



OPEN

An optimized ensemble grey wolf-based pipeline for monkeypox diagnosis

Ahmed I. Saleh¹, Asmaa H. Rabie¹, Shima E. ElSayyad^{1,2}, Ali E. Takieldeem³ & Fahmi Khalifa^{4,5}✉

As the world recovered from the coronavirus, the emergence of the monkeypox virus signaled a potential new pandemic, highlighting the need for faster and more efficient diagnostic methods. This study introduces a hybrid architecture for automatic monkeypox diagnosis by leveraging a modified grey wolf optimization model for effective feature selection and weighting. Additionally, the system uses an ensemble of classifiers, incorporating confusion based voting scheme to combine salient data features. Evaluation on public data sets, at various of training samples percentages, showed that the proposed strategy achieves promising performance. Namely, the system yielded an overall accuracy of 98.91% with testing run time of 5.5 seconds, while using machine classifiers with small number of hyper-parameters. Additional experimental comparison reveals superior performance of the proposed system over literature approaches using various metrics. Statistical analysis also confirmed that the proposed AMDS outperformed other models after running 50 times. Finally, the generalizability of the proposed model is evaluated by testing its performance on external data sets for monkeypox and COVID-19. Our model achieved an overall diagnostic accuracy of 98.00% and 99.00% on external COVID and monkeypox data sets, respectively.

Keywords Grey Wolf, Monkeypox, Neural network, Ensemble classifier, Confusion-based voting

The world has recently faced numerous crises in the medical sector due to the emergence of previously unknown viral strains and genetic mutations in known viruses. Monkeypox, a member of the orthopoxvirus family, is a rare disease that has been spreading, particularly in Africa-the main source of the virus-and recently in other regions, particularly the UK and the USA^{1,2}. Diagnosing monkeypox presents challenges where symptoms overlapped with other diseases like smallpox, chickenpox, measles, and psoriasis³⁻⁷. These overlapping, non-specific symptoms make it difficult to identify true signs of the disease. In regions where monkeypox is less common, healthcare providers may be less familiar with the disease and its symptoms, leading to delayed diagnosis. Access to diagnostic testing may be limited in some areas, especially during outbreaks.

Because the clinical presentation of monkeypox closely mirrors that of smallpox, it is at risk of being misdiagnosed⁸. This underlines the need for precise diagnostic techniques that incorporate artificial intelligence (AI), powered by machine learning (ML), and data mining (DM). These advanced methods offer the potential for more accurate diagnoses, which in turn will help physicians make better treatment decisions. While current diagnostic techniques exist, they often rely on medical images of skin lesions that can be time-consuming and resource-intensive⁸⁻¹⁰. This research gap necessitates the development of novel, efficient, and accessible diagnostic tools. Generally, AI-based classification architectures use training data to develop ML predictive tools capable of diagnostic decision-making^{1,2}. Recent research has explored the potential of AI and optimization algorithms for monkeypox diagnosis. However, existing methods often face limitations, such as reliance on large datasets, sensitivity to image quality, and potential for misclassification. This research aims to address these gaps by developing a novel AI-based diagnostic approach that leverages advanced optimization techniques to improve accuracy, reduce computational cost, and enhance the early detection of monkeypox, ultimately aiding in timely and effective disease management.

¹Computers and Control Systems Engineering Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt. ²Communications and Computers Engineering Department, MISR Higher Institute for Engineering and Technology, Mansoura 35516, Egypt. ³Faculty of Artificial Intelligence, Delta University for Science and Technology, Gamasa 35712, Egypt. ⁴Electronics and Communication Engineering Department, Mansoura University, Mansoura 35516, Egypt. ⁵Department of Electrical and Computer Engineering, Morgan State University, Baltimore, MD 21251, USA. ✉email: fahmikhalfah@mans.edu.eg

Feature selection and weighing play a vital role in enhancing the efficiency of AI systems by identifying the most relevant features while discarding redundant ones^{11,12}. This process reduces the dimensionality of the dataset, enabling faster model training and improving performance. Traditional feature selection techniques such as wrappers, filters, and hybrid methods often struggle in high-dimensional spaces^{13,14}. To address these limitations, researchers have turned to evolutionary computation (EC) algorithms, inspired by natural behaviors like the social patterns of animals searching for food^{15–17}. Examples of such algorithms include genetic algorithm (GA)^{13,14}, particle swarm optimization¹⁸, multi-objective optimisation¹⁹, colony optimization^{20,21}, bat optimization, whale optimization²², and ant lion optimization²³. Additionally, various other optimization algorithms have been proposed recently for improved convergence speed and global search balance, such as multiobjective brown bear optimization²⁴, Hippopotamus Optimizer²⁵ and Aijun et al. hybrid algorithm²⁶. Among these, grey wolf optimization (GWO) has gained significant attention²⁷. In addition to feature selection and weighing, ensemble classification has emerged as a method to boost classification accuracy by leveraging the collective knowledge of multiple classifiers rather than relying on a single model^{28,29}. Typically, an ensemble of classifiers (EoC) combines the predictions of various base classifiers to reach a final decision. There are two types of ensemble classifiers: homogeneous (which uses the same type of classifiers) and heterogeneous (which uses different classifiers). This ensemble approach offers advantages such as improved prediction performance, increased robustness, and reduced prediction variance^{28,29}.

The main objective of this study is to introduce an efficient system for monkeypox diagnosis, which is termed as, accurate monkeypox diagnosis strategy (ADMS). The proposed pipeline is hybrid and integrates an EoC to enhance the accuracy of monkeypox diagnosis. The developed method aims to automatically analyze laboratory-based tests, enabling rapid and more accurate identification of the disease, thus improving patient outcomes and public health response to monkeypox outbreaks. The main contribution aspects of the presented work can be summarized as follows.

- Introducing a hybrid architecture for automatic monkeypox diagnosis (ADMS: Accurate monkeypox diagnosis strategy)
- Deploying a modified Gray Wolf optimization technique for feature selection and weighting.
- Ensembling the salient features extracted via hybrid approach that uses confusion-based voting (CBV).
- Achieving promising performance with a smaller number of parameters compared with alternative approaches. The rest of the paper is categorized into the following section. The *Introduction* section covers and outlines the problem, significance as well as the overall objectives of this work. Literature work for monkeypox diagnosis is summarized in the *Related Work* section. The *Methodology* section details the proposed learnable architecture and the modified GWO-based feature selection. It also covers the machine classifiers integrated in the learnable ensemble classification. This is followed by *experimental* results, outcomes, and detailed performance comparisons. The associated discussion are presented in the *Discussion* section and finally the findings and future research directions are given in the *Conclusions* section.

Related work

In literature, previous research work on medical diagnostic models reveal a variety of techniques used for monkeypox diagnosis. For example, early diagnosis for monkeypox patients was performed using a neuro-fuzzy model (NFM)², a hybrid model that integrates the benefits of both neural network and fuzzy inference techniques. The uncertainty handling is provided by fuzzy logic while the learning capability is achieved by neural network. Experimental results demonstrated that NFM outperformed other models but it did not depend on all input symptoms. Also, NFM did not apply feature selection method before using the diagnostic method to improve its results. Another study by Abdelhamid et al.³⁰ introduced two hybrid feature selection methods to improve the accuracy of monkeypox diagnosis model using artificial neural networks (ANNs). The first method combined PSO and al-biruni earth radius (BER), while the second method integrates sine cosine and BER optimization algorithm. The performance of these two algorithms outperformed other algorithms; however, it has been performed on small-scale dataset. A deep learning-based study utilizing MobileNetV2 architecture has been used to accurately diagnose monkeypox patients by Arora et al.³¹. Despite accurate diagnosis results, it depends only on one off-the-self CNN architectures. A hybrid model by Alharbi et al.³² includes feature extraction using GoogLeNet, feature selection using dipper throated optimization, and decision tree for monkeypox diagnosis. Despite their model outperformed other models by providing more accurate results, it utilized pertained architectures.

To boost monkeypox diagnosis, five deep architectures (VGG19, ResNet50, VGG16, EfficientNetB3, and MobileNetV2) have been evaluated by Jaradat et al.³³. Results proved that MobileNetV2 outperformed other models where it gave recall of 96%, F1-score of 98%, accuracy of 98.16%, and precision of 99%. However, these models pretrained on Image net, thus, it is suggested to conduct fine tuning to learn intricate features of the problem at hand. Another deep learning -based monkeypox model utilizing Harris Hawks optimizer has been introduced by Almutairi et al.³⁴. After extracting features using optimized deep learning, seven ML models (AdaBoost, gradient boosting, histogram gradient boosting, k-nearest neighbors, support vector machine, extra trees, and random forest) have been used to provide diagnosis. Although the benefits of using these models, it based on limited samples in dataset. Residual networks and SqueezeNet model was also used to diagnose monkeypox from measles, chickenpox and healthy patients³⁵. The model use standard camera to capture skin images from patients and run it against deep learning models and achieved an average accuracy of 91.19% for the monkeypox class. Despite promising performance, the experiments were conducted by utilizing transfer learning. Similarly, deep neural networks (DNNs) has been used to diagnose monkeypox using skin images by Sorayaie et al.³⁶. They tested seven DNN modules to identify Monkeypox for binary and multi-class classification. Their results showed that DenseNet module achieved the best performance (Accuracy = 97.63% and 95.18%,

for binary and four-class results, respectively. Additionally, they integrated explainable AI modules (i.e., LIME and Grad-Cam) to provide insights into the decision-making process. Although DenseNet offers significant advantages in terms of feature propagation and parameter efficiency, it also comes with certain limitations such as it has high memory consumption, computational complexity, and it is sensitive to hyperparameters. Another study by Hapsari et al.³⁷ used the optimized random forest algorithm that depends on using PSO for providing fast and accurate monkeypox diagnosis. Their model has been tested using three classes: monkeypox, health, and PulsarStar. Their mode, however, requires large training time and it is sensitive to parameter tuning. Yadav et al.⁹ used modified extreme gradient boosting to diagnose monkeypox cases. Their system integrated a statistical loss function and a feature selection method. Better performance against other models was reported, however, MXGBoost is a complex model that suffers from overfitting with large datasets.

Recently, various research work has been directed towards diagnosing monkeypox and other diseases based on DNA or genes dataset. For example Xia, et. al.³⁸ developed a meta-learning based alternating minimization technique for global losses reduction, and train an adaptive strategy to improve performance by replacing the handcrafted counterpart. Lin et al.³⁹ provided programmable macrophage vesicle for defending against monkeypox virus. This vesicle depends on bionic self-adjuvanting vaccine. The study by Su et al.⁴⁰ introduced a ML-based framework for colon cancer diagnosis and staging using bioinformatics analysis of extracted feature genes. Their study identified key biomarkers and integrated gene expression data with predictive models to enhance diagnostic accuracy. This approach demonstrates the potential for improving precision medicine in colon cancer. Huang et al.⁴¹ used self-paced learning strategy to select gene and classify phenotype. For gene selection, the suggested approach by Yaqoob et al.⁴² combined PSO with mutual information. The latter serves as an initial filter for locating genes that provide a wealth of information about cancer. In order to identify the ideal subset of genes for precise categorization, PSO refines this selection in the second stage. A recent review by Fan et al.⁴³ provided the development of functional probes for diagnosing and treating infectious diseases. The survey showed that the probes offer potential for advancing diagnostic accuracy and targeted therapies in infectious disease management.

Various research studies have been proposed in literature for monkeypox diagnosis with promising results, please see Ref.⁴⁴ for details. Most of the mentioned studies used classical ML and deep architecture which are effective at recognizing monkeypox, but they are also have limitations related to the use of off-the-shelf-architectures acombined with transfer learning. Additionally, the classification of most of the existing approaches are based on single classifiers, while ensemble-based methods did not investigate the effect of feature fusion. Some studies deployed advanced deep architectures (e.g., CNN and vision transformers), however, contrary to recent literature, solo CNN and ViT-based methods exhibit limited performance and require further development. This paper extends the existing work for monkeypox diagnosis and introduces a hybrid strategy that include optimized feature selection and weighing, and novel confusion-based weighted ensemble classification.

Materials and methods

As illustrated in Fig. 1, the proposed pipeline consists of multiple sequential phases: Reprocessing (i.e., feature extraction, selection, and weighting), and classification for the diagnosis using ensemble classification. Generally, medical diagnosing process depends on several types of patient-related features, such as blood test or specialized analyzes of specific body organs (e.g., the heart) and may also include medical scans (e.g., magnetic resonance imaging). Initially, the related features should be extracted from the input data of the case being processed. Then, the most relevant and effective features are selected. Finally, the selected features are weighted. Both feature selection and feature weighting utilize a new modified technique based on grey wolf optimization (MGWO). In the following subsections, the details of MGWO will be discussed. After that, the implemented feature selection and feature weighting techniques will be introduced.

Modified grey wolf optimization (MGWO)

Undoubtedly, GWO is one of the most important and famous bio-inspired optimization techniques, that simulates the behavior and strategies of wolves in hunting. GWO has demonstrated competitive performance across a wide range of applications, including engineering design, image processing, and ML. Many reasons

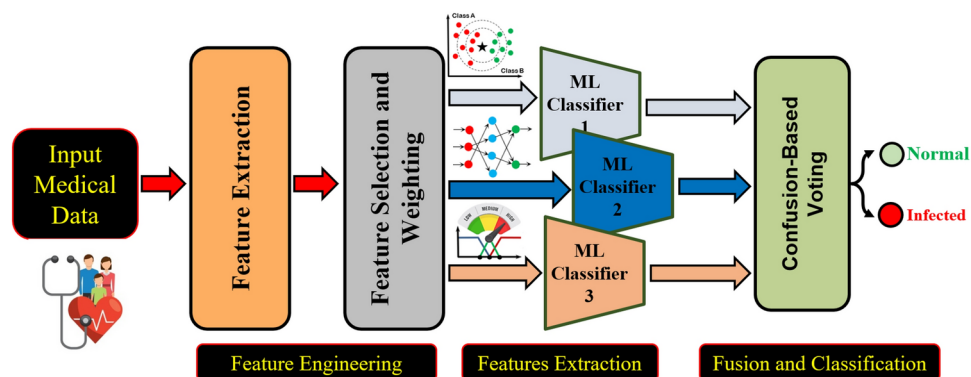


Fig. 1. Blockdiagram of the proposed pipeline for monkeypox diagnosing.

confirmed that GWO stands out as a robust and efficient optimization algorithm compared to other recent optimization algorithms, including its (i) simple structure and few control parameters make it easy to implement and adjust; (ii) strong exploration and exploitation capability, allowing it to effectively search for both global and local optima; and (iii) population-based nature enables it to handle complex optimization problems with multiple decision variables. These factors collectively contribute to the importance of GWO as a valuable tool for solving complex optimization challenges.

Generally, GWO predicts the prey's location based on the location of three leader wolves (Alpha, Beta, and Delta), denoted as; WL_α , WL_β , and WL_δ respectively. Although GWO assumes that WL_α is the closest wolf to the prey, followed by WL_β , and then WL_δ , it treats the leaders equally when locating the prey. It would be better to assign a weight to each of the three leaders based on their importance and then take the assigned weight into account when locating the prey. Hence, WL_α should be the closest to the prey, followed by WL_β , and then WL_δ . The weight of a specific wolf WL_i , where $i \in \{\alpha, \beta, \delta\}$, is computed as:

$$\text{Weight}(WL_i) = \frac{\xi(WL_i)}{\sum_{j \in \{\alpha, \beta, \delta\}} \xi(WL_j)} \quad , \text{ and } \quad \xi(WL_i) = \begin{cases} f(\vec{X}_i) & \text{if Target} = \text{maximize} \\ \frac{1}{f(\vec{X}_i)} & \text{if Target} = \text{minimize} \end{cases} \quad (1)$$

where $f(\vec{X}_i)$ is the objective function's value for the agent (wolf) WL_i whose location vector is \vec{X}_i . Based on the proposed MGWO, locating potential prey's position is accomplished through the following steps. Initially, as WL_α and WL_β are the nearest two agents to the prey, it is assumed that the prey lies between these two wolves, but it is closer to WL_α than to WL_β . Thus, the initial location of the prey $\vec{X}_{\text{Prey_in}}$ is a point between WL_α and WL_β . Locating $\vec{X}_{\text{Prey_in}}$ is achieved by dividing the distance between the points \vec{X}_α and \vec{X}_β based on the relative weight of WL_α and WL_β as presented in Fig. 2. Hence, as the weight of the wolf increases, it will be much closer to the prey and the initial prey position is expressed as follows:

$$\vec{X}_{\text{Prey_in}} = (x_1^{\text{Prey_in}}, x_2^{\text{Prey_in}}, x_3^{\text{Prey_in}}, \dots, x_n^{\text{Prey_in}}) \quad (2)$$

where $x_i^{\text{Prey_in}} = \frac{\omega_\alpha x_i^\alpha + \omega_\beta x_i^\beta}{\omega_\alpha + \omega_\beta}$. The distance between \vec{X}_α and $\vec{X}_{\text{Prey_in}}$, denoted as $D_\alpha^{\text{Prey_in}}$, is then calculated using as $D_\alpha^{\text{Prey_in}} = \sqrt{(x_1^\alpha - x_1^{\text{Prey_in}})^2 + (x_2^\alpha - x_2^{\text{Prey_in}})^2 + \dots + (x_n^\alpha - x_n^{\text{Prey_in}})^2}$.

To identify the prey's location, the next step is to add the effect of WL_δ , which is the third nearest wolf to the prey. The distance from \vec{X}_δ to $\vec{X}_{\text{Prey_in}}$, denoted as $D_\delta^{\text{Prey_in}}$, is calculated using Eq. (3):

$$D_\delta^{\text{Prey_in}} = \sqrt{(x_1^\delta - x_1^{\text{Prey_in}})^2 + (x_2^\delta - x_2^{\text{Prey_in}})^2 + \dots + (x_n^\delta - x_n^{\text{Prey_in}})^2} \quad (3)$$

The approximated distance between WL_δ and the actual prey can then be concluded as $D_\delta^{\text{Prey_Act}} = \frac{\omega_\delta D_\alpha^{\text{Prey_in}}}{\omega_\alpha}$.

Here, the prey is supposed to lie on the ray connecting the points $\vec{X}_{\text{Prey_in}}$ and \vec{X}_δ . Hence, there are two possibilities regarding the actual location of the prey, which is denoted as $\vec{X}_{\text{Prey_Act}}$. The first possibility is that $\vec{X}_{\text{Prey_Act}}$ is located between $\vec{X}_{\text{Prey_in}}$ and \vec{X}_δ . This occurs if $D_\delta^{\text{Prey_Act}} \leq D_\delta^{\text{Prey_in}}$. Hence, $\vec{X}_{\text{Prey_Act}}$ can be identified in the same manner as depicted in Fig. 2 and Eq. (4), as shown in Fig. 3a.

$$\vec{X}_{\text{Prey_Act}} = (x_1^{\text{Prey_Act}}, x_2^{\text{Prey_Act}}, x_3^{\text{Prey_Act}}, \dots, x_n^{\text{Prey_Act}}) \quad (4)$$

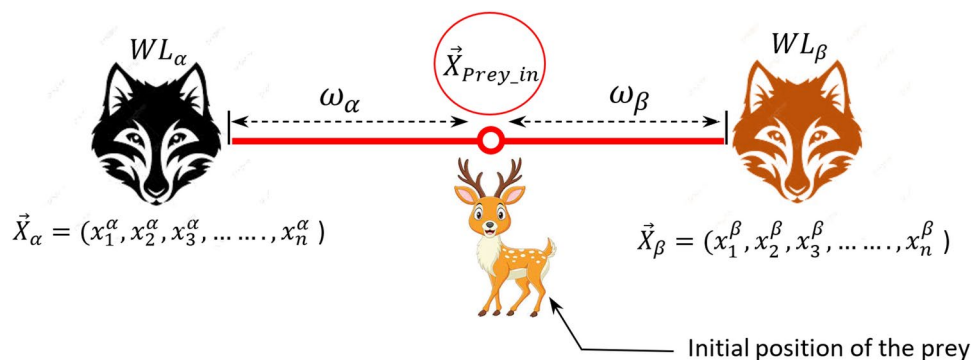


Fig. 2. Illustration of the initial position of the prey as a point between alpha and beta wolves.

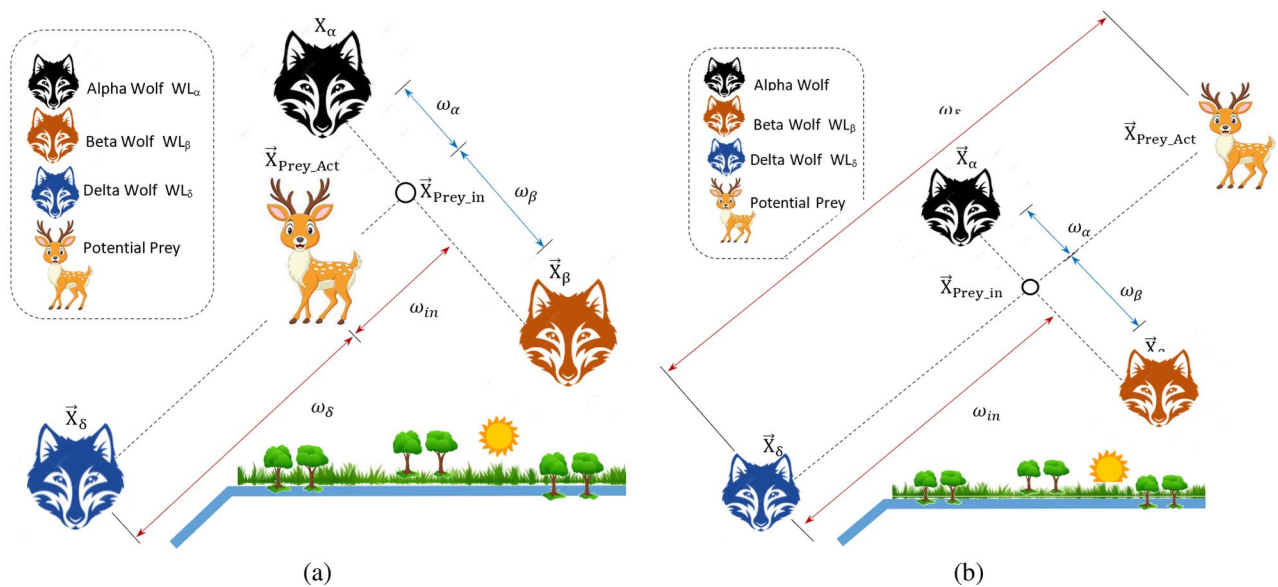


Fig. 3. Locating the prey in the modified grey wolf according to two cases when (a) $D_{\delta}^{(PreyAct)} \leq D_{\delta}^{(PreyIn)}$;

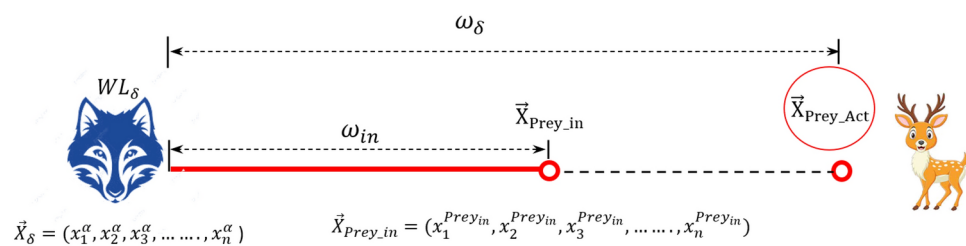


Fig. 4. Locating the prey $\vec{X}_{PreyAct}$ when $D_{\delta}^{(PreyAct)} > D_{\delta}^{(Preyin)}$.

where $x_i^{\text{Prey-Act}} = \frac{\omega_\delta x_i^{\text{Prey-in}} + \omega_{\text{in}} x_i^\delta}{\omega_\delta + \omega_{\text{in}}}$ and $\omega_{\text{in}} = \frac{\omega_\delta D_\alpha^{\text{Prey-in}}}{D_\delta^{\text{Prey-Act}}} - \omega_\delta$. The second possibility is that if $D_\delta^{\text{Prey-Act}} > D_\delta^{\text{Prey-in}}$, as presented in Fig. 4, the prey is located along the ray connecting $\vec{X}_{\text{Prey-in}}$ and \vec{X}_δ and in the direction of the point $\vec{X}_{\text{Prey-in}}$.

As illustrated in Fig. 4, $\vec{X}_{\text{P}_{\text{rev}} \text{ Act}}$ can be calculated using Eq. (5) and Fig. 3b.

$$\vec{X}_{\text{Prey_Act}} = (x_1^{\text{Prey_Act}}, x_2^{\text{Prey_Act}}, x_3^{\text{Prey_Act}}, \dots, x_n^{\text{Prey_Act}}) \quad (5)$$

where $x_i^{\text{Prey-Act}} = \frac{\omega_\delta x_i^{\text{Prey-in}} - \omega_{\text{in}} x_i^\delta}{\omega_\delta - \omega_{\text{in}}}$ and $\omega_{\text{in}} = \frac{\omega_\delta D_\alpha^{\text{Prey-in}}}{D_\delta^{\text{Prey-Act}}} - \omega_\delta$. Based on the prey's position, the location of the pack wolves can be modified using Eqs. (6) and (7).

$$\vec{D}_m = |\vec{C} \cdot \vec{X}_{\text{Prey Act}}(t) - \vec{X}_m(t)| \quad (6)$$

$$\begin{aligned}\bar{X}_m(t+1) &= \bar{X}_{\text{Prey_Act}}(t) - \vec{A} \cdot \vec{D}_m \\ a &= 2 \cdot \left(1 - \frac{t}{Z}\right) \\ \vec{A} &= 2 \cdot a \cdot \vec{r}_1 - a \\ \vec{C} &= 2 \cdot \vec{r}_2\end{aligned}\tag{7}$$

Where \vec{D}_m is the distance between the m th wolf (the m th solution) and the prey, $\vec{X}_{\text{Prey_Act}}(t)$ is the position vector of the prey, \vec{r}_1 and \vec{r}_2 are random vectors where $\vec{r}_1, \vec{r}_2 \in [0, 1]$, \vec{A} and \vec{C} are coefficient vectors, t is the iteration number, and Z is the total number of iterations. The complete description of the proposed MGWO is

depicted in Algorithm 1. The vector \vec{A} will be in the range $[-1, 1]$ during the exploitation phase, while it will be a vector of random values during the exploration phase (searching for prey). Initially, MGWO begins with a population that includes a set of random search agents (solutions). The positions of pack wolves are updated after each iteration. In fact, MGWO is a global optimization technique that uses adaptive variation of the search vector \vec{A} and can easily pass between exploration and exploitation. Additionally, MGWO contains little internal parameters to be modified.

Inputs:

- $WL = \{WL_1, \dots, WL_m\}$: Set of n available suggested solutions (search agents or Wolves).
- $F(X)$: Objective function to minimize or maximize.
- Z : Total number of iterations.
- m : Search Space dimensions.
- Search Space allowed bounds.

Outputs:

- \vec{X}_{optimal} : The best position of the search agents in the considered search space that verifies the target (minimize or maximize the objective function).

Steps:

1. Randomly distribute the search agents across the search space.

2. **For** $t = 1$ To Z :

$$- a = 2 \times \left(1 - \frac{t}{Z}\right)$$

- Calculate the objective function value for each wolf.

- Identify the leader wolves WL_α , WL_β , and WL_δ .

- Identify the positions of the leader wolves, \vec{X}_α , \vec{X}_β , \vec{X}_δ .

- Calculate the weight of each leader wolf $A \in \{\alpha, \beta, \delta\}$ as: $\text{Weight}(WL_A) = \frac{\xi(WL_A)}{\sum_{j \in \{\alpha, \beta, \delta\}} \xi(WL_j)}$

$$\text{Where: } \xi(WL_i) = \begin{cases} f(\vec{X}_i) & \text{if Target = maximize} \\ \frac{1}{f(\vec{X}_i)} & \text{if Target = minimize} \end{cases} \quad \forall i \in \{\alpha, \beta, \delta\}$$

- Calculate the initial position of the prey, expressed by the vector $\vec{X}_{\text{Prey_in}} = (x_1^{\text{Prey_in}}, x_2^{\text{Prey_in}}, \dots, x_n^{\text{Prey_in}})$ Where: $x_i^{\text{Prey_in}} = \frac{\omega_\delta x_i^\alpha + \omega_m x_i^\beta}{\omega_\delta + \omega_m}$

- Calculate the distance between \vec{X}_α and $\vec{X}_{\text{Prey_in}}$, $D_\alpha^{\text{Prey_in}} = \sqrt{(x_1^\alpha - x_1^{\text{Prey_in}})^2 + \dots + (x_n^\alpha - x_n^{\text{Prey_in}})^2}$

- Calculate the distance between \vec{X}_δ and $\vec{X}_{\text{Prey_in}}$, $D_\delta^{\text{Prey_in}} = \sqrt{(x_1^\delta - x_1^{\text{Prey_in}})^2 + \dots + (x_n^\delta - x_n^{\text{Prey_in}})^2}$

- Calculate the approximated distance between WL_δ and the actual prey: $D_\delta^{\text{Prey_Act}} = \frac{\omega_\delta D_\alpha^{\text{Prey_in}}}{\omega_\alpha}$

If $D_\delta^{\text{Prey_Act}} \leq D_\delta^{\text{Prey_in}}$ **THEN**

$$x_i^{\text{Prey_Act}} = \frac{\omega_\delta x_i^{\text{Prey_in}} + \omega_m x_i^\delta}{\omega_\delta + \omega_m}, \quad \omega_m = \frac{\omega_\delta D_\alpha^{\text{Prey_in}}}{D_\delta^{\text{Prey_Act}}} - \omega_\delta$$

ELSE

$$x_i^{\text{Prey_Act}} = \frac{\omega_\delta x_i^{\text{Prey_in}} - \omega_m x_i^\delta}{\omega_\delta - \omega_m}, \quad \omega_m = \frac{\omega_\delta D_\alpha^{\text{Prey_in}}}{D_\delta^{\text{Prey_Act}}} - \omega_\delta$$

EndIf

For each wolf $WL_m \in WL$:

$$- \vec{A} = 2 \times a \times \vec{r}_1 - a$$

$$- \vec{C} = 2 \times \vec{r}_2$$

$$- \vec{D}_m = \left| \vec{C} \cdot \vec{X}_{\text{Prey_Act}}(t) - \vec{X}_m(t) \right|$$

$$- \vec{X}_m(t+1) = \vec{X}_{\text{Prey_Act}}(t) - \vec{A} \cdot \vec{D}_m$$

If $F(\vec{X}_m(t+1))$ **is better than** $F(\vec{X}_m(t))$ **THEN**

Update the position of wolf WL_m

End If

Next

Next

The Optimal Solution $\vec{X}_{\text{optimal}} = \vec{X}_\alpha$ at the last iteration (iteration Z).

Parameters: n : No. of search agents; WL : The m^{th} search agent; WL : Set of available search agents; t : Iteration number; $F(X)$: Objective function; Z : Total number of iterations; WL_α , WL_β , and WL_δ : The Alpha, Beta and Delta wolf, respectively; X_α , X_β , X_δ : The position vectors of the leader wolves; r , r : Random vectors $\in [0, 1]$; D_m : The distance vector from the m^{th} agent to the prey; A, C : Coefficient vectors; $X_m(t)$: The position vector of the m^{th} agent at iteration t ; $X_{m(t+1)}$: The position vector of the m^{th} agent at iteration $t+1$; $X_{\text{PreyAct}}(t)$: The position vector of the prey at iteration t ; X_{optimal} : The optimal solution.

Algorithm 1. Modified gray wolf optimization (MGWO).

Through the modify position equations and allowing the vector \vec{A} to be in range $[-1, 1]$ in the prey encircling and attacking, high exploitation and convergence are achieved. Thus, during the iterations, high local optima avoidance and convergence speed are demonstrated by MGWO. \vec{A} decreases when the value of a decreases, hence, the behavior of pack wolves in MGWO is simulated. Generally, $\vec{A} \in [-a, a]$. Over the successive iterations, a decrease from 2 to 0. \vec{A} Value lies in the range $[-1, 1]$ during the exploitation. Accordingly, the new

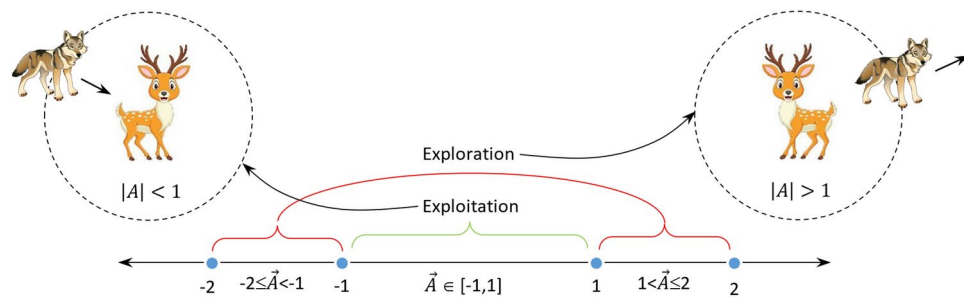


Fig. 5. Illustration of the value of \vec{A} in exploration and exploitation phases.

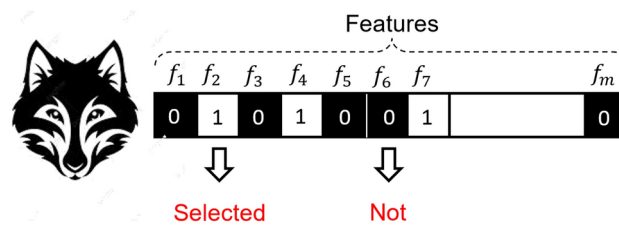


Fig. 6. A wolf representation in binary feature space.

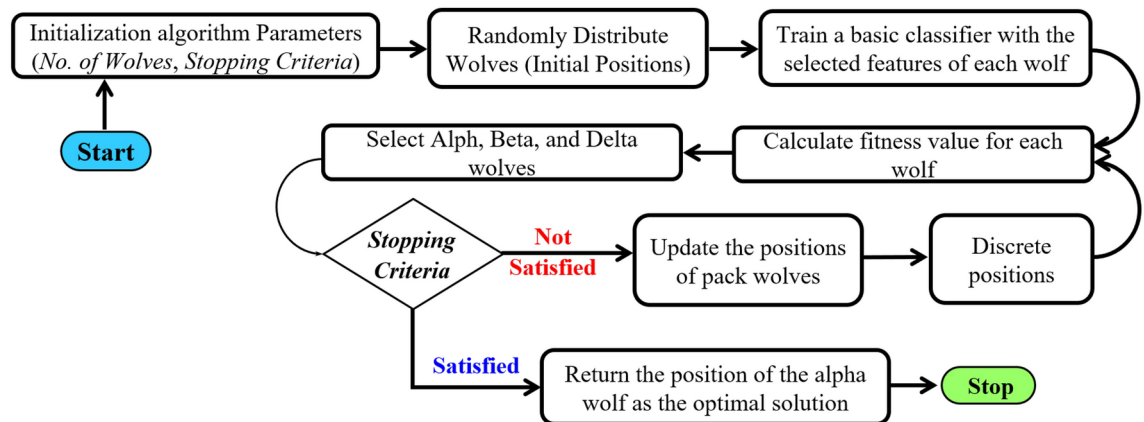


Fig. 7. A flow diagram illustrating the use of the proposed BMGWO in feature selection.

position of the MGWO will be between its current position and the position of the prey. On the other hand, \vec{A} value will be outside the range $[-1, 1]$ during exploration for allowing the search agent (wolf) to move far away from the prey for discovering new regions. Using this mechanism, MGWO can perform a global search and can get rid of local optima during exploration and exploitation, respectively. Figure 5 illustrates this concept.

Feature selection using binary MGWO

In spite of the effectiveness of the proposed MGWO, it cannot be applied directly to binary search space optimization problems. Like traditional GWO, MGWO updates the wolves' positions in a continuous manner, which is unsuitable to work in binary spaces. Here, a binary version of MGWO will be introduced, which is called binary MGWO (BMGWO). In m -dimensional binary feature space, each wolf is represented by a binary vector of m slots, which represents a set of selected features. Hence, the slots of zero value in the vector indicate that the corresponding features are not selected, otherwise, the corresponding feature is selected as illustrated in Fig. 6.

The proposed BMGWO will be used to select the feature set maximizing the accuracy of the used classifier, while keeping the selected features as minimum as possible. As illustrated in Fig. 7, initially, BMGWO randomly distribute the pack wolves across the m -dimensional binary feature space. The location of each wolf in the feature space is expressed by a binary vector, and represents a set of features. A basic classifier is trained for each set of

features presented by each wolf, then, the corresponding objective (fitness) function is calculated for each wolf. The employed objective function is illustrated in Eq. (8).

$$\text{Obj_Fn} = \lambda_1 \times \text{Acc} + \lambda_2 \times \frac{m - u}{m} \quad (8)$$

where Acc is the classification accuracy calculated for a basic classifier for the given wolf, m is the total number of features, u is the number of features selected by the wolf's vector (e.g., number of ones in the wolf's vector), λ_1 and λ_2 are weighting factors. As it is needed to maximize the fitness function, the alpha wolf is the one that presents the maximum objective function, the next best is beta wolf, then the third best is the delta wolf. Then, the location of the pack wolves are continuously updated across a set of sequential iterations based on the locations of the leader wolves.

After that, binary locations of wolves are concluded, an objective function (after retraining the employed classifier) is calculated again for each wolf based on its new updated binary location. The leader wolves (alpha, beta, and delta) are identified. These steps are repeated until the termination criteria is valid (either number of iterations finished or reaching a specific level of accuracy). Although there are several operators to transform the updated continuous positions of the pack wolves to binary, sigmoid function is the most popular one, which is used in this paper and described by Eq. (9)^{13,14}.

$$x_{zd}^b(t+1) = \begin{cases} 1 & \text{sigmoid}(X_{zd}) \geq \text{Rand} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Where b refers to binary, $x_{zd}^b(t+1)$ is the binary updated value for the z^{th} wolf in dimension d at iteration t , $x_{zd}(t+1)$ is the continuous updated value of the z^{th} wolf in dimension d , Rand is a random number $\in [0, 1]$. On the other hand, $\text{sigmoid}(x_{zd})$ can be described by $\text{sigmoid}(X_{zd}) = \frac{1}{1 - e^{-10(z_{zd} - 0.5)}}$.

Feature weighting using MGWO

Although NB is considered one of the most important and powerful classifiers, its performance may suffer from poor classification accuracy due to its reliance on two assumptions that may not align with reality. These problematic assumptions are (i) feature independence and (ii) equal weighting of features. To mitigate these primary drawbacks, feature weighting techniques are introduced to enhance the performance of the NB classifier, which can relax these harmful and unrealistic assumptions. Assume that it is necessary to classify a new case I expressed as $F = \{f_1, f_2, f_3, \dots, f_n\}$ where the target classes are represented as $C = \{c_1, c_2, c_3, \dots, c_m\}$. Then, NB can be used to calculate the probability that $I \in c_j$ using Eq. (10).

$$\text{Target}(I) = \arg \max_{c_j \in C} (c_j|F) = \arg \max_{c_j \in C} \left(\frac{P(F|c_j) \times P(c_j)}{P(F)} \right) \quad (10)$$

where $P(c_j|F)$ represents the conditional probability of c_j given F , $P(F|c_j)$ represents the conditional probability of F given c_j , $P(c_j)$ represents the prior probability of c_j , and c_j represents the j th class. In the case where features are independent, $P(F|c_j) = \prod_{i=1}^n P(f_i|c_j)$, which yields Eq. (11). In fact, the denominator in Eq. (11) can be neglected because it is constant for the input across all target classes.

$$\text{Target}(I) = \arg \max_{c_j \in C} \left(\frac{P(c_j) \times \prod_{i=1}^n P(f_i|c_j)}{\prod_{i=1}^n P(f_i)} \right) = \arg \max_{c_j \in C} \left(P(c_j) \times \prod_{i=1}^n P(f_i|c_j) \right) \quad (11)$$

In fact, equal weighting of features cannot satisfy the nature of applications in the real world. Hence, each feature has a different weight that indicates the importance of the feature. Unlike classical NB, each feature f_i has its weight w_i in Weighted Naïve Bayes (WNB). This weight can be a positive number as given in Eq. (12).

$$\text{Target}(I) = \arg \max_{c_j \in C} \left(P(c_j) \times \prod_{i=1}^n P(f_i|c_j)^{w_i} \right) \quad (12)$$

The majority of feature weighting methods can be divided into two main categories: filter methods and wrapper methods. The former is data driven since it determines the feature weights based on the overall properties of the data. Conversely, wrapper techniques are hypothesis driven since they determine the feature weights based on the performance feedback received from the classifier itself. Thus, filter techniques employ statistical methods to evaluate a set of characteristics, while wrapper approaches use cross-validation. Filter techniques may not always identify the optimal set of features, despite being significantly faster than wrapper techniques due to the lack of method training. Conversely, wrapper approaches always yield the optimal subset of characteristics but come at a high computational cost. Therefore, it is imperative to use novel strategies that provide efficient feature weighing with the least amount of computational time, as existing feature weighting approaches suffer from a number of issues, including instability and high computational cost. This section will offer a novel feature

weighting methodology that is based on the suggested MGWO algorithm, a recently developed bio-inspired optimization tool. Wolf-based feature weighting is the name of the suggested feature weighting scheme. It works on a similar idea as the wrapper, but instead of taking as long, it aims to cut the time by rapidly convergent to the optimal solution, which is the best possible combination of feature weights. Because it is based primarily on MGWO, which inherits the advantages of the basic GWO in terms of simplicity, low parameters, and high speed, WBFW is able to introduce the best solutions at a high speed. Additionally, it operates on the same principle as the wrapper technique, which guarantees finding the best available solutions.

The following demonstrates how MGWO can be used for feature weighting: first, a proposed space known as the feature weight space (FWS) is employed, and its dimensions correspond to the weights of the characteristics that are taken into consideration. For this reason, in the proposed FWS, the weight of each attribute is stated as a point. A feature's weight is represented by a positive real value, w_t , where $0 < w_t < 1$. There are h dimensions in FWS, which are labeled fw_1, fw_2, \dots, fw_h , assuming there are h selected features that require weighting. As a result, the weight of feature f_i , which can be stated numerically as wt_i , is represented by the axis fw_i . If not, wt_i is the weight of feature f_i on the FWS axis (dimension) fw_i . As a result, each wolf's location conveys a set of h weights that correspond to the features that are provided on separate weights in every dimension. $\vec{X}_m = (wt_{m1}, wt_{m2}, \dots, wt_{mh})$ if \vec{X}_m is the position vector of WL_m in FWS. Assuming that there are two selected features (i.e., $h = 2$), Fig. 8 provides an example. These features are designated as f_1 and f_2 . The weights of f_1 and f_2 , respectively, are represented by the dimensions of FWS, which is a two-dimensional space. Considering 5 wolves labeled WL_1, WL_2, WL_3, WL_4 , and WL_5 , which are represented by the position vectors $\vec{X}_1, \vec{X}_2, \vec{X}_3, \vec{X}_4$, and \vec{X}_5 , respectively.

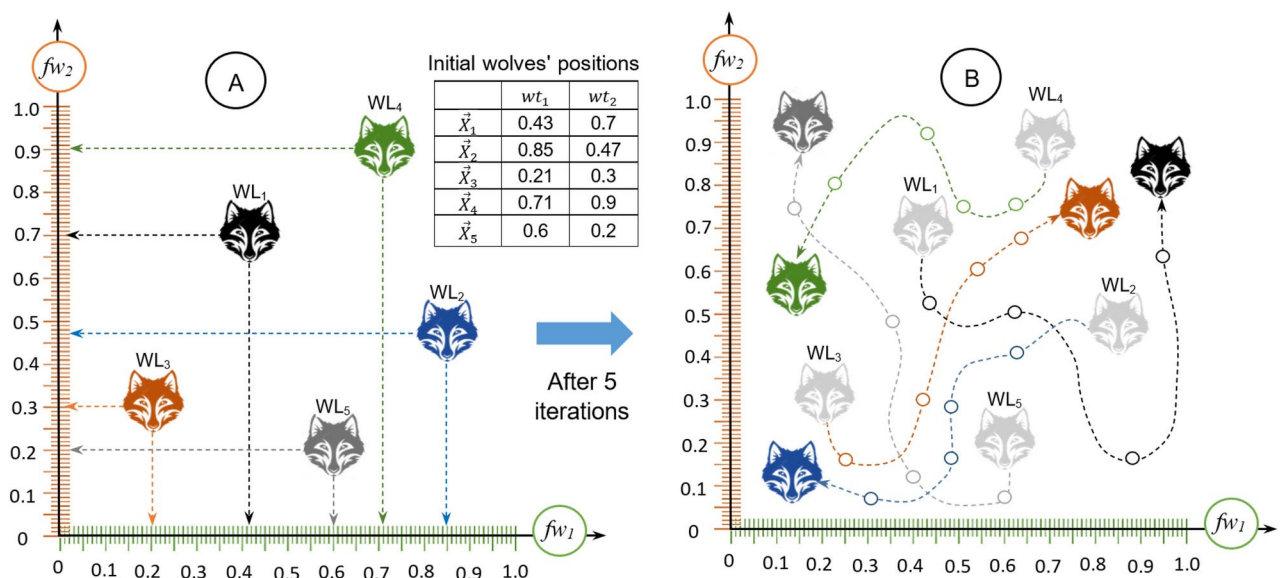


Fig. 8. The movement of wolves through the successive iterations to calculate the optimal feature weights: weights: (A) initial wolves' positions; and (B) wolves' positions after 5 iterations.

Inputs:

- $WL = \{WL_1, WL_2, WL_3, \dots, WL_n\}$: Set of n available suggested solutions or search agents (n Wolves).
- $Acc(X)$: Objective function to maximize, which is the accuracy of weighted Naïve Bayes classifier.
- Z : Total number of iterations.
- h : Search Space dimensions, also number of employed features.
- F : Set of employed features, where $F = \{f_1, f_2, f_3, \dots, f_h\}$
- Search Space allowed bounds, $0 < Weight(f_i) \leq 1$.

Outputs:

- $\vec{X}_{optimal}$: The best position of the search agents in FWS, which represents the best set of feature weights that maximizes the objective function (classification accuracy).

Steps:

- Train the model using the available dataset.
- Randomly distribute the search agents across the proposed Feature Weight Space (FWS).
- **For each wolf** $WL_i \in WL$ **Do**
 - Calculate the model accuracy using weighted NB based on the feature weights expressed by the current location of WL_i (e.g., $\vec{X}_i(t)$)
- Next**
- **For iteration_No = 1 to Z**
 - Identify the leader wolves WL_α , WL_β , and WL_δ
 - For each wolf** $WL_i \in WL$ **Do**
 - Update the position of WL_i based on the proposed MGWO, e.g., calculate $\vec{X}_i(t+1)$
 - Next**
 - For each wolf** $WL_i \in WL$ **Do**
 - * Calculate the model accuracy using weighted NB based on the feature weights expressed by the new location of WL_i (e.g., $\vec{X}_i(t+1)$)
 - * Perform Greedy selection, hence: **if** $Acc(\vec{X}_i(t)) > Acc(\vec{X}_i(t+1))$
 - Then** $\vec{X}_i(t+1) = \vec{X}_i(t)$
 - Next**
- Next**
- $\vec{X}_{optimal} = \vec{X}_\alpha$

Parameters, n: No. of search agents; WL_m : The m^{th} search agent; **WL:** Set of available search agents; t : Iteration number; $F(X)$: Objective function; Z : Total number of iterations; WL_α , WL_β , and WL_δ : The Alpha, Beta and Delta wolf, respectively; X_α , X_β , X_δ : The position vectors of the leader wolves; $X_{m(t)}$: The position vector of the m^{th} agent at iteration t ; $X_{m(t+1)}$: The position vector of the m^{th} agent at iteration $t+1$.

Algorithm 2. Wolf based feature weighting (WBFW).

The sequential processes for calculating the weights of the features used in the machine learning model are shown in Algorithm 2. The initial distribution of available wolves in the h -dimensional feature weight space is random, assuming h -selected features. In its starting posture, every wolf expresses a series of weights, each of which is associated with a distinct trait. The weighted Naïve Bayes (WNB) classifier's classification accuracy equation, represented in Eq. (10), serves as the considered objective function. After the model has been trained, the classification accuracy is calculated, taking into account the weights that each wolf has added. Next, WL_α , WL_β , and WL_δ -the three leader wolves-are recognized. The suggested MGWO is used to update the pack wolves' locations based on where the leader wolves are. Each wolf reflects one of the recommended solutions through a set of suggested weights for the accessible features. These weights are based on the new locations of the pack wolves. The objective function (classification accuracy of WNB) is computed for every proposed solution (wolf) without the requirement to retrain the model. After that, another iteration is initiated by identifying the leader wolves. Until the specified number of repetitions has been reached, this process is repeated. Finally, the collection of feature weights that the alpha wolf considers represents the optimal option.

The proposed WBFW is based on the proposed MGWO, which uses fewer computations than the traditional GWO with the minimum number of parameters, making it simple, fast, and accurate. Although it operates on a similar principle to the wrapper-based model, which continuously calculates the classification accuracy of the machine learning model, the proposed WBFW has several advantages over the traditional wrapper-based models. These advantages include (i) no need to retrain the model, as required by the traditional wrapper-based weighting techniques; (ii) unlike traditional wrapper-based techniques, the proposed WBFW does not need to make a complete scan to all available solutions, but rather can reach the optimal or at least a semi-optimal solution with the minimal number of attempts (iterations); and (iii) unlike traditional wrapper-based techniques, the proposed WBFW does not have to make a complete scan to all possible solutions.

Classification phase (CP)

Because it provides both a high degree of decision reliability and excellent classification efficiency, EoC has demonstrated high efficiency in numerous sectors. In this part, a novel instance of heterogeneous EoC is provided, based on three distinct types of base classifiers. Subsequently, the classifiers' decisions (outputs) are suitably amalgamated to generate the ultimate decision. Three classifiers are under consideration: (i) a deep learning based classifier (DLBC); (ii) a weighted naive bayes classifier (WNBC); and (iii) a Fuzzified distance-based classifier (FDBC). DLBC excels at learning complex patterns from large datasets, but it can be prone to overfitting and lacks interpretability. FDBC, on the other hand, is adept at handling uncertainty and imprecision, but it may struggle with large-scale datasets. Finally, WNBC provides a simple and efficient probabilistic classification approach, but it relies on the assumption of feature independence, which may not always hold

in real-world scenarios. By combining these techniques, the ensemble can mitigate the weaknesses of each individual model and capitalize on their complementary strengths. This results in a more robust, accurate, and interpretable classification model capable of handling diverse and complex datasets.

The *weighted Naïve bayes classifier (WNBC)* is the first classifier integrated in the proposed EoC pipeline. Numerous industries have benefited from NB's excellent classification effectiveness; the field of medical data mining (MDM), particularly in the area of disease diagnosis, is probably the most recent and significant⁴⁵. NB has proven to be an effective tool in helping physicians make decisions without hesitation, speeding up diagnosis, enhancing treatment quality, and reducing diagnostic errors. In general, NB has several benefits, including (i) being a straightforward technique that is simple to use; (ii) requiring less training data; (iii) NB's insensitivity to irrelevant attributes; and (vi) being scalable and quick, making it suitable for real-time prediction applications. As a result, NB has been selected to run the ensemble's second base classifier in this paper. The proposed MGWO is utilized to assign a weight for each of the selected features, thereby reducing the shortcomings of the traditional NB. Therefore, as the second proposed basis classifier, Wolf-based Weighted Naïve Bayes (WWNB) is developed. This new instance of WNB integrates the evidence from both the proposed MGWO and the standard WNB. The *second classifier* employed in the proposed pipeline is the long short-term memory (LSTM), which is a development of the recurrent neural network (RNN) to solve gradient vanishing and explosion difficulties^{46–48}. LSTM can be used for a wide range of real-time applications, including sequence-to-sequence predictions, language modeling, and medical diagnosis. This article handles multi-label diagnostics using a many-to-one LSTM structure, as seen in Fig. 9.

As seen in Fig. 9, the outputs of the i th LSTM are passed on to the $(i + 1)$ th LSTM, which is the following LSTM. To put it another way, the values of the " r " features in the input dataset are fed to the " r " LSTM cells, which in turn pass the cell state (c_i) and the current output state (h_i) of the i th LSTM as inputs to the $(i + 1)$ th LSTM, which finally provides the final diagnosis. The input, forget, and output gates make up each of the three gates that make up an LSTM cell, as shown to the right of Fig. 9. These gates serve to update the output value, maintain the cell state, and regulate the information flow across cell states. For each of the three gates, to accurately determine how to regulate the information flow, sigmoid activation (σ) is used. Information is added or deleted through each gate, but it remains unchanged in the cell state. The forget gate's primary function is to eliminate unnecessary data, whereas the input gate's primary function is to identify the input values needed to modify the cell state. The output gate can calculate output^{46–48}. The LSTM structure is constructed in three steps. The forget gate first recognizes undesired data and then removes it from the cell. Using the current input (f_i) and the previous output (h_{i-1}) in the cell state (c_{i-1}), the forget gate's output (x_i) can be given a value between zero and one. Zero indicates fully keeping the information, and one indicates forgetting it. Second, by multiplying its output (t_i) by the output of the tanh activation layer (\tilde{c}_i), the input gate decides whether to keep the data in the current cell state (c_i). Finally, the output gate combines its output (o_i) with the output of another tanh activation layer to produce the flow of a fraction of information (h_i) in the present cell state (c_i) at the output of the LSTM cell. Mathematical depiction of the governing equations can be found in Refs.^{46–48}.

Fuzzified distance-based classifier (FDBC)

In addition to WNBC and DLBC, a new fuzzy inference engine-based classifier is proposed and is integrated in the proposed system. The proposed FDBC is implemented through three consecutive processes, namely *inputs fuzzification*, *fuzzy rule induction*, and *defuzzification*. Four distinct fuzzy sets are taken into account in FDBC: (i) friend support (FS), (ii) number of friends (NF), (iii) average distance to friends (ADF), and (iv) distance to class centre (DCC). First, as *Definition 1* illustrates, let us define the concept of item friends.

Definition 1 *Item Friends* The friends of an input item I_j given a class c_m denoted as Friends (I_j, c_m) is a set of items whose distance to I_j less than or equal a critical distance denoted as DC_{rt} in the n -dimensional feature space and belong to c_m .

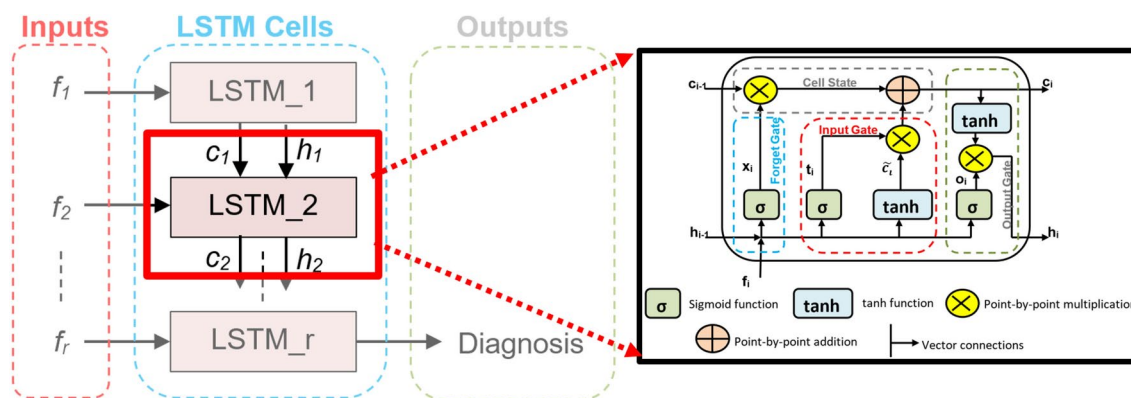


Fig. 9. Illustration of the multi-label many-to-one LSTM-based diagnostic structure and internal LSTM cell structure.

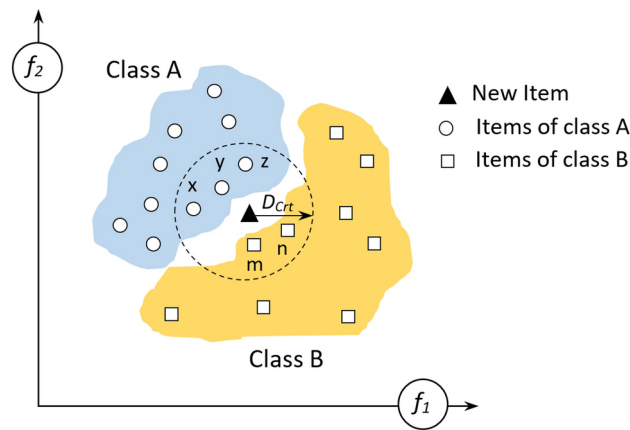


Fig. 10. The concept of item friends according to its distance to other items.

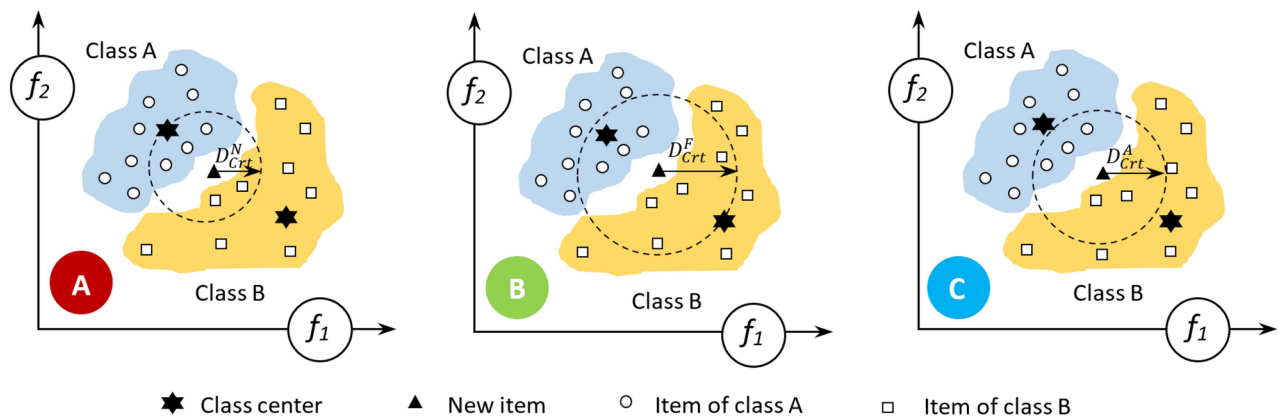


Fig. 11. Different methodologies for assigning the critical distance: (A) nearest class assignment; (B) furthest class assignment; and (C) average class assignment.

As an example, consider a two-dimensional feature space, as shown in Fig. 10, with two target classes, denoted by the set $C = \{c_1, c_2\}$, where f_1 and f_2 are the features under consideration. Within a circle of radius DC_{rt} , a new item I_j has two sets of friends: Friends $(I_j, c_1) = \{x, y, z\}$ and Friends $(I_j, c_2) = \{m, n\}$. It is observed that the item buddies are situated inside a circle in two-dimensional feature space and inside a ball with a radius of DC_{rt} in three-dimensional space. The sets of friends assigned to the new item are directly impacted, making it difficult to determine the appropriate value for DC_{rt} .

Three distinct approaches to DC_{rt} assignment are discussed here: (i) nearest class assignment (NCA), represented by the symbol DC_{rt}^N ; (ii) furthest class assignment (FCA), represented by the symbol DC_{rt}^F ; and (iii) average class assignment (ACA), represented by the symbol DC_{rt}^A . For example, DC_{rt}^N represents the distance from the item to the center of its nearest class, DC_{rt}^F represents the distance from the item to the center of its furthest class, and DC_{rt}^A represents the average distance between the item and the nearest and furthest classes, according to NCA. An example of a two-dimensional feature space with two target classes is presented in Fig. 11.

The DCC, which shows the distance between the new item I_j and the class center under examination, is the first fuzzy set that is taken into consideration. Given a class c_i in n -dimensional feature space with t samples,

the center of c_i can be identified as: $\text{Center}(c_i) = \left\{ \frac{\sum_{q=1}^t V_q^1}{t}, \frac{\sum_{q=1}^t V_q^2}{t}, \dots, \frac{\sum_{q=1}^t V_q^n}{t} \right\}$ Where t is the

number of examples within c_i , V_q^i is the value of the i th dimension of the q th example, and $\text{Center}(c_i)$ is the center of class c_i in the studied n -dimensional feature space. An accurate indicator of the degree of I_j 's affiliation

with class c_i is the Euclidean distance, $\text{DCC}(I_j, c_i): \text{Dis}(p_x, p_y) = \sqrt{\sum_{i=1}^n (p_x^i - p_y^i)^2}$ in the n -dimensional feature space, where p_x^i and p_y^i represent the i th dimension values of the points p_x and p_y , respectively. The NF of the new item I_j , which belongs to each of the considered classes, is the second fuzzy set that is taken into consideration to categorize the new item I_j into one of the target classes. The number of objects belonging to c_i within the distance DC_{rt} away from I_j is the number of friends of I_j that belong to class c_i , and is represented as $\text{NF}(I_j, c_i)$. As an example, look at Fig. 13a, where $\text{NF}(I_j, A) = 3$ and $\text{NF}(I_j, B) = 2$. In general, this supports I_j 's belonging to c_i as $\text{NF}(I_j, c_i)$ rises. The Average Distance to Friends (ADF) is the third fuzzy set

that is taken into account. The average distance from item I_j to its friends that belong to class c_i , denoted as $ADF(I_j, c_i)$, is defined as the average distance from I_j to all of its friends that belong to class c_i . Assuming $\text{Friends}(I_j, c_i) = \{\text{fr}_{i1}^j, \text{fr}_{i2}^j, \dots, \text{fr}_{ik}^j\}$, it is calculated by Eq. (13).

$$ADF(I_j, c_i) = \sum_{r=1}^k \frac{\text{Dis}(I_j, \text{fr}_{ir}^j)}{k} \quad (13)$$

where the number of I_j 's friends who are members of class c_i is denoted by $k = \text{NF}(I_j, c_i)$. The Euclidean distance between the new item I_j and the r th friend in class c_i is represented by the expression $\text{Dis}(I_j, \text{fr}_{ir}^j)$. Friends support is the fourth fuzzy set. Besides the other fuzzy sets, considering the number of friends without paying regard to the weight or strength of the friend wouldn't be correct. The strength of those friends also influences the item's classification, potentially increasing the possibility that a new element will be classified into one class rather than another, particularly if the number of friends of the new item is equal to that of two different classes. The strength of a friend expresses the belonging degree of the friend to the class it belongs to. Assuming a new item I_j whose r th friend fr_{xr}^j belongs to class c_x while three different target classes are available expressed by the set $C = \{c_x, c_y, c_z\}$. Generally, the strength of the friend fr_{xr}^j given the class c_m in which $m \neq r$, denoted as $\text{Strength}(\text{fr}_{xr}^j, c_m)_{m \neq r} = 0$. Hence, $\text{Strength}(\text{fr}_{xr}^j, c_y) = \text{Strength}(\text{fr}_{xr}^j, c_z) = 0$. On the other hand, the strength of the friend fr_{xr}^j belonging to class c_x is calculated as $\text{Strength}(\text{fr}_{xr}^j, c_x) = \left(\frac{1}{\text{Dis}(\text{fr}_{xr}^j, c_x)} \right)$. Based on their respective strengths, the group of friends of the new case I_j that are members of c_i support the claim that I_j is a member of c_i . Therefore, the probability that $I_j \in c_i$ increases as the strengths of the items $\text{Strength}(I_j, c_i)$ increase. The set $\text{Friends}(I_j, c_i) = \{\text{fr}_{i1}^j, \text{fr}_{i2}^j, \dots, \text{fr}_{ik}^j\}$ provides I_j with the support of belonging to c_i . This support is computed as $\text{FS}(I_j, c_i) = \left(\sum_{r=1}^k \text{Strength}(\text{fr}_{xr}^j, c_x) \right)$. The four fuzzy sets considered are clarified in Definitions 2–5.

Definition 2 Distance to class center (DCC) is the distance from the new item I_j to the center of the class under consideration in the m -dimensional feature space.

Definition 3 Number of friends (NF) of I_j given the class c_i , denoted as $\text{NF}(I_j, c_i)$, is the number of items that belong to c_i within the distance D_{Crt} away from I_j .

Definition 4 Average distance to friends (ADF) from item I_j to its friends given the class c_i , denoted as $ADF(I_j, c_i)$, is defined as the average distance from I_j to all of its friends that belong to c_i .

Definition 5 Friends Support (FS) that Friends(I_j, c_i) give to I_j , denoted as; $\text{FS}(I_j, c_i)$ is the sum of the strengths of all items $\in \text{Friends}(I_j, c_m)$.

Although fuzzy logic addresses the fuzziness in the data, it is not fuzzy in and of itself. The fuzzy membership function (FMF) can achieve this fuzziness in the data. The initial stage of any fuzzy inference system is

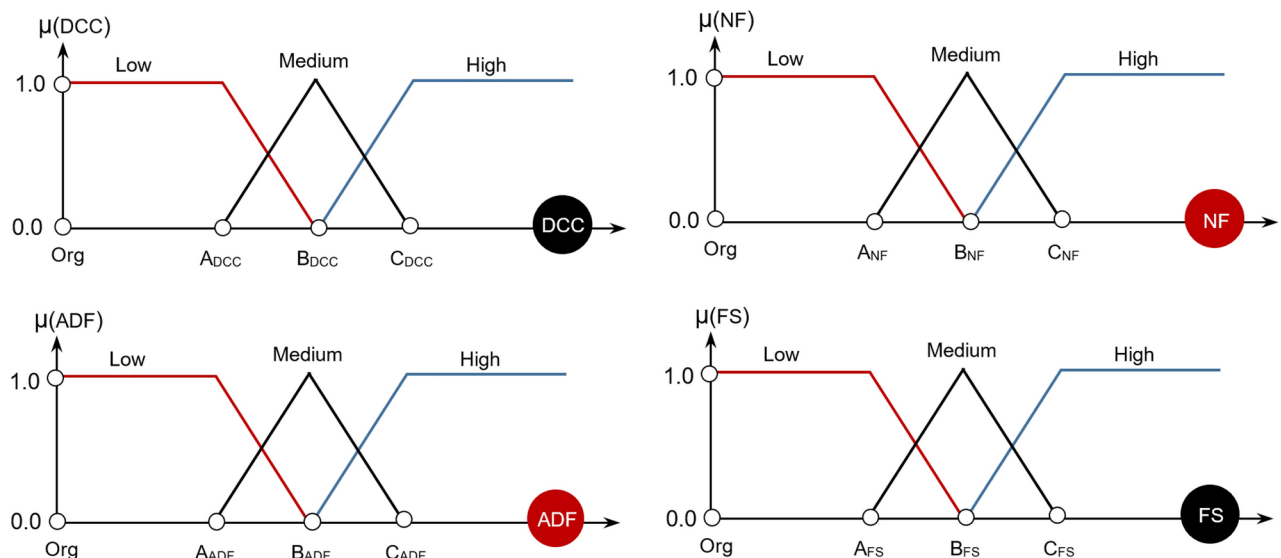


Fig. 12. The considered four input membership functions.

fuzzification. It is a procedure that uses a FMF of the related fuzzy set to convert the input crisp values into grades of membership for linguistic terms, “Low”, “Medium”, and “High” of the used fuzzy sets⁴⁹. For the considered four fuzzy sets, the employed membership functions are depicted in Fig. 12.

Setting the proper values of the membership parameters, A , B , and C , for the various input fuzzy sets is a challenging issue, as Fig. 12 illustrates. Initially, for A_{DCC} , B_{DCC} , and C_{DCC} , assume that $\text{Center}(c_n)$ and $\text{Center}(c_f)$ are the centers of the nearest and farthest classes to the input item I_j , respectively. Locating A_{DCC} , B_{DCC} , and C_{DCC} can be accomplished by following the next restrictions: (i) $A_{DCC} < B_{DCC} < C_{DCC}$, (ii) $A_{DCC} \geq \text{Dis}(\text{Center}(c_n), I_j)$, (iii) $\text{Center}(c_n)$ is the nearest class center to the input item I_j in the feature space, (iv) $C_{DCC} \leq \text{Dis}(\text{Center}(c_f), I_j)$, (v) where $\text{Center}(c_f)$ is the farthest class center to the input item I_j in the feature space, (vi) $B_{DCC} = \frac{A_{DCC} + C_{DCC}}{2}$.

In this paper, it is assumed that $A_{DCC} = \text{Dis}(\text{Center}(c_n), I_j)$ and $C_{DCC} = \text{Dis}(\text{Center}(c_f), I_j)$. For A_{NF} , B_{NF} , and C_{NF} , assume $\text{Friends}(I_j)$ to be the set of I_j 's friends that are located within a distance less than or equal to D_{Crt} from I_j in the considered feature space, while $\text{Friends}(I_j, c_i) = \text{Friends}(I_j) \cap \text{items}(c_i)$ is the number of I_j 's friends that belong to class c_i . Hence, $\text{Num_Friends}(I_j) = |\text{Friends}(I_j)|$ is the number of all friends associated with I_j , while $\text{Num_Friends}(I_j, c_i) = |\text{Friends}(I_j, c_i)|$. Let $\text{NF}_{\max}(I_j) = \max_{c_i \in C} [\text{Num_Friends}(I_j, c_i)]$, and $\text{NF}_{\min}(I_j) = \min_{c_i \in C} [\text{Num_Friends}(I_j, c_i)]$. Similarly, locating A_{NF} , B_{NF} , and C_{NF} can be accomplished by following the next restrictions: (i) $A_{NF} < B_{NF} < C_{NF}$, (ii) $A_{NF} \geq \text{NF}_{\min}(I_j)$, (iii) $C_{NF} \leq \text{NF}_{\max}(I_j)$, and (iv) $B_{NF} = \frac{A_{NF} + C_{NF}}{2}$. In this paper, it is assumed that $A_{NF} = \text{NF}_{\min}(I_j)$ and $C_{NF} = \text{NF}_{\max}(I_j)$.

The same procedure is followed to locate A_{ADF} , B_{ADF} , and C_{ADF} . Let $\text{ADF}_{\max}(I_j) = \max_{c_i \in C} [\text{ADF}(I_j, c_i)]$ and $\text{ADF}_{\min}(I_j) = \min_{c_i \in C} [\text{ADF}(I_j, c_i)]$. Consider the following restrictions: (i) $A_{ADF} < B_{ADF} < C_{ADF}$, (ii) $A_{ADF} \geq \text{ADF}_{\min}(I_j)$, (iii) $C_{ADF} \leq \text{ADF}_{\max}(I_j)$, (iv) $B_{ADF} = \frac{A_{ADF} + C_{ADF}}{2}$. In this paper, it is assumed that $A_{ADF} = \text{ADF}_{\min}(I_j)$ and $C_{ADF} = \text{ADF}_{\max}(I_j)$. Finally, for locating A_{FS} , B_{FS} , and C_{FS} , let $\text{FS}_{\max}(I_j) = \max_{c_i \in C} [\text{FS}(I_j, c_i)]$ and $\text{FS}_{\min}(I_j) = \min_{c_i \in C} [\text{FS}(I_j, c_i)]$. Consider the following restrictions: (i) $A_{FS} < B_{FS} < C_{FS}$, (ii) $A_{FS} \geq \text{FS}_{\min}(I_j)$, (iii) $C_{FS} \leq \text{FS}_{\max}(I_j)$, (vi) $B_{FS} = \frac{A_{FS} + C_{FS}}{2}$. In this paper, it is assumed that $A_{FS} = \text{FS}_{\min}(I_j)$ and $C_{FS} = \text{FS}_{\max}(I_j)$. After that, fuzzy rule induction will be used to conclude the corresponding output given the input variables via fuzzy rules using max-min⁴⁹. Finally, Defuzzification will be used using the Center of Gravity (COG) method to extract a single, sharp value from the output of the combined fuzzy sets⁴⁹.

Merging the predictions of EoC

Combining the predictions of EoC is a true challenge as it directly affects the final model decision. Majority voting (MV) and weighted MV (WMV) are the most commonly used methods. Considered the predicted class for a new item I of 10 classifiers as illustrated in Table 1, with the corresponding predictive accuracy of each classifier on its validation dataset during the ensemble creation process, expressed as a proportion from 0 to 1. If MV is used, class A gained 4 votes, class B gained 3 votes, and class C gained 3 votes. Hence, class A is the target class for the new item, even though only 4 out of 10 classifiers made that prediction. On the other hand, if WMV is used, class A gains $0.66 + 0.66 + 0.71 + 0.66 = 2.69$ votes. Also, B gains $0.91 + 0.86 + 0.96 = 2.73$ votes and C gains $0.71 + 0.91 + 0.81 = 2.43$ votes. Hence, the target class will be class B since it gained the votes of three of the best classifiers, judged by their performance on their validation datasets which vary from one classifier to another. On the other hand, A gained the votes of four relatively weak classifiers.

Although it seems reasonable that N classifiers ‘working together’ can give better predictive accuracy than a single classifier, EoC has several drawbacks such as; (i) prediction using EoC is slower than single classifier prediction, (ii) there is no guarantee that the performance of EoC is always better than the performance of a single classifier. However, the first drawback can be eliminated by using parallel ensemble classification (PEC), which is a relatively new research field⁵⁰. On the other hand, the second drawback of EoC can be eliminated by precisely generating the classifiers, accurately choosing the employed types of the classifiers, and putting strong rules for combining the predictions of the contributing classifiers.

This study presents a newly proposed method for merging ensemble classifiers, called confusion-based voting (CBV). Based on the validation dataset, the confusion matrices (CMs) of the three basic classifiers used in the ensemble. For example, Fig. 13 illustrates the CM representation for the three binary employed classifiers (FDBC, WNBC, and DLBC) separately depending on two class category (A and B). The Figure shows that FDBC, WNBC, and DLBC have general accuracy rates of 71%, 59%, and 3%, respectively. However, Table 2 shows the results of classifying a new object using the three base classifiers that were taken into consideration. Class B will be the target class if the majority vote holds, as it will receive two votes to Class A's one vote. On the other hand, class B increases $0.55 + 0.32 = 0.87$ in weight, while class A gains 0.92 based on CBV. Class A will therefore be the input item's target class.

	Classifier										Total
	1	2	3	4	5	6	7	8	9	10	
Accuracy	0.66	0.91	0.66	0.86	0.71	0.71	0.91	0.66	0.81	0.96	7.75
Predicted Class	A	B	A	B	A	C	C	A	C	B	

Table 1. Illustrative example of the ensemble of classifier (EoC) with predictive accuracy information.

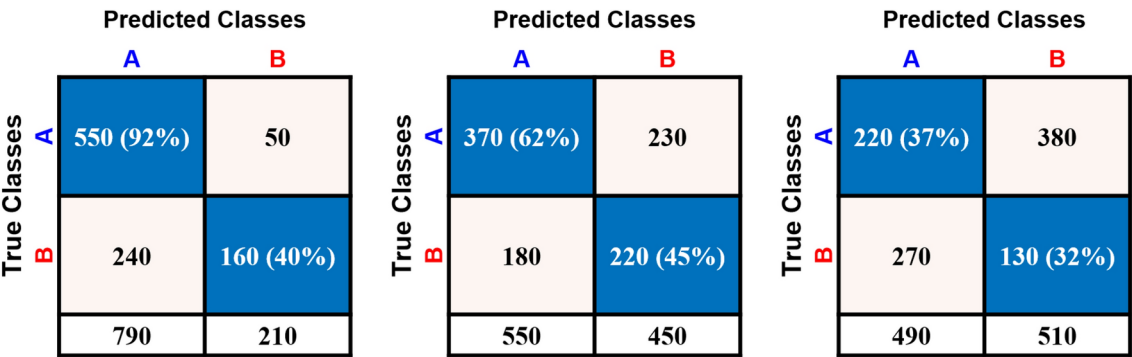


Fig. 13. Confusion matrices—from left to right—for the fuzzified distance-based classifier (FDBC), weighted Naïve bayes classifier (WNBC), and deep learning-based classifier (DLBC).

Classifier	Predicted class	Voted class	
		A	B
FDBC	A	0.92	0.40
WNBC	B	0.62	0.55
DLBC	B	0.37	0.32

Table 2. The ensemble of classifier (EoC) with confusion-based voting (CVB) using the three classifiers in Fig.13. Here, FDBC, WNBC, and DLBC stand for fuzzified distance-based classifier, weighted Naïve bayes classifier, and deep learning-based classifier, respectively. Bold values indicate the classified class for each classifier.

Experimental results

To diagnose patients with monkeypox, the proposed AMDS strategy with the proposed MGWO is used for feature selection and feature weighting. To give a more precise diagnosis, the informative features are passed to EoC, integrating FDBC, WNBC, and DLBC diagnostic techniques and utilizing CBV as a weighted voting method. Two scenarios will be used to implement and test AMDS. Firstly, the BMGWO will be tested and its outcomes compared with other current selection techniques and other of versions of GWO will be assessed. Secondly, the AMDS strategy will be implemented and evaluated on two datasets. The outcomes will be compared with those of other contemporary strategies. The effectiveness of the employed strategies is determined through the application of confusion matrix metrics^{13,14} and fractional dataset training is utilized. Namely, the proposed model is trained five different times using different percentage of the training data, selected randomly every time. Particularly, the system will be trained using 70, 140, 210, 280, and 350 samples of the training samples and accuracy is assessed each time using the performance metrics. This procedure evaluates the system sensitivity to dataset size and helps identify trends in performance. This approach also assesses data sufficiency, determining if additional samples significantly impact the overall performance. Furthermore, varying the data size explores bias-variance trade-offs, with smaller datasets exposing high variance and larger ones highlighting potential biases. This analysis provides practical insights into the minimum data required for effective training, particularly in scenarios where data collection is costly or challenging, ensuring model robustness. Typical values of the systems' parameters utilized for the two independent ransom numbers r_1 and r_2 : $0 \leq r_1, r_2 \leq 1$; the Uniform distribution value in BMGWO: $0 \leq \text{Rand}() \leq 1$; the maximum number of iterations for MGWO: $Z = 100$; and $a \in [2, 0]$. Additionally, the population's size (number of solutions) is 30 and the number of runs is 50. To implement the proposed model, MATLAB R2021b win64 running on a 2.4 GHZ Core i-9 Dell computer with Windows operating system, 8GB of RAM, and 1 TB hard drive.

This research uses the monkeypox dataset to confirm that the suggested AMDS approach works⁵¹. Patients in this dataset were categorized as "positive" or "negative". People between the dates of June 5, 2022, and September 19, 2022, provided data for the Monkeypox dataset, which was compiled online⁵¹. The Monkeypox dataset consists of blood test results gathered from various parts of various nations, including UK, Nigeria, Spain. In actuality, 500 cases of various ages and sexes were included in this dataset and assigned to the "positive" and "negative" classifications. This dataset has six class groups, based on which patients were categorized into numerous infectious diseases. These include alopecia, acne, psoriasis, monkeypox, smallpox, and normal, see Table 3. This dataset has 47 features, including both demographic and laboratory blood test features. These characteristic features are used to characterize each patient's status in the dataset based on blood tests. Following BMGWO implementation, 32 features were chosen. Figure 14 represents the importance of the selected features and their effects on diagnosis using the Shapley additive explanations (SHAP) method. Table 3 shows that 296 of the 500 cases in the sample are considered to be monkeypox "positive" cases. The monkeypox dataset has been divided into two subsets: 350 and 150 cases making up the training and testing dataset, respectively.

Criteria		Value/description		
Total # of cases		Monkeypox patients	Normal people	Other diseases
Other diseases		297	95	108
	Acne		Alopecia	Psoriasis
	18		9	15
	Small pox		Other	
	29		37	
Sex		Male	Female	
	Monkeypox	152	145	
	Normal	48	47	
	Others	57	51	

Table 3. Monkeypox dataset distribution based on infection.

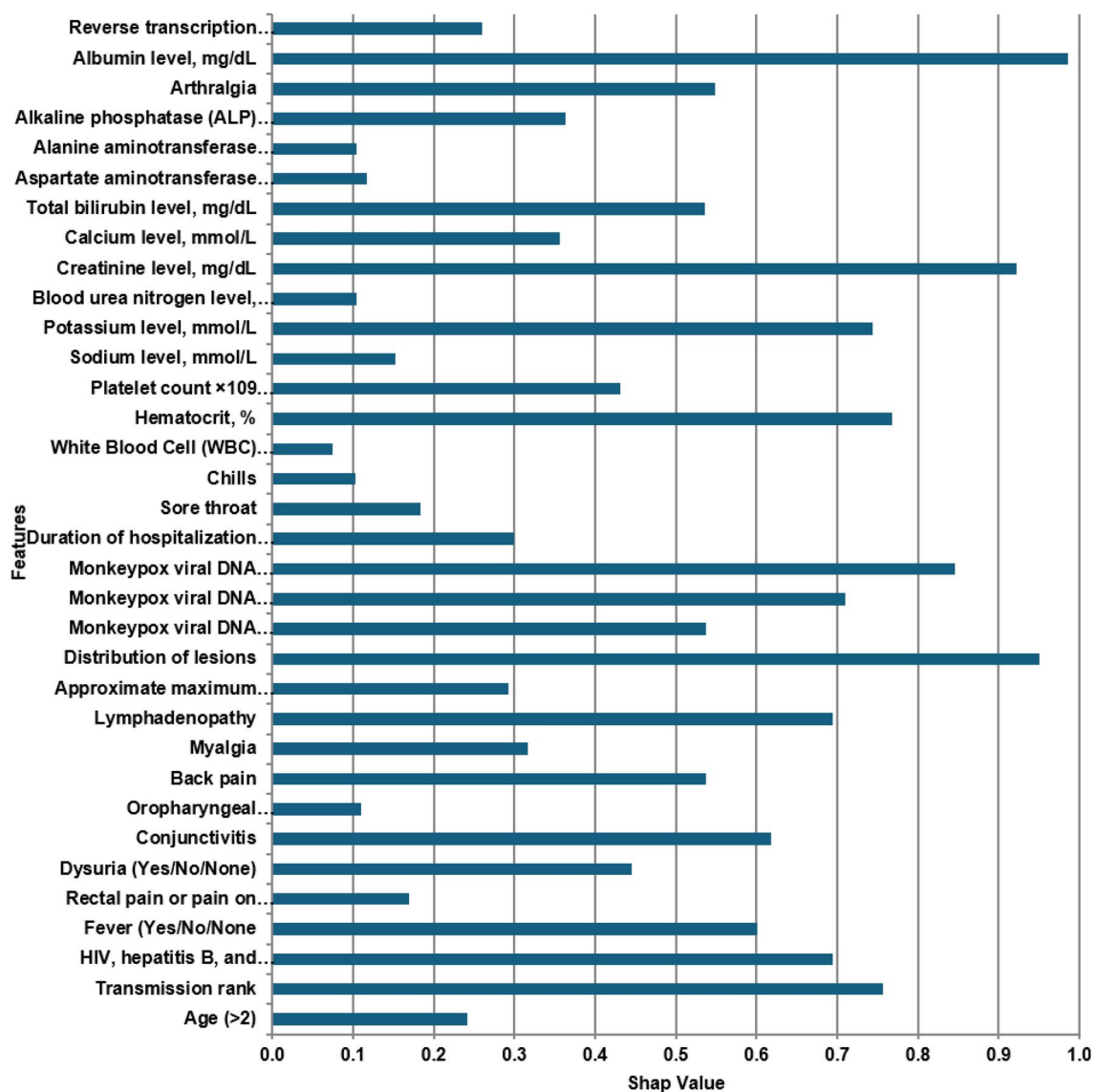


Fig. 14. Features importance emphasized using Shapley additive explanations (SHAP).

	Method			
Metric	Proposed	GWO ⁵²	Modified GW ₁ ⁵³	Modified GW ₂ ⁵⁴
Accuracy (%)	97.70	91.00	93.00	92.50
Precision(%)	90.01	85.02	88.00	87.23
Recall (%)	91.11	89.16	91.25	91.00
F1-Score (%)	93.0	94.00	93.50	93.05
AUC-ROC (%)	96.00	94.00	94.85	93.00

Table 4. Comparative accuracy of the proposed MGWO algorithm compared with enhanced versions of GWO algorithm.

At first the efficiency and of the proposed BMGWO feature selection algorithm is demonstrated by testing it against original and other improved versions of GWO proposed in Refs.^{52–54} using the above mentioned data set. The results of the comparisons is presented in Table 4. According to results, the proposed algorithm showed enhanced performance when compared with other versions of GWO based on the accuracy, precision, F1-score, and other metrics. The results also revealed that the modified GWO versions showed enhanced performance over the original algorithm.

Secondly, the proposed BMGWO has been compared against modern feature selection approaches. The latter include, GA²⁸, the enhanced genetic algorithm (EGA)¹⁴, the modified brain storm optimization (MBSO)⁵⁵, the hybrid selection method (HSM)¹³, and the binary GWO (BGWO) methods⁵². Following the implementation of these selection techniques, an NB classifier is created, employing the chosen set of features from each selection algorithm independently to be trained on a valid dataset empty of irrelevant features⁵⁶. The measurements of the performance metrics—precision, recall, accuracy, and F1-score—are shown in first column of Fig. 15. Additional evaluation is conducted using the receiver operating characteristic (ROC). This is demonstrated in Fig. 16. The results shows enhanced performance as compared with other methods.

BMGWO outperforms GA, EGA, MBSO, HSM, and BGWO. The method attains the highest accuracy, precision, recall, F1-score, and AUC-ROC, values—98.70%, 90.01%, 91.11%, 93.05%, and 96.12% respectively—at the maximum number of training data sets—350. The GA yields the lowest results when compared to other techniques, giving 62.25%, 62.02%, 85.21%, 37.75%, 60.00%, and 85.00% outcomes, respectively, at the maximum number of training data. Additionally, the accuracy values of GA, EGA, MBSO, HSM, BGWO, and BMGWO at the maximum quantity of training data are 62.25%, 64.65%, 75.05%, 83.54%, 91.00%, and 98.70%, respectively, as shown in Fig. 15a. Figure 15c illustrates the precision values of different methods at the maximum number of training data sets: 62.00%, 65.36%, 72.50%, 82.50%, 85.02%, and 90.01%, respectively. From these measurements, it is noted that BMGWO is superior to other recent optimization algorithms.

At the maximum number of training data, Fig. 15d shows that the recall values of these selection methods, in the same order, are 85%, 65%, 78.25%, 86.01%, 89.16%, and 91.11%, respectively. Figure 15g illustrates that the F1-score values of these selection methods, in the same order, are 60%, 67%, 80%, 88%, 91%, and 93%, at the maximum number of training data. According to Fig. 16a, the AUC of the ROC of GA, EGA, MBSO, HSM, BGWO, and BMGWO at the maximum quantity of training data are 85%, 86%, 84%, 93%, 94%, and 96% respectively. These measures show that, whereas GA provides the least desirable set of features that restrict the NB model’s ability to learn, BMGWO can provide the best features that enable the NB model to learn effectively. With the highest number of training data sets, shows that BMGWO provides the least execution time and GA provides the maximum time, with values of 2 and 8.2 seconds, respectively. The execution times of GA, EGA, MBSO, HSM, BGWO, and BMGWO are 8.2, 5, 4.25, 3.99, 2.9, and 2 seconds, respectively.

To confirm the AMDS strategy’s efficacy and show that it can produce accurate results, it has been tested against other contemporary diagnostic strategies, including NFM³, distance based classification (DBC)¹³, CPE¹⁴, ensemble diagnosis based genetic algorithm (EDGA)²⁸, ensemble diagnosis method (EDM)²⁹, optimized random forest algorithm (ORFA)³⁷, and extreme gradient boosting (XGBoost)⁵⁷. The *second column* of Fig. 15 illustrates the obtained accuracy, precision, recall, and F1-Score metrics based on the confusion matrix. As the figure indicates, the AMDS outperforms other techniques by providing the best performance values across all metrics. Fig. 15b shows that at the number of training data = 350, AMDS provides the maximum accuracy value and NFM provides the minimum, with values of 98.91% and 88.12%, respectively. Moreover, the accuracy percentages for DBC, CPE, EDGA, EDM, ORFA, and XGBoost are, in order, 92.30%, 94.25%, 90.90%, 95.00%, 90.00%, and 92.25%. It can be shown in Fig. 15d and f that AMDS provides the maximum precision and recall values, which are 92.01% and 89.91%, respectively. Also, the NFM provides the minimum precision and recall values, which are 64.50% and 65.00%, respectively. At 350 training data sets, the precision values obtained from DBC, CPE, EDGA, EDM, ORFA, and XGBoost are 75.25%, 83.00%, 70.00%, 90.00%, 88.00%, and 90.00% respectively. Moreover, the recall values for these techniques are in the same order: 73%, 80%, 68%, and 87%, 89%, and 89% respectively. Fig. 15h summarizes the F1-score values of the compared methods which are 66.00%, 74.00%, 81.00%, 69.00%, 88.00%, 90.00%, 90.35%, and 90.91%, respectively. According to Fig. 16b, the AUC-ROC of the NFM, DBC, CPE, EDGA, EDM, ORFA, XGBoost, and AMDS at the maximum quantity of training data are 95.84%, 96.00%, 94.45%, 92.28%, 99.29% and 99.54%, respectively. These measures show that the NFM provides the least desirable results and AMDS provide the best.

In addition to the above results, statistical analysis is conducted to prove the efficiency of the proposed method against other models. Namely, various statistical measures, including mean, median, standard deviation (STD), and variance (VAR) are calculated. Table 5 includes the maximum (Max), minimum (Min), mean, median, STD,

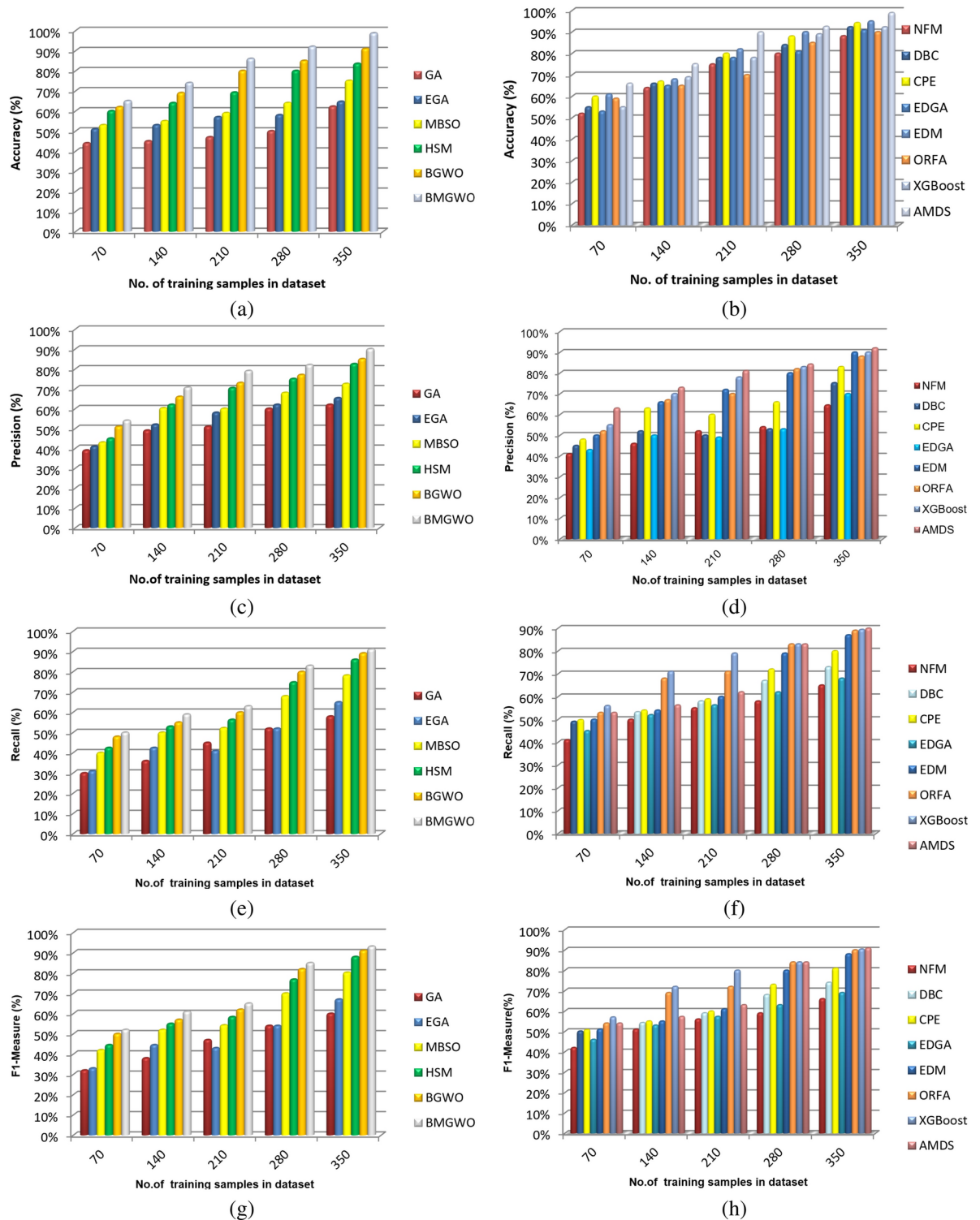


Fig. 15. System accuracy for the feature selection techniques using NB (first column) and the proposed monkeypox diagnostic strategy compared with other methods (second column). The rows show the accuracy, precision, recall, and F1- score, top-to-down, respectively.

and VAR values of the accuracy of the proposed methods against other models after running 50 independent runs. Additionally, the Boxplot of the objective function across independent runs is shown in Fig. 17. According to Table 5, it is noted that the best results have been obtained from AMDS and the second best method is EDM while the worst method is NFM. The convergence graph for the proposed AMDS compared to other strategies is also illustrated in Fig. 18.

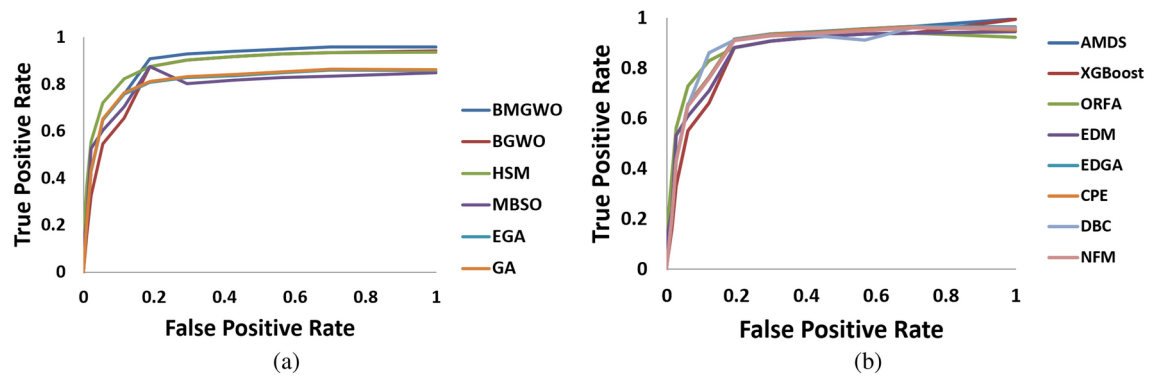


Fig. 16. Receiver operating characteristics comparing (a) various feature selection techniques using NB; (b) the proposed monkeypox diagnostic strategy against other methods.

Model	Statistical measures				
	Max	Min	Median	Mean±STD	VAR
NMF ³	88.12	84.00	84.00	82±0.2	0.04
DBC ¹³	92.30	88.01	90.00	89±0.4	0.05
CPE ¹⁴	94.25	87.21	89.00	88±0.3	0.02
EDGA ²⁸	90.90	89.98	89.02	87±0.2	0.04
EDM ²⁹	95.00	90.00	91.02	90±0.1	0.2
ORFA ³⁷	90.00	87.01	90.00	88±0.25	0.03
XGBoost ⁵⁷	92.25	88.02	91.00	88±0.3	0.09
Proposed (AMDS)	98.91	91.00	92.00	91±0.05	0.02

Table 5. Statistical measures of the proposed AMDS accuracy against other models using run= 50. The “STD”, “VAR”, “NMF”, “DCB”, “CPE”, “EDGA”, “EDM”, “ORFA”, and “XGBoost” indicate standard deviation, variance, distance based classification, ensemble diagnosis based genetic algorithm, ensemble diagnosis method, optimized random forest algorithm, and extreme gradient boosting (XGBoost), respectively.

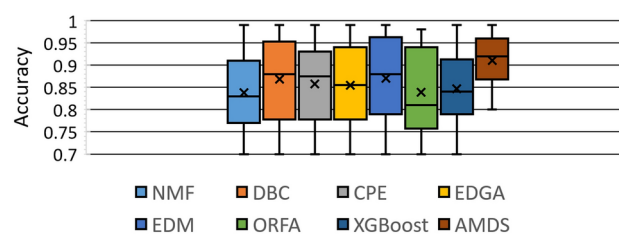


Fig. 17. The Boxplot of the objective function across independent runs for proposed monkeypox diagnostic strategy against other methods. the “X” indicates the mean.

Moreover, a series of ablation studies have been conducted to test how various system components contribute to the overall performance. Thus, the proposed AMDS is first evaluated using three different voting scenarios, MV, WMV, and CBV. The accuracy of AMDS based on CBV showed better results compared with MV and WMV with values 98.91%, 94.02%, and 96.23% respectively, as shown in Table 6. Secondly, ablation studies using a combination of the modified BMGWO, EOC, and CBV are conducted. According to Table 6, it is noted that the accuracy of the integration of MGWO, CBV, and EOC is better than other scenarios of ablated modules.

To ensure the generalizability of the proposed strategy, the AMDS is trained on the monkeypox data set, then the trained model is tested on an external COVID data set⁵¹. The results comparing our analysis framework with to others strategies are presented in Table 7. It is clearly observed from the results of the second experiment that when our model is tested on external datasets it still provided the best results.

Discussion

This work introduces a comprehensive strategy, the AMDS, for Moneybox detection. Sequentially, three fundamental operations are carried out. The first is feature extraction followed by the proposed novel MGWO

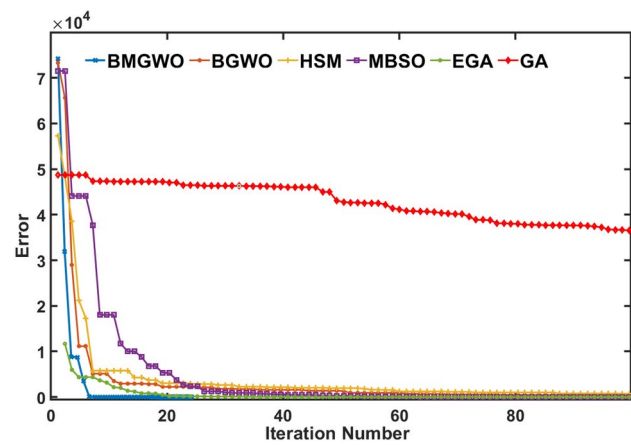


Fig. 18. The convergence curves for monkeypox diagnosis strategies for proposed monkeypox diagnostic method against other methods.

MGWO	CBV	Classifier	Accuracy (%)
	✓	✓	90.00
✓		✓	91.92
		✓	88.00
✓	✓	DLBC	89.02
✓	✓	WNBC	87.01
✓	✓	FDBC	80.00
✓	✓	✓	98.91

Table 6. Ablation studies of key components of the proposed AMDS model.

Metrics (%)	Dataset	Compared methods							
		Proposed	XGBoost ⁵⁷	ORFA ³⁷	EDM ²⁹	EDGA ²⁸	CPE ¹⁴	DBC ¹³	NMF ³
Accuracy	COVID ⁵¹	98.00	91.69	88.05	94.05	89.00	93.58	91.23	87.65
	MPV ⁵⁸	99.00	92.03	88.98	95.00	90.85	94.02	92.98	89.65
Precision	COVID ⁵¹	91.00	89.80	87.00	89.25	69.00	82.32	74.00	63.40
	MPV ⁵⁸	91.65	90.25	88.65	90.25	71.00	83.00	75.98	63.99
Recall	COVID ⁵¹	89.00	88.00	72.00	79.25	67.88	79.23	72.00	64.00
	MPV ⁵⁸	91.00	90.00	74.65	80.95	69.00	80.52	73.65	64.89
F1-score	COVID ⁵¹	89.58	89.23	89.52	87.69	68.00	80.69	73.69	65.00
	MPV ⁵⁸	90.52	91.28	90.00	89.95	70.21	81.95	74.00	65.25
AUC-ROC	COVID ⁵¹	98.26	98.52	91.00	93.85	95.85	94.00	92.00	91.85
	MPV ⁵⁸	98.69	98.85	92.63	93.89	96.00	94.85	93.01	92.36

Table 7. Additional validation of the proposed system using two external COVID and monkeypox virus (MPV) datasets compared with other techniques.

for feature selection and feature weighting. To give a more precise diagnosis, the informative features are passed to EoC as a monkeypox diagnostic model. The FDBC, WNBC, and DLBC diagnostic techniques are integrated, utilizing a weighted voting method to create the hybrid ensemble of classification model. Two scenarios were used to test and evaluate the proposed AMDS. First, the BMGWO was tested and its outcomes compared with other current selection techniques with a conventional classifier⁵⁶.

The first set of experiments for evaluating the proposed pipeline with the proposed BMGWO to alternative methods. As indicated by the measurements of the performance metrics in Fig. 15, the suggested BMGWO produces more accurate findings. Namely, BMGWO outperforms other methods and attains the highest accuracy, precision, recall, F1-score, and AUC-ROC values-98.70%, 90.01%, 91.11%, 93.05%, and 96.00% respectively, with only 1.3% minimal error value. Following BMGWO, BGWO presents the greatest results (91.00%, 85.02%, 89.16%, 91.00%, and 94.00% for accuracy, precision, accurate findings. Namely, BMGWO outperforms other methods and attains the highest accuracy, precision, recall, F1-score, and AUC-ROC, respectively). Furthermore,

with the highest number of training data sets, its error value is 9%. The BMGWO, however, is slower than BGWO, but is quicker than GA, EGA, MBSO, and HSM. BMGWO can quickly and precisely identify a subset of attributes from these readings to diagnose patients. Further investigation of the results in Fig. 15 shows that the BGWO method is the second-best approach after BMGWO. As a result, MBGWO is better than BGWO at improving diagnostic performance. Ultimately, the most optimally chosen features from MBGWO will advance to the following phase, when the suggested EoC model will be trained and tested on a valid dataset containing informative features. Further evaluation is performed to demonstrate the efficiency of the proposed MGWO against other versions of GWO for binary classification problems. The results in Table 4 documents the superiority of MGWO compared to other GWO versions. The results also showed that the modified versions of the GWO generally yielded improved performance over the original algorithm.

Additional comparison with other methods reveals that the suggested AMDS method outperforms the NFM, DBC, CPE, EDGA, EDM, ORFA, and XGBoost strategies. This is confirmed by the results shown in Fig. 15, second column. The proposed method produces the best results when the maximum number of training data sets (350) is used. It can be seen from these measurements that AMDS produces the best results, while NFM produces the worst ones. This can be explained in part by the fact that NFM is used on the original dataset without utilizing the feature selection approach, whereas AMDS starts by removing unnecessary features before learning the EoC model. At the maximum number of training data, Fig. 15k demonstrates that AMDS takes a lengthy time to implement, while NFM takes a short time-5.5 and 4 seconds, respectively. Furthermore, after AMDS, EDM produces the best outcomes and requires a lengthy implementation period. As a result, AMDS takes a lengthy time to diagnose patients accurately; however, this delay is overlooked in favor of a precise diagnosis. Ultimately, AMDS is found to be more effective than NFM, DBC, CPE, EDGA, EDM, ORFA, and XGBoost methods. Additional statistical measures confirmed that AMDS outperformed FM, DBC, CPE, EDGA, EDM, ORFA, and XGBoost methods after running 50 independent runs where it provides the best values across all measures. Generalizability of the proposed AMDS is also evaluated, i.e., how well the proposed framework trained on one dataset performs on a new, unseen dataset. Here, the model is trained on the original monkeypox dataset and tested on the two additional data sets for binary classification problems; the MPV and COVID dataset⁵¹. The results shown in Table 7 suggest that the model was better able to generalize and correctly classify cases in different datasets.

Further robustness analysis has been conducted using the ROC analysis, as shown in Fig. 16. According to Fig. 16a, the AUC of the associated ROC of BMGWO feature selection method shows enhanced performance compared with other algorithms with value reach to 96.00% at the maximum number of training data. Figure 16b, on the other hand documents that the proposed AMDS strategy is better than other algorithms with value reach to 99.54%. Further analysis of the metaheuristics algorithm is conducted by motioning the objective function across independent runs, as this is an important visual measurement of the performance of the optimization algorithm. The results of such analysis is demonstrated using Boxplot in Fig. 17. The figure shows that the proposed pipeline has the highest accuracy and the smallest spread of the objective function across independent runs. This is also consistent with the results in Table 5.

Moreover, the convergence of the proposed modified GWO is visually analyzed by depicting the value of the objective function over the course of iterations. As demonstrated in Fig. 18 the proposed BMGWO offers faster convergence with fewer parameters. This can be explained in part by the fact that it achieves better local optima avoidance through dynamic weighting of leader wolves, which is also highlighted in Fig. 8. This is confirmed with the statistical measures after running 50 independent runs in Table 5. Finally, the conducted ablation studies in Table 6 prove the efficiency of AMDS by demonstrating how MGWO, ensemble classifiers, and CBV interact for better performance.

The current work introduced a viable technique for Monkeypox diagnosis, which achieved promising accuracy. Nonetheless, certain limitations exist and there is a room for additional research for future improvement. First, the available dataset consists of a single modal data binary samples and only contains structured data. In the future, a large dataset of monkeypox having multiple classes, as well as multimodal data will be explored for enhanced robustness and predictive power. Moreover, the model mainly considers the fusion of classifier-based architecture, which primarily focuses on selected features through BMGWO. Despite the results demonstrated the potential of the proposed prediction pipeline; however, it lacks explainability and interpretability of the machine's decisions. Different weighing schemes should be explored. Future research addressing these limitations could improve the system's classification performance and offer insightful information for clinical decision-making.

Conclusions and future directions

This paper has introduced an integrative monkeypox diagnostic strategy, so-called AMDS. The pipeline is hybrid and integrates a modified grey wolf-based approach for feature selection and weighting with an ensemble machine classifier. The latter is integrates a confusion based voting strategy combining three learnable machine classifiers to introduce the best diagnostic results. Quantitative and qualitative results documented that proposed hybrid method offers an efficient solution for fast and accurate monkeypox diagnosis, addressing a critical need in public health. Furthermore, the suggested AMDS performed better than alternative approaches. At 350 training data points, the AMDS yielded results for accuracy, error, precision, recall, F1-measure, and ROC-AUC, and run-time of 98.91%, 1.09%, 92.01%, 89.91%, 90.91%, 99.54%, and 5.5 seconds, respectively. Further, the strength and generalizability of AMDS were proved by testing it against other models using a different data sets. Despite better diagnostic capabilities, AMDS was trained and tested on a single modal data and introduced the largest run-time value. In the future it is planned plan (1) guide the implementation and testing of AMDS on multimodal inputs, i.e., structure clinical data and imaging data to increase the system capability via the inclusion of deep features; (2) incorporate RNA-seq and gene dataset to verify the robustness and ability of AMDS to diagnose

monkeypox using different types of datasets; (2) enhance methodological pipeline by incorporating outlier rejection procedures, investigate other learnable modules and voting strategies to boost AMDS performance.

Data availability

The dataset used and analyzed during the current study are available from Nile Lab for Artificial Intelligence (AI) repository http://covid19.nilehi.edu.eg/Available_datasets.php

Received: 28 October 2024; Accepted: 20 January 2025

Published online: 30 January 2025

References

- Shaban, W. M., Rabie, A. H., Saleh, A. I. & Abo-Elsoud, M. A new covid-19 patients detection strategy (cpds) based on hybrid feature selection and enhanced knn classifier. *Knowl.-Based Syst.* **205**, 106270 (2020).
- Iqbal, N. & Kumar, P. Integrated covid-19 predictor: Differential expression analysis to reveal potential biomarkers and prediction of coronavirus using rna-seq profile data. *Comput. Biol. Med.* **147**, 105684 (2022).
- Tom, J.J. & Anebo, N.P. A neuro-fuzzy based model for diagnosis of monkeypox diseases (2018).
- Centers for Disease Control and Prevention. Monkeypox. <https://www.cdc.gov/poxvirus/monkeypox/symptoms.html> (Accessed 16 July 2021).
- Centers for Disease Control and Prevention. Monkeypox. <https://www.cdc.gov/poxvirus/monkeypox/outbreak/us-outbreaks.html> (Accessed 17 May 2022).
- World Health Organization (WHO). Monkeypox. <https://www.who.int/news-room/fact-sheets/detail/monkeypox> (Accessed 19 May 2022).
- Oladoye, M. J. Monkeypox: a neglected viral zoonotic disease. *Electron. J. Med. Educ. Technol.* **14**, em2108 (2021).
- Ou, G. et al. Automated robot and artificial intelligence-powered wastewater surveillance for proactive mpox outbreak prediction. *Biosaf. Health* **6**, 225–234 (2024).
- Yadav, S. & Qidwai, T. Machine learning-based monkeypox virus image prognosis with feature selection and advanced statistical loss function. *Med. Microecol.* **19**, 100098 (2024).
- Muhammed Kalo Hamdan, A. & Ekmekci, D. Prediction of monkeypox infection from clinical symptoms with adaptive artificial bee colony-based artificial neural network. *Neural Comput. Appl.* 1–16 (2024).
- Rustagi, T. & Vijarana, M. Hybridizing wolf search algorithm with xgboost model for accurate identification of cardiac disorders. *Front. Health Inform.* 1439–1461 (2024).
- Bacanin, N. et al. Improving performance of extreme learning machine for classification challenges by modified firefly algorithm and validation on medical benchmark datasets. *Multimed. Tools Appl.* 1–41 (2024).
- Rabie, A. H., Saleh, A. I. & Mansour, N. A. A covid-19's integrated herd immunity (cihi) based on classifying people vulnerability. *Comput. Biol. Med.* **140**, 105112 (2022).
- Rabie, A. H., Mansour, N. A., Saleh, A. I. & Takieldein, A. E. Expecting individuals' body reaction to covid-19 based on statistical naïve bayes technique. *Pattern Recogn.* **128**, 108693 (2022).
- Bacanin, N. et al. Addressing feature selection and extreme learning machine tuning by diversity-oriented social network search: an application for phishing websites detection. *Complex Intell. Syst.* **9**, 7269–7304 (2023).
- Bacanin, N. et al. The explainable potential of coupling hybridized metaheuristics, xgboost, and shap in revealing toluene behavior in the atmosphere. *Sci. Total Environ.* **929**, 172195 (2024).
- Latha, R. et al. Feature selection using grey wolf optimization with random differential grouping. *Comput. Syst. Sci. Eng.* **43**, 317–332 (2022).
- Harrison, K. R., Engelbrecht, A. P. & Ombuki-Berman, B. M. An adaptive particle swarm optimization algorithm based on optimal parameter regions. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1–8 (IEEE, 2017).
- Xu, X., Lin, Z., Li, X., Shang, C. & Shen, Q. Multi-objective robust optimisation model for mdvrpls in refined oil distribution. *Int. J. Prod. Res.* **60**, 6772–6792 (2022).
- Parpinelli, R. S., Lopes, H. S. & Freitas, A. A. Data mining with an ant colony optimization algorithm. *IEEE Trans. Evol. Comput.* **6**, 321–332 (2002).
- Rajeswari, M., Amudhavel, J., Pothula, S. & Dhavachelvan, P. Directed bee colony optimization algorithm to solve the nurse rostering problem. *Comput. Intell. Neurosci.* **2017**, 6563498 (2017).
- Mirjalili, S. & Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016).
- Mirjalili, S. The ant lion optimizer. *Adv. Eng. Softw.* **83**, 80–98 (2015).
- Jin, X., Zhang, S., Ding, Y. & Wang, Z. Task offloading for multi-server edge computing in industrial internet with joint load balance and fuzzy security. *Sci. Rep.* **14**, 27813 (2024).
- Mashru, N., Tejani, G. G., Patel, P. & Khishe, M. Optimal truss design with moho: A multi-objective optimization perspective. *PLoS One* **19**, e0308474 (2024).
- Yan, A. & Yan, J. Method of feature weight optimization based on grey wolf and bird swarm algorithm. *J. Beijing Univ. Technol.* **49**, 1088–1098 (2023).
- Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014).
- Abdollahi, J. & Nouri-Moghaddam, B. A hybrid method for heart disease diagnosis utilizing feature selection based ensemble classifier model generation. *Iran J. Comput. Sci.* **5**, 229–246 (2022).
- Umarani, N., Samanta, D. & Chakraborty, P. Machine learning technology-based heart disease detection models. *J. Healthc. Eng.* (2022).
- Abdelhamid, A. A. et al. Classification of monkeypox images based on transfer learning and the al-biruni earth radius optimization algorithm. *Mathematics* **10**, 3614 (2022).
- Arora, K. et al. Using deep learning algorithms for accurate diagnosis and outbreak prediction of monkeypox. In *2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM)*, 1–5 (IEEE, 2024).
- Alharbi, A. H. et al. Diagnosis of monkeypox disease using transfer learning and binary advanced dipper throated optimization algorithm. *Biomimetics* **8**, 313 (2023).
- Jaradat, A. S. et al. Automated monkeypox skin lesion detection using deep learning and transfer learning techniques. *Int. J. Environ. Res. Public Health* **20**, 4422 (2023).
- Almutairi, S. A. DI-mdf-oh2: optimized deep learning-based monkeypox diagnostic framework using the metaheuristic harris hawks optimizer algorithm. *Electronics* **11**, 4077 (2022).
- Nayak, T. et al. Detection of monkeypox from skin lesion images using deep learning networks and explainable artificial intelligence. *Appl. Math. Sci. Eng.* **31**, 2225698 (2023).
- Sorayaie Azar, A. et al. Monkeypox detection using deep neural networks. *BMC Infect. Dis.* **23**, 438 (2023).
- Hapsari, R. K. et al. Optimization based random forest algorithm modification for detecting monkeypox disease. In *2023 Sixth International Conference on Vocational Education and Electrical Engineering (ICVEE)*, 340–346 (IEEE, 2023).

38. Xia, J.-Y. et al. Metalearning-based alternating minimization algorithm for nonconvex optimization. *IEEE Trans. Neural Netw. Learn. Syst.* **34**, 5366–5380 (2022).
39. Lin, W. et al. Programmable macrophage vesicle based bionic self-adjuvanting vaccine for immunization against monkeypox virus. *Adv. Sci.* 2408608 (2024).
40. Su, Y. et al. Colon cancer diagnosis and staging classification based on machine learning and bioinformatics analysis. *Comput. Biol. Med.* **145**, 105409 (2022).
41. Huang, H., Wu, N., Liang, Y., Peng, X. & Shu, J. Slnl: a novel method for gene selection and phenotype classification. *Int. J. Intell. Syst.* **37**, 6283–6304 (2022).
42. Yaqoob, A., Mir, M. A., Jagannadha Rao, G. & Tejani, G. G. Transforming cancer classification: The role of advanced gene selection. *Diagnostics* **14**, 2632 (2024).
43. Fan, Z., Liu, Y., Ye, Y. & Liao, Y. Functional probes for the diagnosis and treatment of infectious diseases. *Aggregate* e620 (2024).
44. Rampogu, S. A review on the use of machine learning techniques in monkeypox disease prediction. *Sci. One Health* 100040 (2023).
45. Mansour, N. A., Saleh, A. I., Badawy, M. & Ali, H. A. Accurate detection of covid-19 patients based on feature correlated naïve bayes (fcnb) classification strategy. *J. Ambient Intell. Hum. Comput.* 1–33 (2022).
46. Yang, J. & Kim, J. An accident diagnosis algorithm using long short-term memory. *Nucl. Eng. Technol.* **50**, 582–588 (2018).
47. Le, X.-H., Ho, H. V., Lee, G. & Jung, S. Application of long short-term memory (lstm) neural network for flood forecasting. *Water* **11**, 1387 (2019).
48. Wu, X. et al. Long short-term memory model-a deep learning approach for medical data with irregularity in cancer predication with tumor markers. *Comput. Biol. Med.* **144**, 105362 (2022).
49. Rabie, A. H., Ali, S. H., Ali, H. A. & Saleh, A. I. A fog based load forecasting strategy for smart grids using big electrical data. *Clust. Comput.* **22**, 241–270 (2019).
50. Mienye, I. D. & Sun, Y. A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access* **10**, 99129–99149 (2022).
51. Nile Lab for Artificial Intelligence (AI). MonkeyPox. http://covid19.nilehi.edu.eg/Available_datasets.php (Accessed 26 July 2022).
52. Chantar, H. et al. Feature selection using binary grey wolf optimizer with elite-based crossover for arabic text classification. *Neural Comput. Appl.* **32**, 12201–12220 (2020).
53. Nadimi-Shahraki, M. H., Taghian, S. & Mirjalili, S. An improved grey wolf optimizer for solving engineering problems. *Expert Syst. Appl.* **166**, 113917 (2021).
54. Qiu, Y., Yang, X. & Chen, S. An improved gray wolf optimization algorithm solving to functional optimization and engineering design problems. *Sci. Rep.* **14**, 14190 (2024).
55. Tuba, E., Strumberger, I., Bezdán, T., Bacanin, N. & Tuba, M. Classification and feature selection method for medical datasets by brain storm optimization algorithm and support vector machine. *Procedia Comput. Sci.* **162**, 307–315 (2019).
56. Edeh, M. O. et al. A classification algorithm-based hybrid diabetes prediction model. *Front. Public Health* **10**, 829519 (2022).
57. Farzipour, A., Elmi, R. & Nasiri, H. Detection of monkeypox cases based on symptoms using xgboost and shapley additive explanations methods. *Diagnostics* **13**, 2391 (2023).
58. Eid, M. M. et al. Meta-heuristic optimization of lstm-based deep network for boosting the prediction of monkeypox cases. *Mathematics* **10** (2022).

Acknowledgments

This work is supported by the Center for Equitable Artificial Intelligence and Machine Learning Systems (CEAMLS), Morgan State University, Project #11202202.

Author contributions

All authors contributed to the study's conception and experimental design. Material preparation and data collection were performed by S. E. E., A. H. R., and A. I. S.; Data analysis, validation and visualization were performed by S. E. E., A. T. and F. K.; Supervision: A. H. R., A. I. S.; Fund Acquisition: A. S. and F. K.; The first draft of the manuscript was written by S. E. E.; A. H. R. and A. T.; All authors commented, edited, and approved the final manuscript before submission

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to F.K.

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