNNSVM model for detection and classification of various crop diseases with explainable AI visualization," Frontiers in Plant Science, vol. 15, no. 1412988, pp. 1-18, 2024. https://pmc.ncbi.nlm.nih.gov/articles/PMC11257924/pdf/fpls-15-1412988.pdf
39. B. Khan, S. Das, N. S. Fahim, S. Banerjee, S. Khan, M. K. Al-Sadoon, H. S. Al-Otaibi, A. Reza, and M. T. Islam, "Bayesian optimized multimodal deep hybrid learning approach for tomato leaf disease classification," Scientific Reports, vol. 14, no. 21525, pp. 1-30, 2024. https://pmc.ncbi.nlm.nih.gov/articles/PMC11401875/pdf/41598_2024_Article_72237.pdf
40. X. S. Yang, "Free lunch or no free lunch: that is not just a question, "International Journal of Artificial Intelligence Tools, vol. 21, no. 3, pp. 1-13, 2012.
41. L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaria, and M. A. Fadhel, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," Journal of Big Data, vol. 8, no. 53, pp. 1-74, 2021. https://journalofbigdata.springeropen.com/counter/pdf/10.1186/s40537-021-00444-8.pdf
42. P. Khang, "7 Common machine learning and deep learning mistakes and limitations to avoid," 2023. https://medium.com/@khang.pham.exxact/7-common-machine-learning-and-deep-learning-mistakes-and-limitations-to-avoid-1a19636631a7
43. Metana, Deep learning models for classification: A comprehensive guide, 2025. https://metana.io/blog/deep-learning-models-for-classification-a-comprehensive-guide/
44. S. A. Nossier, J. Wall, M. Moniri, C. Glackin, and N. Cannings, " An experimental analysis of deep learning architectures for supervised speech enhancement," Electronics, vol. 10, no. 17, pp. 1-32, 2020.