



Adaptive Disease Diagnosis Strategy (ADDS) Based on Enhanced Incremental Artificial Intelligence

Citation: Mansour, N.; Rabie, A.; saraya, M.; saleh, I.

Inter. Jour. of Telecommunications, IJT 2025, Vol. 05, Issue 01, pp. 1-14, 2025.

Editor-in-Chief: Youssef Fayed.

Received: 20/01/2025.

Accepted: date 17/03/2025.

Published: date 17/03/2025.

Publisher's Note: The International Journal of Telecommunications, IJT, stays neutral regarding jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2024 by the authors. Submitted for possible open access publication under the terms and conditions of the International Journal of Telecommunications, Air Defense College, ADC, (<https://ijt.journals.ekb.eg/>).

A.Mansour, Nehal*¹, H.Rabie, asmaa², Sabry saraya, M.³, and I.saleh, Ahmed⁴

¹Nile Higher Institute for Engineering and Technology, Artificial Intelligence Lab, Mansoura, Egypt, nehalegypt1993@gmail.com.

²Computers and Control Dept. Faculty of Engineering, Mansoura University, Mansoura, Egypt. asmaa91hamdy@yahoo.com

³Computers and Control Dept. Faculty of Engineering, Mansoura University, Mansoura, Egypt, mohamedsabry@mans.edu.eg.

⁴Computers and Control Dept. Faculty of Engineering, Mansoura University, Mansoura, Egypt, aisaleh@yahoo.com.

Abstract: Artificial intelligence (AI) has demonstrated significant promise in medical diagnostics. Healthcare providers can enhance patient outcomes and advance personalized healthcare systems by utilizing these AI-driven tools to make quicker, more accurate diagnoses. Despite its capabilities, AI has not yet achieved the same level of human intelligence. The main contribution of this paper is to present a new strategy called Adaptive Disease Diagnosis Strategy (ADDS). ADDS introduces a new approach to disease diagnosis, particularly for emergent diseases such as monkeypox virus (MPXV), by imitating human learning processes in contrast to conventional diagnostic procedures that depend on static models. ADDS uses the binary groupers and moray eels (BGME) optimization algorithm to select the most critical features from the MPXV dataset. ADDS is based on a modified version of Naive Bayes (NB), which is called Enhanced Incremental NB (EINB). EINB emulates how humans acquire knowledge by continually adapting to new information while building upon prior knowledge. It modifies its diagnostic capabilities without necessitating complete retraining. EINB optimizes the model's efficacy by adding a selective data approach that only incorporates the most significant information into the learning process, thereby avoiding useless updates from irrelevant data. This selective approach enables the model to retain previous information, guaranteeing that past data is valuable as new insights are incorporated into the model. The results demonstrate that the ADDS significantly enhances the performance of the monkeypox diagnostic system with an accuracy value equal to 99.46%, ensuring that the model remains accurate, adaptive, and ready for emerging challenges.

Keywords: Enhanced Incremental Artificial intelligence, disease diagnosis, Enhanced Naïve Bayes, and MPXV detection.

1. Introduction

Delivering precise and easily obtainable diagnoses is a major obstacle for healthcare systems worldwide. Approximately 5% of outpatients in the United States are given an incorrect diagnosis annually. These errors frequently occur when diagnosing patients with severe medical problems, with over 20% of these patients being misdiagnosed in primary care. Furthermore, one out of every three of these misdiagnoses leads to significant harm to the patient. Artificial intelligence (AI) and machine learning (ML) have recently become potent tools for

addressing intricate challenges across various fields[1]. Specifically, using ML in diagnosis can transform healthcare greatly[2]. Despite extensive studies, diagnostic algorithms have faced challenges in matching the accuracy of doctors in differential diagnosis. This is particularly true when several potential explanations exist for a patient's symptoms[3]. Furthermore, healthcare systems consistently gather a significant volume of data, which continues to grow exponentially. The analysis of medical data involves several challenges and concerns, which can be listed as follows: (i) the process of obtaining large amounts of data with clear indications necessitates a significant investment of human resources, materials, and time, (ii) once the dataset hits a specific threshold, the training period of the dataset becomes significantly prolonged, leading to time consumption, (iii) when retraining the dataset using traditional methods, the program will enter a closed state, which is not appropriate for online applications, (iv) the general ML classification approach is unable to automatically identify and update a new class as it is formed. Therefore, it is unable to fulfill the current requirements, and (v) over time, the distribution and characteristics of the data will inevitably change. Then, the developed classification model that uses past data may become inadequate for certain new data [4].

Due to the previous problems, there has been an increasing need for Incremental Artificial Intelligence (IAI). It is an AI model that learns and improves its knowledge in stages. In simple terms, it emulates human learning patterns by gaining new information over time while maintaining and expanding the existing knowledge. IAI is critical when data arrives in sequential order or when storing all data for processing becomes impossible. From the standpoint of computational intelligence, IAI holds significance for at least two key reasons. First, from a data mining viewpoint, many modern applications require algorithms that can continuously learn from large and ever-changing data streams to enhance future learning and decision-making processes[5]. Secondly, from the perspective of ML, biological systems inspire IAI by demonstrating the ability to learn incrementally and form associations over time to achieve objectives. This approach is especially valuable in real-world applications, as it allows models to update themselves with new data, eliminating the need to rebuild models entirely[6]. This method upgrades the existing AI models to an accurate one by replacing the traditional methods, which necessitate constructing a new model from scratch[7].

IAI has many challenges, such as catastrophic forgetting, difficulty handling concept drift, the risk of overfitting, and the new data quality [6]. When the model tends to forget old information as it gets new data, this is called catastrophic forgetting [7]. Concept drift refers to the dynamic changes in the data distribution or relationships between input features and the target feature over time. Overfitting risk occurs when an incremental learning algorithm over-adjusts its parameters based on recent data, which may not accurately reflect the overall distribution [8]. New data quality denotes that inconsistent, incorrect, or noisy data can impact the model's predictions, resulting in poor learning. This paper introduces Enhanced IAI (EIAI) to implement practical approaches to overcome the challenges inherent to IAI.

This paper introduces the Adaptive Disease Diagnosis Strategy (ADDS), a new strategy to address the urgent need for early diagnosis of the monkeypox virus (MPXV) and overcome existing challenges in MPXV diagnosis. Leveraging the Enhanced IAI (EIAI) in combination with the Binary Groupers and Moray Eels (BGME) optimization algorithm. The BGME is a binary version of the GME optimization algorithm[9]. BGME emulates the associative hunting behavior observed between groupers and moray eels. Associative hunting, or cooperative hunting among distinct species of animals, is exceedingly uncommon. This hunting practice contrasts with that of conspecific groups owing to the elevated coordination between the two species. BGME algorithm may effectively evade the local optima trap. This occurred due to a combination of evidence derived from the behaviors of two distinct species, each exhibiting a unique movement manner. This yields comprehensive search domain coverage and a balanced approach between the exploitation and exploration phases. GME is uncomplicated to implement in relation to most metaheuristic algorithms. It comprises four phases: primary search (PS), pair association (PA), encircling or extended search (ES), and attacking and catching (AC). The optimization rule in BGME generally entails choosing the optimal solutions, continuing the process for a certain number of itera-

tions or until a termination criterion is met, such as achieving the desired solution quality or completing a predetermined number of iterations. ADDS uses the BGME to enhance the strategy's predictive accuracy by identifying the most significant features of MPXV.

ADDS is based on EINB which is a modified version of Naive Bayes. It is distinguished by its continuous ability to improve performance. It begins training with an appropriate amount of examples and then starts working initially, even in degraded performance. Then, it has the ability to adjust performance by taking advantage of experiences gained from new cases that it diagnoses without losing previous experiences or retraining. Thus, it mimics the way humans learn. The EINB learns from continuous unprocessed data, accumulates experience over time, and utilizes this knowledge to enhance future learning and the performance of medical systems. Additionally, it seeks to address the limitations of IAI.

IAI is advantageous for the diagnosis of MPXV for a variety of reasons:

1. In the case of an emergent disease such as MPXV, the quantity of historical diagnostic data may be restricted. IAI can resolve this by progressively integrating new data from ongoing outbreaks, research studies, and clinical observations. This guarantees that diagnostic models enhance over time.
2. The disease's changing patterns: As MPXV spreads across various regions and populations, patterns of symptoms, characteristics, transmission dynamics, and viral variants may change over time. IAI can manage this concept drift and enable the diagnostic model to continuously update its knowledge base with the most recent data, guaranteeing that it remains pertinent and accurate in diagnosing new cases under various conditions.
3. Enhanced diagnostic accuracy: IAI enhances diagnostic accuracy as additional cases are diagnosed and more data becomes available. This necessitates improved differentiation between MPXV and comparable diseases (such as chickenpox or smallpox), which is crucial for implementing appropriate treatment strategies.

IAI in the diagnosis of MPXV mimics human learning from several perspectives, such as;

- Mimicking human memory: Humans use memory to remember past cases and diagnose current issues. A doctor may recall how a symptom appeared in prior situations to help diagnose current patients. IAI systems learn new information while retaining old.
- Mimicking human learning over time: Humans continuously learn from their encounters with the world. Doctors and researchers improve their diagnostic skills with new cases, symptoms, and studies. More data improves IAI models, making them more accurate. As cases increase, the algorithm improves its ability to diagnose MPXV.
- Similar to Human Experience: Learning from past cases, medical research, patient histories, and new symptoms improve disease diagnosis over time. Medical data are used to train an IAI diagnosis algorithm. The system incrementally learns from recent information (such as outbreak reports, symptom changes, or demographic patterns) without forgetting past knowledge.

The main advantages and the key contributions of this paper are summarized below;

- This paper presents the Adaptive Disease Diagnosis Strategy (ADDS), a new strategy to meet the critical demand for early diagnosis of the MPXV.
- ADDS consists of two stages, which are the preprocessing stage and the classification stage.
- The preprocessing stage focuses on feature selection, where the most relevant features are identified to improve the efficiency and accuracy of the model. The classification stage then utilizes EINB) to classify patients.
- BGME, which mimics the rare phenomenon of associative hunting between different species in nature, is employed to select the most significant features from the MPXV dataset. Experimental results show that the BGME outperforms comparable algorithms in feature selection issues.
- EINB updates the model's parameters (class priors and feature likelihoods) after each Incremental Update Trigger value denotes as (IUT), instead of updating them for every new data. This approach has many key

benefits, such as reducing computational overhead, smoothing learning, avoiding overfitting, and being beneficial for handling concept drift.

- The process of updating the training set with new data relies on evaluating its accuracy. This is done to ensure that the model's performance improves or, at the very least, does not deteriorate. The updating mechanism can be understood through three scenarios:
 - When accuracy increases:
If the model's accuracy improves compared to the previous iteration, the new data points (or cases) are added to the training set. This triggers an update of the model, allowing it to learn from the additional data that positively contributes to its performance.
 - When accuracy decreases or remains the same:
The new data points are not added to the training set if the accuracy decreases or remains the same. This ensures the model does not incorporate data that could degrade its performance or provide no added value.
- EINB addresses catastrophic forgetting because it retains knowledge from earlier iterations, ensuring that valuable information learned previously is not overwritten or forgotten.
- EINB is efficient in memory utilization. Instead of retaining the complete dataset, only feature statistics and class counts need to be preserved and updated when new data arrives.
- EINB is recoverable; it can restore its performance after a decline, enabling it to return to its previous optimal performance despite a decrease in accuracy during certain learning phases.
- EINB is suitable for diagnosing different diseases, especially emerging diseases because it is trained to work with incomplete or limited data, unlike traditional methods that rely on large datasets for accurate results. This is crucial when new diseases emerge, and the data is limited or restricted, making it harder to diagnose and predict the disease's behavior.
- The results of the experiments proved the efficiency and accuracy of the strategy in diagnosing MPXV compared to other methods.

The paper's structure is as follows: Section 2 provides an overview of AI classification based on various learning methods. Section 3 reviews the related work in the field. Section 4 provides a detailed explanation of the ADDS. Section 5 evaluates the proposed strategy's performance, presenting results and analysis. Finally, the paper concludes with a summary and remarks in Section 6.

2. AI Classification

AI can be classified based on its capabilities, functionality, and underlying technologies [8]. In terms of capabilities, examples include general AI (strong AI), narrow AI (weak AI), and super intelligent AI. Classification by functionalities includes types such as limited memory systems, theory of mind AI, reactive machines, and self-aware AI. Additionally, AI leverages technologies like ML, Deep Learning (DL), Robotics, Natural Language Processing (NLP), Expert Systems, and Computer Vision. In this paper, we classify AI specifically based on the learning methods, introducing categories such as batched AI, incremental AI (IAI), online AI, active AI, adaptive AI, transfer AI, federated AI, ensemble AI, and enhanced incremental AI, as illustrated in Figure 1. The next subsection provides a brief explanation of AI classification based on these learning methods.

2.1. Batched AI

In batch learning, the ML model is trained using the entire dataset that is available at a particular time. The training procedure is considered complete when the model achieves satisfactory performance on the test set. This approach is also referred to as offline learning, as it does not continuously adjust to new data. In the event that the model requires updating, it must be retrained from scratch using a combination of the previously ac-

quired and newly acquired data samples if additional data becomes available at a later time. Nevertheless, batch learning is not appropriate for situations in which the data is perpetually generated from the source, as it is unable to effectively manage real-time updates or incremental learning requirements[10][11].

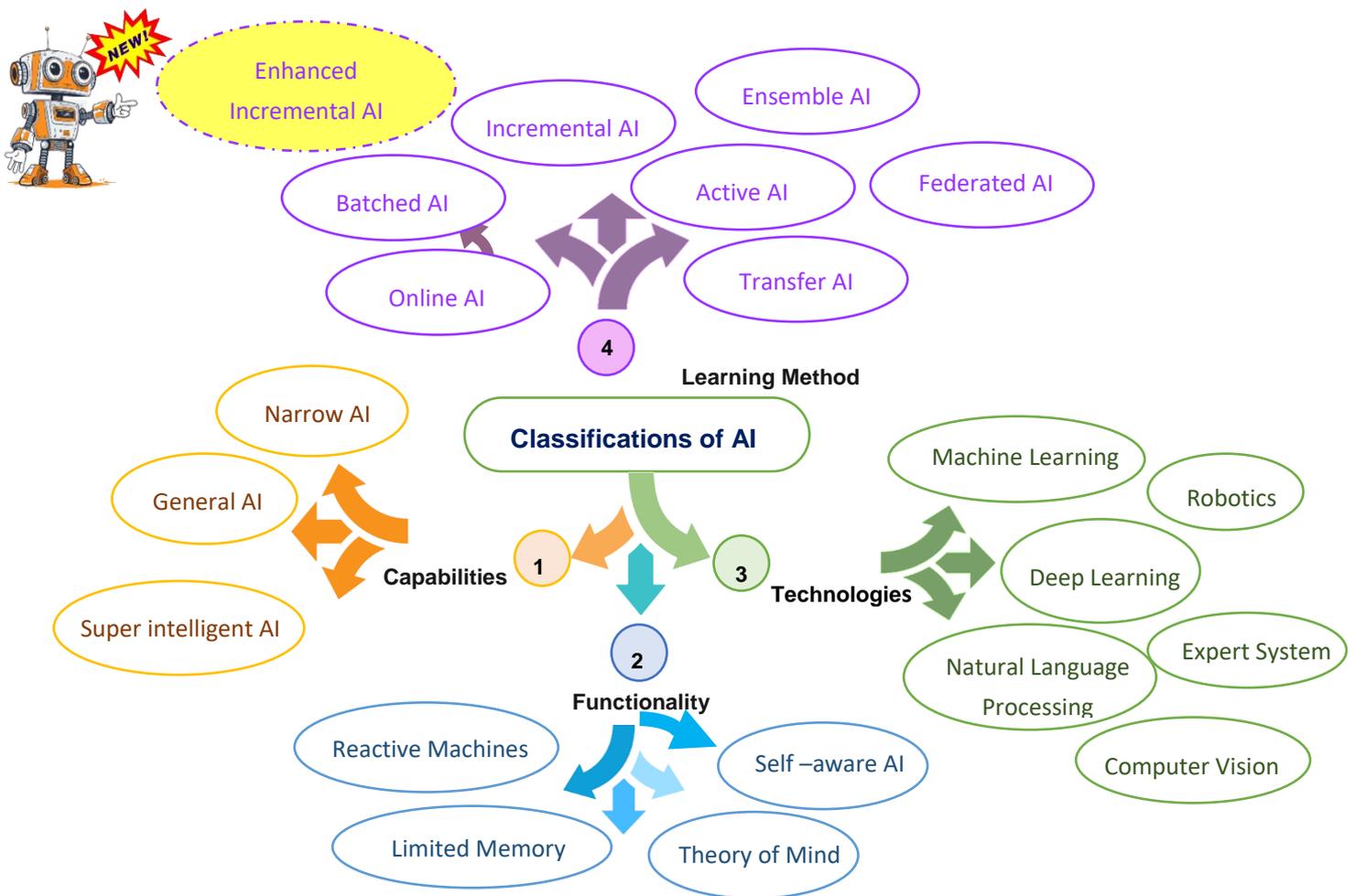


Figure 1. The different classifications of AI

2.2. Transfer AI

The capacity of the system to recognize and apply the knowledge and skills acquired from prior tasks to new tasks is referred to as transfer AI. Transfer learning is defined as the process of extracting knowledge from one or more sources of tasks and applying it to a target task [12]. The Domain and the task are two fundamental concepts in transfer learning. The learning process involves inputting data into the model for training to derive the prediction function required for the task in each domain [13]. Knowledge is transferred from the source domain to the target domain. Typically, the source domain contains a substantial amount of labeled training data, while the target domain has either a minimal amount or none. Transfer learning addresses the issue of insufficient training data in the target domain, enabling better performance in the subject of interest.

2.3. Ensemble AI

Ensemble AI is fundamentally based on the concept of aggregating the predictions of multiple models, each of which may possess its strengths and weaknesses. This can result in enhancing the efficiency, achieving superior predictive performance compared to a single model, and generalization[14].

2.4. Federated AI

Federated AI is a configuration in which a central aggregator coordinates the efforts of multiple clients to resolve ML challenges. Additionally, this configuration permits the decentralization of training data to guarantee the privacy of each device[15].

2.5. Online AI

Online AI is a learning technique that involves the model learning incrementally from a stream of data elements in real-time. It is a dynamic process that adjusts its predictive algorithm over time, enabling the model to evolve in response to new data. This approach has paramount importance in rapidly changing data environments, as it provides precise timely predictions. Although incremental learning and online learning both process data incrementally, there are slight differences. In contrast to incremental learning, which processes data in batches at predetermined intervals, online learning processes data in real-time and modifies its model continuously.

2.6. Active AI

Active AI is an ML methodology wherein a learner requests an oracle, acting as an instructor, to label particular ambiguous samples that will yield relevant insights for the learning process. Consequently, the learner enhances the learning performance by employing fewer training examples. It is highly effective in situations where there is a substantial amount of unlabeled data, but the annotation process is either costly or time-consuming. This method involves the learner being in control of the data and requesting annotations from an entity with extensive domain knowledge (usually a human expert) for unlabeled examples[16].

2.7. Incremental AI (IAI)

In IAI, the learning process continues as new data instances become available[17]. The algorithm updates the current data model in accordance with the knowledge it has acquired from the most recent data instances. Initially, a data model is developed and subsequently updated in response to the arrival of new data. This method updates the existing model to an accurate version, in contrast to conventional methods that involve the construction of a new model from the start [18]. Figure 2 illustrates the challenges of IAI.

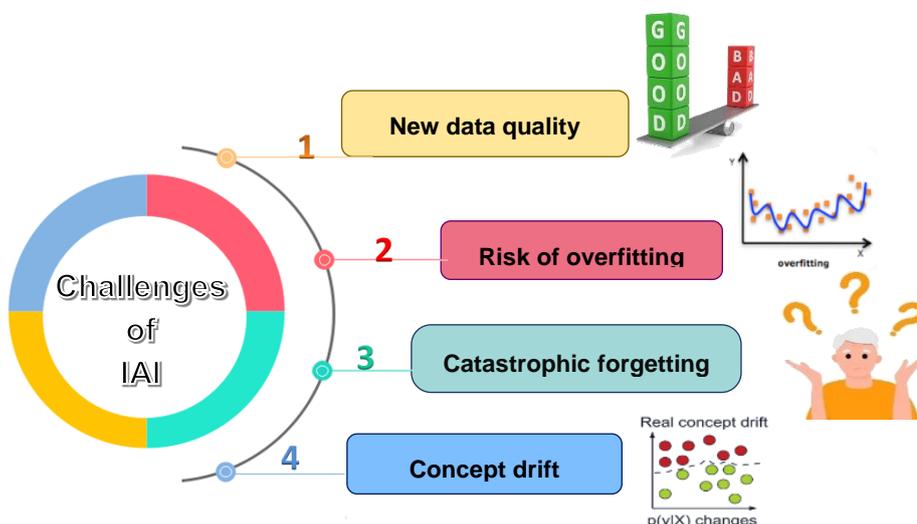


Figure 2. Challenges of IAI

2.8. Enhanced Incremental AI (EIAI)

In the field of ML, there has long been an underlying assumption that a "good" training set is readily available for any domain. A training set is considered "good" if it includes all the critical knowledge necessary for a learning algorithm to perform effectively. This means that once the algorithm learns from this dataset, it should be able to apply the acquired knowledge accurately to any new examples within the same domain. As a result, traditional ML approaches focus on maximizing the information extracted from this pre-defined dataset. However, this ideal scenario often fails to align with the complexities of real-world applications. For example, the environment constantly changes in dynamic control systems, disease diagnosis, web mining, and time series analysis. These applications generate data that evolves, meaning the learning algorithms must work with training sets that are updated or expanded incrementally. In such cases, the learning process becomes continuous, requiring the algorithm to adapt to new information as it becomes available rather than relying solely on a static dataset prepared in advance.

Therefore, IAI methodologies are designed to address challenges associated with continuously evolving data by enabling models to incorporate newly acquired information while modifying their structure. However, IAI faces several significant challenges, including difficulty handling concept drift, catastrophic forgetting, the risk of overfitting, and the new data quality [17][19]. Enhanced IAI (EIAI) is introduced to implement effective strategies to overcome the challenges inherent to IAI. First, EIAI must handle scenarios where the entire dataset is unavailable during initial model development, allowing the system to learn incrementally. Second, EIAI must integrate new data seamlessly without compromising the knowledge already acquired. Third, it must balance stability (preserving existing knowledge) and plasticity (adapting to new data).

3. Related Work

Diagnosis is the most difficult challenge, as certain symptoms and indications are ambiguous. ML is a discipline that can assist in the disease diagnosis by utilizing prior training data [20]. The most effective methods for reducing disease-related mortality rates are early diagnosis and timely treatment. Consequently, the majority of researchers are attracted to the development of predictive model that are founded on ML or Deep Learning (DL) to predict diseases. In this section, the previous efforts in disease diagnosis will be discussed. Table 1 provides a comparison between the state of the art methods for diagnosing MPXV.

The authors in [14] proposed an ensemble-based approach employing DL and ML models (EMLDL) to forecast an individual's propensity to develop cardiovascular disease. In order to forecast cardiovascular disease, they implemented an ML-based ensemble model that includes K nearest neighbor (KNN), Extreme Gradient Boosting (XGB), and Decision Tree (DT). Deep Neural Network (DNN), Keras Deep Neural Network (KDNN), and ML ensemble models are used to improve the classification results. A dataset of cardiovascular disease cases is employed to train the models. Random forest (RF) is used to extract critical cardiovascular disease features. The EMLDL model obtains a disease prediction accuracy of 88.70%, as evidenced by the experiment results.

In [17], an approach for developing predictive models of the SEIRD (Susceptible, Exposed, Infected, Recovered, and Dead) variables for the COVID-19 pandemic is proposed. The incremental learning architecture considered into account two critical processes: feature engineering and incremental learning. ILA is a two-step process that is founded on the Ensemble Learning paradigm: The incremental learning model employs a bagging scheme in the initial stage to generate predictions that are derived from a variety of predictive models that have been constructed using a variety of machine learning techniques. Next, the most accurate prediction is chosen as the

output by this model. In the event that the outcome is inadequate, it either constructs new prediction models (possibly utilizing new machine learning techniques) or modifies the existing models.

The study in [21] aims to identify MPXV through its clinical symptoms, which comprise boolean and categorical variables. It utilizes several ML models (MLM): DT, logistic regression, NB, support vector machine (SVM), random forest, and Adaptive Boosting. Data preprocessing techniques were employed to clean and prepare the dataset for ML analysis. The data cleaning process involved eliminating nonessential variables, verifying missing values, identifying duplicates, assessing conflicting values, and converting all binary variables to 0 and 1. The optimal model was achieved using the support vector machine, yielding the largest area under the precision-recall curve at 79.67%, with a recall of 88.35%, precision of 72.86%, and an F2-score of 84.53% under the baseline model configuration.

In the event of a potential pandemic, the authors in [13] utilized DL techniques to identify MPXV through skin lesions efficiently. The study incorporated hyperparameter tuning and transfer learning (TL) technologies to enhance the performance of the DL models. By modifying the TL model and incorporating hyper-parameters, a hybrid function learning model was developed within the framework of Convolutional Neural Networks (CNNs). This approach was applied to multiple models, including Xception, Vgg19, DenseNet121, ResNet50, EfficientNetV2, and MobileNetV3-s. Metrics such as AUC, accuracy, recall, loss, and F1-score were used for performance evaluation. Among the models, the optimized hybrid MobileNetV3-s achieved the best results, with an average F1-score of 0.98, AUC of 0.99, accuracy of 0.96, and recall of 0.97.

The researcher in [22] Proposed a federated learning-based architecture (FLA)utilizing deep learning models for the secure classification of MPXV and other pox viruses. The proposed framework comprises three principal components: (a) a cycle-consistent generative adversarial network for data sample augmentation during training; (b) DL models including MobileNetV2, Vision Transformer (ViT), and ResNet50 for classification; and (c) a flower-federated learning environment to ensure security. The ViT-B32 model has an accuracy value of 97.90% in the testing.

A methodology based on ensemble learning to detect the MPXV in skin lesion images is proposed in [23]. The approach begins with fine-tuning three pre-trained base learners, Inception V3, Xception, and DenseNet169. Probabilities generated by these deep models are then incorporated into an ensemble framework. A normalization technique based on the Beta function is employed to effectively combine the outputs, enabling efficient aggregation of complementary information from the base learners. The final ensemble decision is achieved using the sum rule. The deep learning framework (DLF) is rigorously evaluated on a publicly available MPXV skin lesion dataset using a five-fold cross-validation technique. This methodology demonstrates strong performance, achieving average accuracy, precision, recall, and F1 scores of 93.39%, 88.91%, 96.78%, and 92.35%, respectively.

The authors in [24] presented a hybrid architecture for the diagnostic of MPXV called the Accurate monkeypox diagnosis strategy(ADMS), utilizing a modified grey wolf optimization model for efficient selection of features and weighting. Furthermore, the system employs an ensemble of classifiers that utilizes a confusion-based voting mechanism for combining significant data features. Weighted Naïve Bayes Classifier (WNBC) is the initial classifier, and the subsequent classifier used in the pipeline is the Long Short-Term Memory (LSTM). The Fuzzified Distance-Based Classifier (FDBC) is the third classifier. The generalizability of the proposed model is assessed by evaluating its performance on external datasets for MPXV and COVID-19. The model attained an overall diagnosis accuracy of 98.00% for external COVID data sets and 99.00% for MPXV datasets.

A novel detection strategy utilizing AI approaches is proposed in [25] for the early identification of MPXV patients. This approach is called the Human Monkeypox Detection (HMD) strategy and has two pri-

mary phases: (i) Selection Phase (SP) and (ii) Detection Phase (DP). Improved Binary Chimp Optimization (IBCO) algorithm is presented in SP as a feature selection method prior to training an Ensemble Diagnosis (ED) model, which serves as a new diagnostic approach in the subsequent phase termed DP. The IBCO algorithm is a hybrid selection algorithm that incorporates both filter and wrapper approaches. Three diagnostic algorithms, Weighted Naïve Bayes (WNB), Weighted K-Nearest Neighbors (WKNN), and DL are integrated through a weighted voting technique to yield optimal diagnostic outcomes. The weighted values of the WNB algorithm are calculated by assessing the influence of each feature on the class categories, whereas the Grey Wolf Optimization (GWO) algorithm is employed to ascertain the weighted values of WKNN. The HMD technique yields superior outcomes compared to other comparable strategies, achieving accuracy, precision, and recall scores of 98.48%, 91.1%, and 88.91%, respectively, and the implementation time is 5.4 Sec.

The objective of the research in [26] is to establish a model for differentiating monkeypox infection based on the clinical symptoms that manifest in the infected individual. The proposed model combined the adaptive artificial bee colony (aABC) algorithm with an artificial neural network (ANN). The results of the proposed model were compared to those of ten other ML models that were trained on the same dataset. The deep learning model obtained the highest accuracy, with a score of 75%. It was succeeded by the random forest model, which achieved an accuracy of 71.1%.

The authors provided an approach to detect the MPXV by employing symptoms as the premise for detection as proposed in [27]. In order to accomplish this, a dataset was generated by utilizing published data on MPXV. The output of all models that have been presented is not interpreted, and they are similar to a black box. The problem was resolved by analyzing the model's output using Shapley additive explanation (SHAP). This study compared Random Forest, GBoost, CatBoost, SVM and LightGBM, all gradient-boosting algorithms that are frequently employed in ML. The results of the study indicated that XGBoost was the most accurate in diagnosing monkeypox based on symptoms.

The authors in [28] suggested an approach (GWO-CNN) that employs CNNs to classify MPXV skin lesions. The MPXV prediction model is comprised of four stages: (1) Feature selection to identify the most significant symptoms that can improve the accuracy of MPXV diagnosis, (2) pre-processing of the MPXV data, (3) Monkeypox prediction using the CNN model, and (4) optimization of the CNN hyperparameters with the GWO algorithm. The optimized model attained an accuracy value of 95.3%.

The research in [29] proposed a new method for precisely predicting confirmed MPXV cases with an optimized Long Short-Term Memory (LSTM) deep network. The model utilized the Al-Biruni Earth Radius (BER) optimization process to refine the hyper-parameters of the LSTM-based deep network, resulting in the designation BER-LSTM for the proposed approach. The prediction model consists of five functional modules: input layer, hidden layer, output layer, network training, and network prediction. The Mean Bias Error of this model is 0.06.

4. Adaptive Disease Diagnosis Strategy (ADDS)

This section outlines the ADDS, an innovative strategy to meet the critical demand for early diagnosis of MPXV and overcome current challenges in MPXV diagnosis by utilizing the BGME and EIAl. ADDS consists of two stages: the preprocessing stage and the classification stage. The preprocessing stage focuses on feature selection, where the most relevant features are identified to improve the efficiency and accuracy of the model. The classification stage then utilizes EINB) to classify patients. Feature selection is performed to reduce dimensionality by identifying the most informative features, thereby enhancing model performance and computational efficiency. The BGME is employed as the feature selection method due to its proven effectiveness in real-world ML applications. [9]. Subsequently, an EINB is utilized to detect the presence of MPXV. This classifier is designed to adapt dynamically by incorporating enhanced incremental learning, making it particularly effective for evolving datasets.

Table 1: State-of-the-art Methods for diagnosing MPXV.

Technique	Description	Advantages	Disadvantages
EMLDL [14]	An ensemble-based approach using DL and machine ML models to predict cardiovascular disease risk (CVD). Six classification techniques to predict CVD. Random forest is used to extract important CVD features. The ML ensemble model achieved the maximum disease prediction accuracy of 88.70%.	-Combining DL and ML models yields a powerful strategy for accurate MPXV classification. -Successful ML-based ensemble model on given data. -Implementing STACKED ensemble model improves deep learning results	-Low Accuracy -Because of limited data, DL models performed poorly. -For better and more accurate results, additional datasets may be employed.
ILA[17]	A method for constructing COVID-19 pandemic predictive models of SEIRD variables is suggested. The incremental learning architecture (ILA) considered feature engineering and incremental learning. In the initial step, the incremental learning model used a bagging scheme to generate predictions from a range of machine learning-based predictive models. Then, this model outputted the best accurate prediction. If the result is poor, it creates new prediction models (perhaps using machine learning) or adjusts existing ones.	-Enhancement of prediction accuracy as the training process advances and the learning algorithm encounters a greater number of training examples.	-The data dependency analysis was evaluated only by one metric which is Mean Absolute Percentage Error (MAPE), and the predictions were evaluated by MAPE, Mean Square Error (MSE), and coefficient of determination (R2). -Sometimes the model had a problem in the prediction when data changes.
MLM[21]	The boolean and categorical data to detect MPXV through its clinical symptoms. DT, logistic regression, NB, SVM, random forest, and Adaptive Boosting were used. Data preparation cleaned and prepared the dataset for ML analysis. Data cleaning included removing unnecessary variables, validating missing values, detecting duplicates, and converting all binary variables to 0 and 1. The support vector machine produced the optimal model with the biggest area under the precision-recall curve at 79.67%, 88.35% recall, 72.86% accuracy, and 84.53% F2-score under the baseline model setup.	The model is easy and simple to implement.	-The chi-square is used as a feature selection method. The limitation of using this method is the potential that the selected features may not be truly optimal, as the subset may still include irrelevant features.
TL [13]	DL was used to identify MPXV through skin lesions during an epidemic. Hyperparameter adjustment and TL improved the DL model performance in the study. This method was used on Xception, Vgg19, Dense-Net121, ResNet50, EfficientNetV2, and MobileNetV3-s. Performance was measured by AUC, accuracy, recall, loss, and F1-score. Best model: optimized hybrid MobileNetV3-s, with an average F1-score of 0.98, AUC of 0.99, accuracy of 0.96, and recall of 0.97.	-Optimization and hyperparameter updates of CNN and transfer learning models led to high success and low loss rates.	-Low-quality dataset. -Image datasets utilized for the detection of MPXV disease lack classification by accredited medical professionals. No official dataset has been released by health institutions yet.
FL [22]	a deep learning-based federated learning-based architecture (FLA) is proposed for secure MPXV and other pox virus categorization. The proposed framework includes a cycle-consistent generative adversarial network for data sample augmentation during training, DL models like MobileNetV2, Vision Transformer (ViT), and ResNet50 for classification, and a flower-federated learning environment for security. Testing shows 97.90% accuracy for the ViT-B32 model.	-The first paper that used the federated learning to diagnose the MPXV.	-The scalability and generalizability challenges of the suggested approach across Additional datasets and clinical environments require consideration.
DLF[23]	Ensemble learning that used to detect MPXV in skin lesion images is proposed. The process starts with fine-tuning three pre-trained base learners: Inception V3, Xception, and Dense-Net169. These deep model probabilities are then used in an ensemble framework. A Beta-based normalizing technique is used to efficiently aggregate complementary information from base learners and combine outputs. Sum rule determines ensemble decision. Five-fold cross-validation is used to thoroughly analyze the deep learning framework (DLF) using a publically available Monkeypox skin lesion dataset. The values of average accuracy, precision, recall, and F1 scores are 93.39%, 88.91%, 96.78%, and 92.35% respectively.	-Addition of Gaussian noise reduced the overfitting and introduced variability in the learning process	-The model achieved Low Accuracy. -Lack of attention-based methods to focus on important regions for better diagnosis. -The dataset is small
ADMS[24]	The paper presented a hybrid architecture for the diagnostic of MPXV, utilizing a modified grey wolf optimization model for efficient selecting of features and weighting. Furthermore, the system employs an ensemble of classifiers that utilizes a confusion-based voting mechanism for combining significant data features. The generalizability of the proposed model is assessed by evaluating its performance on external datasets for MPXV and COVID-19. The model attained an overall diagnosis accuracy of 98.00% for external COVID data sets and 99.00% for MPXV data sets.	-Fast and accurate strategy for the diagnosis of MPXV. -Small number of hyperparameter. -	-Small dataset is used.
HMD[25]	A novel AI-based strategy for early MPXV patient detection, called the Human Monkeypox Detection (HMD) strategy, is proposed in [25]. It consists of two phases: (i) Selection Phase (SP) and (ii) Detection Phase (DP). In SP, the Improved Binary Chimp Optimization (IBCO) algorithm is used for feature selection before training an Ensemble Diagnosis (ED) model in DP. IBCO is a hybrid algorithm combining filter and wrapper methods. Three diagnostic algorithms—Weighted Naive Bayes (WNB), Weighted K-Nearest Neighbors (WKNN), and Deep Learning (DL)—are integrated using weighted voting for optimal results. WNB weights are based on feature influence, while Grey Wolf Optimization (GWO) is used to determine WKNN weights. The HMD approach outperforms comparable strategies, achieving 98.48% accuracy, 91.1% precision, and 88.91% recall, with an implementation time of 5.4 seconds.	-Several techniques are used in this strategy that obtain good results. -The strategy delivers results with a short implementation time of 5.4 seconds, making it suitable for real-time applications	-Small dataset is used. -The integration of different algorithms require careful parameter tuning.
aABC-ANN [26]	A model for differentiating monkeypox infection based on the clinical symptoms that manifest in the infected individual. The proposed model combined the adaptive artificial bee colony (aABC) algorithm with an artificial neural network (ANN).	-ANNs are highly flexible and capable of learning complex patterns in data.	-The evaluation of the proposed diagnosing methodology is predicated on a very restricted dataset, which comprises only 240 suspected cases, of which 120 are normal and 120 are infected.
XGBoost[27]	The authors provided an approach to detect the MPXV by employing symptoms as the premise for detection. In order to accomplish this, a dataset was generated by utilizing published data on MPXV. The output of all models that have been presented is not interpreted, and they are similar to a black box. The problem was resolved by analyzing the model's output using Shapley additive explanation (SHAP). This study compared Random Forest, SVM, XGBoost, CatBoost, LightGBM, and all gradient-boosting algorithms that are frequently employed in ML. The results of the study indicated that XGBoost was the most accurate in diagnosing MPXV based on symptoms.	-Interpretability through SHAP. -First paper to diagnose the MPXV using the clinical symptoms.	-Risk of Overfitting -Complexity of the system.
GWO-CNN[28]	An approach (GWO-CNN) that employs CNNs to classify MPXV skin lesions. The MPXV prediction model is comprised of four stages: (1) Feature selection to identify the most significant symptoms that can improve the accuracy of MPXV diagnosis, (2) pre-processing of the MPXV data, (3) Monkeypox prediction using the CNN model, and (4) optimization of the CNN hyperparameters with the GWO algorithm. The optimized model attained an accuracy value of 95.3%.	Enhanced the CNN model with the application of the Grey Wolf Optimizer (GWO) algorithm, yielding substantial advancements in accuracy, precision, recall, F1-score, and AUC.	-Further validation and testing of CNN models on more diverse and large datasets may be necessary to guarantee their robustness, generalizability, and reliability. -Training and optimizing CNN models may necessitate substantial computational resources and specialized knowledge.

BER-LSTM [29]

The research in [29] proposed a new method for precisely predicting confirmed MPXV cases with an optimized Long Short-Term Memory (LSTM) deep network. The model utilized the AI-Biruni Earth Radius (BER) optimization process to refine the hyper-parameters of the LSTM-based deep network, resulting in the designation BER-LSTM for the proposed approach. The prediction model consists of five functional modules: input layer, hidden layer, output layer, network training, and network prediction. The Mean Bias Error of this model is 0.06.

The model consists of five clear functional modules: input layer, hidden layer, output layer, network training, and network prediction. This modular design makes the model easy to understand, implement, and potentially extend or modify for similar tasks.

When evaluated on an extensive dataset, the equilibrium between exploration and exploitation stages of the optimization algorithm is time-consuming.

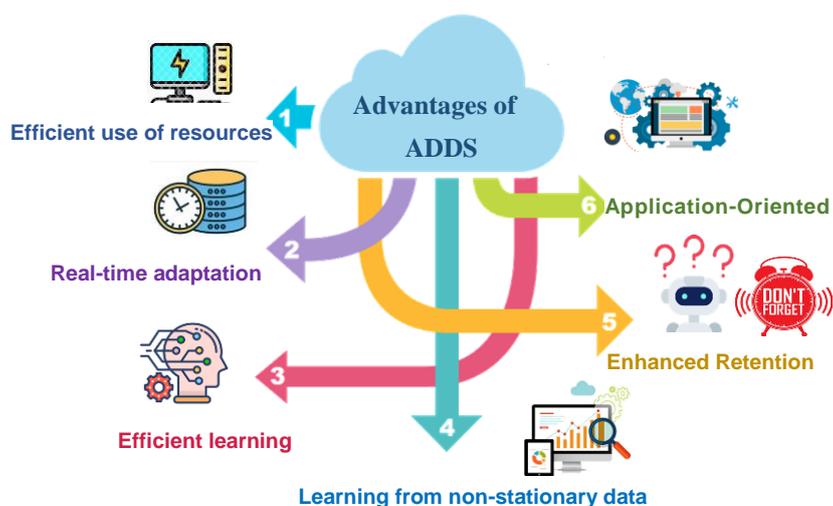


Figure 3. The Advantages of ADDS.

ADDS achieves three key criteria for assessing the robustness of any learning algorithm, which include stability, improvement, and recoverability. Stability guarantees that the prediction accuracy on the test set remains consistent during incremental learning phases, hence ensuring reliability throughout the learning process. Improvement denotes the algorithm's capability to attain a noticeable increase in prediction accuracy as it processes additional training data, illustrating its proficiency in efficiently incorporating new knowledge. Recoverability refers to the algorithm's ability to restore its performance after a decline, enabling it to return to its previous optimal performance despite a decrease in accuracy during certain learning phases. Figure 3 shows the Advantages of ADDS. Figure 4 shows the first step of ADDS. Figure 5 shows the second step of ADDS. Figure 6 and Figure 7 shows the third and fourth steps of ADDS respectively.

1 The first step of the ADDS is to select the most important features from the MPXV dataset by using GME optimization algorithm, and then create the frequency distributions tables for each selected feature according to " patient class " and " non -patient class "

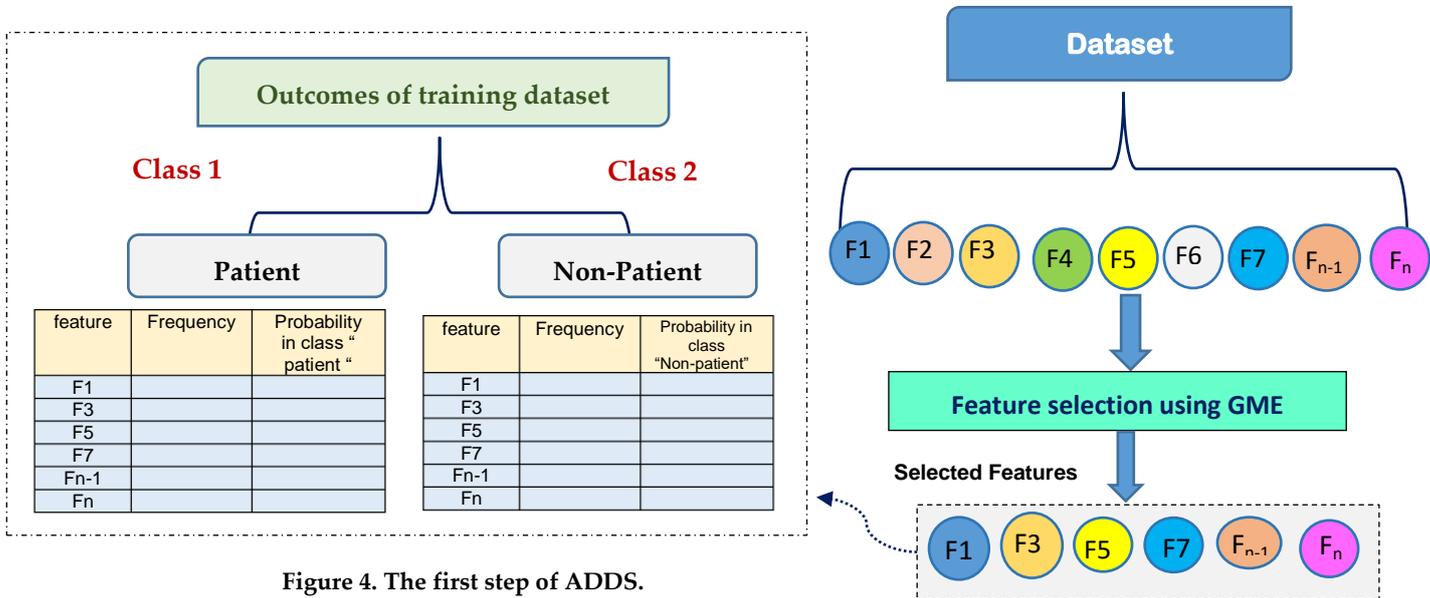
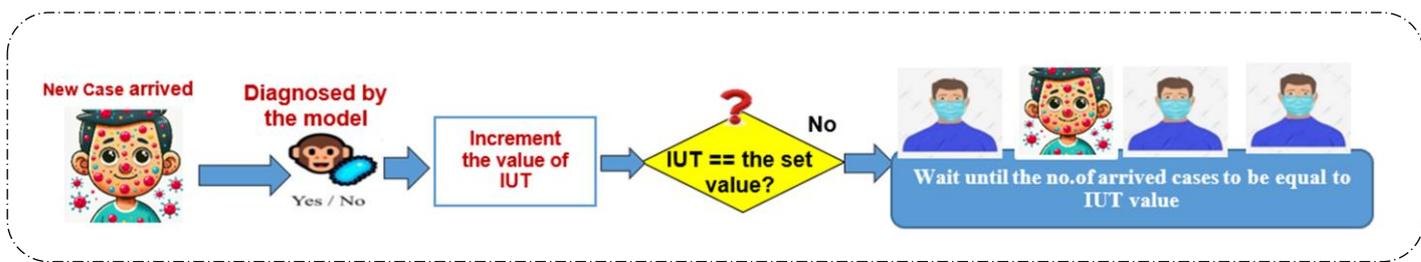


Figure 4. The first step of ADDS.

2

In the second step of ADDS, the EINB classifier is trained using the initial training set, and the initial accuracy of the model is calculated. When a new case arrives, the model will diagnose it. If the number of arrived cases equals to IUT value , the frequency distribution tables will be updated accordingly. If not, the model will remain the same.



3 In the third step of ADDS, if the no.of incoming cases equals the determined value of IUT, the frequency distribution tables will be updated. The initial test cases are used to determine the new accuracy of the model according to the updated values of the probability distribution tables.

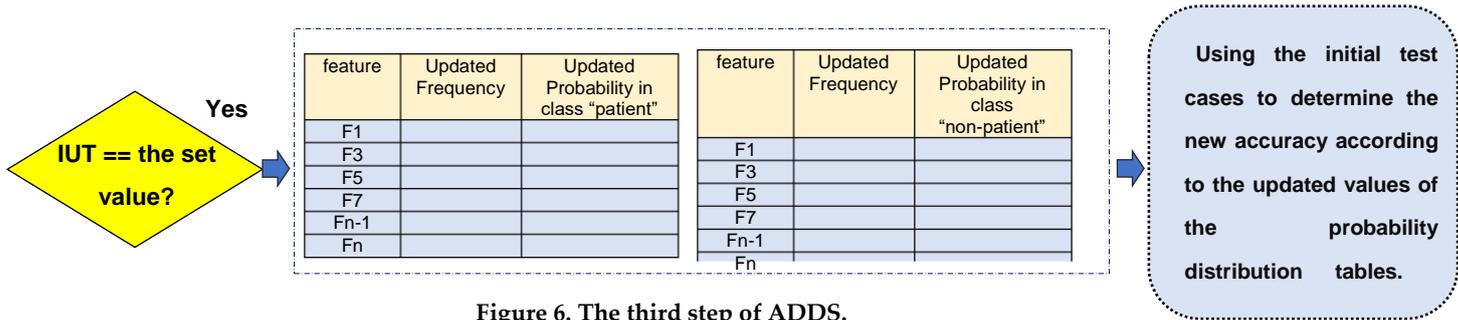


Figure 6. The third step of ADDS.

4

After calculating the new accuracy of the model, it will be compared with the previous value of it .If it is equal to or less than the previous value, the new incoming data will not be added to the training set and the frequency distribution table will not updated. If the accuracy increases, the new incoming data will be added to the training set and the frequency distribution tables will be updated accordingly.

Compare the new accuracy with the previous value of it

$Acc(i+1) > Acc(i)$

Update the training set with the new arrived cases and use the new training set to diagnose the incoming cases.



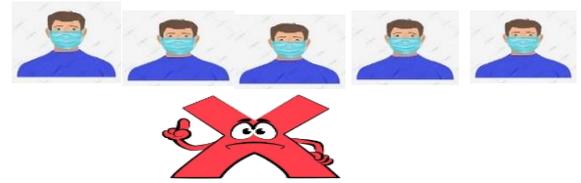
Training set



Used to diagnose the new incoming cases then update the frequency distribution table.

$Acc(i+1) \leq$

Don't add the new cases i.e ignore them, and use the previous model to diagnose the incoming cases



Continue the process of Enhanced incremental learning until reach to the target accuracy.

Figure 7.The fourth step of ADDS.

4.1. Binary Grouper and Moray Eel (BGME) Optimization Algorithm

GME is a new algorithm that is inspired by the collaboration between moray eels and groupers, each of which employs a distinct attack strategy[9]. When these two predators combine their hunting capabilities, the prey has a low likelihood of survival due to the fact that multiple agents are searching for prey with varying strategies and rapid access. This cooperative behavior seems to suggest a high level of intelligence. The foraging between the same species yields inferior outcomes in comparison to the cooperation between the groupers and moray eels. During the foraging process, the grouper fish and Moray eel collaborate in four phases: (i) (PS) for a prey, (ii) PA, (iii) ES, and (iv) AC [9].

This paper uses GME to select the most important features of MPXV. So, it is essential to develop a binary version of GME (BGME) that can be implemented in response to the feature selection issue. A one-dimensional vector represents each solution, and its length is contingent upon the number of features in the dataset. One of two values can be assigned to each element in the vector: 1 or 0. An indicator value of 1 indicates that the corresponding feature is selected, while 0 indicates that the feature is not selected. Consequently, to implement the

GME for the feature selection problem, a mapping mechanism from actual values to binary space must be implemented by using the sigmoid function using (1), and (2).

$$\text{Sigmoid}(Y_m^i) = \frac{1}{1+e^{-Y_m^i}} \quad (1) \quad Y_{binary_m}^i(i+1) = \begin{cases} 1 & \text{if } rand(0,1) \geq \text{sigmoid}(Y_m^i) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The most significant features of MPXV will be identified to evaluate the utility of the proposed BGME in feature selection. BGME initiates with search agents (S) that possess a set of search agents represented by Y and signifies a possible solution. A solution would specify the number of features selected to reduce the dimensionality of a specific dataset. Let P denote the total number of iterations across the PS, ES, and AC phases, with the number of iterations of the PA phase (P_{Ass}) equaling 1. Consequently, the no.of iterations for PS, ES, and AC, represented as P_{Search} , P_{Enc} , and P_{AC} , can be determined using equations (3 → 5) as follows:

$$P_{Search} = \left\lfloor \frac{P}{3} \right\rfloor \quad (3), \quad P_{Enc} = \left\lfloor \frac{P}{3} \right\rfloor \quad (4), \quad P_{AC} = \left\lfloor P - 2 * \frac{P}{3} \right\rfloor \quad (5).$$

The evaluation of these search agents should employ the accuracy metric of the K-Nearest Neighbor (KNN) classifier as a fitness function. The evaluation function can be expressed mathematically using (6). Figure 8 shows the steps for the initialization of BGME.

$$\text{Fitness}(Y_m) = \text{Acc of KNN}(Y_m) \quad (6).$$

Initialization

1. Each search agent indicates a potential solution which represents by the subset of features (1: selected, 0: not selected).

Search agents (Groupers and eels)	Features (1: selected, 0: not selected)				
	F ₁	F ₂	F ₃	F _r
Y ₁ (subset 1)	1	0	1	1
Y ₂ (subset 2)	1	0	0	0
.....
Y _{n-1} (subset n-1)	1	1	0	1
Y _n (subset n)	0	0	1	1



2. Initialize the total no.of iterations (P), and calculate the no.of iteration in the search (P_{Search}), encircling (P_{Enc}), and attacking phases (P_{AC}).
3. Initialize the no. of hops in each iterations (hop).
4. The objective function that is used to evaluate the search agents is the accuracy of KNN classifier:-

Figure 8. The initialization of BGME

In the PS phase, the grouper begins to move in a zigzag movement. Then, the m^{th} grouper's updated position is determined by equation (7), which varies by the number of hop, reflecting the moves in each iteration. When the number of hops is even, the new position is selected at random to exceed the current position, ensuring it remains within the maximum limit of the search space. In contrast, when the number of hops is odd, the new position is randomly selected to be less than the current one while adhering to the minimal boundary of the search region. Figure 9 shows the pseudocode of the PS phase.

$$\text{The updated position of } \begin{cases} Y_m^{\text{hop}+1} = \text{Rand}(Y) \text{ Where, } Y_m^{\text{hop}} < Y \leq \max(Y_m) & \text{if No.of hop is even} \\ Y_m^{\text{hop}+1} = \text{Rand}(Y) \text{ Where, } \min(Y_m) \leq Y < Y_m^{\text{hop}} & \text{if No.of hop is odd} \end{cases} \quad (7)$$

- PS
Phase


1. Partition the search agents to two equally sets which are; groupers and eels.
 2. Randomly allocate the groupers and eels throughout the search space.
 3. Compute the objective function value for each search agent.
 4. Each grouper initiates the movement in a zigzag pattern.
 5. Determine the new position for each grouper depending on the number of movements in the iteration and compute the new objective function using (7).
 6. The new positions of search agents ($Y_{binary_m}^i(i+1)$) will be converted to a binary value by using the sigmoid function.
 7. Choose the best position of the grouper from all hops in the current iteration to serve as the initial position for the next iteration until reach to the number of P_Search
 8. Select the best positions for the groupers to be the initial positions of the next phase.

Figure 9. The pseudocode of the PS phase of BGME optimization algorithm.

In the PA phase, the cooperation between the groupers and the eels allows them to discover new areas of the search domain. Each grouper fish will have an association with an eel according to the value of the objective function i.e. the grouper with the highest value of objective function associates with the eel that has the highest value of objective function and so on.

In the ES phase, each pair endeavors to encircle the prey by moving independently, thereby facilitating the exploration of various regions within the search space. The logarithmic spiral has been designated as the principal mechanism for location updates for groupers throughout the ES phase while the sinusoidal wave is used for updating the positions of the eels. The position of the prey can be determined according to the following steps.

Initially, Equation (8) is used to compute the differences between the location coordinates (Δy_{-m}) of the grouper's position (Y_{-gm}) and the position of the eel (Y_{-Em}) in order to determine the position of the prey. The distance between the grouper and the eel is determined by squaring the differences for each axis, totaling them, and then taking the square root, as indicated in (9). Then, the position of the prey can be obtained by employing (10).

$$\Delta y_{mj} = (Y_{Em} - Y_{gm}) \quad (8), \quad dis = \sqrt{\sum_{j=1}^D (\Delta y_m)^2} \quad (9), \quad c_m = Y_{gm} + \frac{L}{dis} * \Delta y_{mj} \quad (10).$$

After obtaining the position of the prey, the grouper and the eel initiate their respective movements toward the prey after they have located it. The following steps are for updating the position of the grouper in this phase. Initially, determine the distance between the grouper and the potential prey by using (11).

$$\overrightarrow{D1} = |\overrightarrow{Y}_{prey_m}(i) - \overrightarrow{Y}_{gm}(i)| \quad (11)$$

The updated position of the m^{th} grouper in the next hop is calculated by using (12). It depends on several factors: the distance $D1$ between the grouper and its prey, the constant k that defines the shape of the logarithmic spiral guiding the grouper's path, and the predicted location of the prey at iteration i . Additionally, a value w , calculated using equation (13), is influenced by the number of encircling iterations and the number of hops in the current iteration. These elements combine to determine the position of the grouper as it adjusts its movement relative to the prey and the surrounding environment.

$$\overrightarrow{Y}_{gm}(h+1) = \overrightarrow{D1} * e^{kw} \cos(2\pi w) + \overrightarrow{Y}_{prey_m}(i) \quad (12), \quad w = 1 - \frac{2*h}{P_{encircle}} \quad (13)$$

The positions of the moray eels are updated in this phase by using (14), (15), (16), and (17).

$$\vec{\lambda}(i) = |\overrightarrow{Y}_{Em}(i) - \overrightarrow{Y}_{prey_m}(i)| \quad (14), \quad \vec{\eta}(i) = \vec{\lambda}(i) * \xi \quad (15)$$

$$\text{The distance between the hops} = \frac{2\pi}{\text{Total no.of hops}} \quad (16), \quad Y_E^{i+1} = \alpha * \vec{\lambda}(i) * \xi * \sin(g) + Y_E^i \quad (17)$$

Where $\lambda^{\vec{i}}$ denotes the difference between the positions of the moray eel and its prey, the wave amplitude ($\eta^{\vec{i}}$) will be ascertained by multiplying $\lambda^{\vec{i}}$ by a factor ξ . ξ possesses a stochastic value within the interval of 0 to 1. α is a random value and $\sin(g)$ is the value of the sin angle. Figure 10 provides the pseudocode of the ES phase.

- 
1. The best positions of the groupers and moray eels from the PS phase will be used as the initial positions in the ES phase.
 2. The logarithmic spiral has been designated as the principal mechanism for location updates for groupers throughout the ES phase according to the following steps, while the sinusoidal wave is used for updating the positions of the eels.
 3. The location of the prey can be determined by using the equations (8 →10), and its location will be converted to a binary value by the sigmoid function by using (1) and (2).
 4. The groupers update their locations in the same iteration according to no.of (hop) by using the equations (11 →13), then the sigmoid function will be used to convert these locations to binary ones.
 5. The eels update their locations according to equations (14 →17), then the sigmoid function will be used to convert these locations to binary ones.
 6. The objective function will be calculated for each search agent.
 7. The steps from 4 to 6 will be repeated until reach to no. of P_{Enc} , then use the best position of each search agent to be the initial position of it in the next phase.

Figure 10. The pseudocode of the ES phase of BGME optimization algorithm.

During the AC phase, all search agents, regardless of being groupers or eels, engage in the assault on the prey after precisely encircling its position by constructing a circle with the prey positioned in its center. The procedure for establishing the circle around the anticipated prey is as follows: initially, the position of the agent exhibiting the optimal fitness function is presumed to be the location of the predicted prey. The distance between the expected prey and the other agents is subsequently computed. A circle is then established around the prey, with the radius (R) representing the distance from the prey to the most distant agent. To accommodate the numerous search agents and to minimize computational complexity, the radius of the subsequent circle can be determined using (18). During each iteration of the attack, the search agents approach the prey significantly closer until they successfully capture it. Figure 11 shows the pseudocode of the AC phase.

$$R_{i+1} = (1 - \mu) * R_i \quad (18)$$

Where the μ denoted a shrinking ratio. Figure 6 shows the pseudocode of the AC phase.

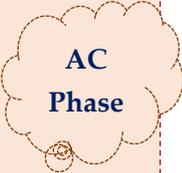
- 
1. Identify the position of the search agent exhibiting the best fitness function, which corresponds to the anticipated location of the prey.
 2. Compute the distance between the anticipated prey and all other agents.
 3. Determine the distance from the prey to the most distant agent and designate this as the radius (R) of the circle with the prey at its center.
 4. Randomly allocate the agents within the circle.
 5. Compute the objective function for each search agent.
 6. Select the search agent with the optimal value to serve as the prey in the subsequent iteration.
 7. The radius of the subsequent circles will be determined using (18).
 8. Upon determining the radius, the circle is established, with the prey situated at its center.
 9. Randomly allocate the search agent inside it.
 10. Compute the objective function for the search agent.
 11. The search agent with the optimal fitness function value will be designated as the prey.
 12. The previous steps will be repeated until reach to the no.of P_{AC}
 13. Return with the search agent which has the best objective function to become the solution which represents the most important features.

Figure 11. The pseudocode of the AC phase of BGME optimization algorithm.

To classify the data point to the target class, the model assigns it to the class that maximizes the posterior probability. First, it calculates the class prior probability and conditional probability. Prior probability refers to the probability distribution of each class based on the available data. Conditional Probability denotes the likelihood of each feature belonging to a particular class. Posterior probability refers to the chance of an unclassified sample belonging to a particular class [30].

Consider $\{C_1, C_2, \dots, C_k, \dots, C_m\}$ to be the names of the m distinct classes, Y is a data sample that consists of n features denoted as $(x_1, x_2, x_3, x_4, \dots, x_i, \dots, x_n)$. The class prior probability of a sample being a member of class C_k is denoted as $P(C_k)$. $P(Y|C_k)$ represents the conditional probability that a sample Y contains the same features as a sample whose class label is C_k . The posterior probability $P(C_k|Y)$ can be calculated using (19). It is the multiplication of the conditional probabilities of all features given the class C_k ($\prod_1^n P(x_i|C_k)$) by the class prior probability ($P(C_k)$). For making a prediction, the model computes the posterior probability for each class, then it subsequently selects the class that possesses the highest posterior probability using (20). Figure 12 illustrates how the EINB works.

$$P(C_k|Y) = [P(C_k) * \prod_1^n P(x_i|C_k)] \quad (19).$$

$$C = \underset{C_k}{\operatorname{argmax}} [P(C_k) * \prod_1^n P(x_i|C_k)] \quad (20)$$

Pseudo Code for EINB Classifier

❖ Training Phase

1. Input Training Data:

- Enter the training dataset, consisting of feature vectors and corresponding class labels (denoted as C).

2. Calculate Prior Probabilities P(C):

- Determine the prior probabilities P(C) for each class C in the dataset. This is done by calculating the relative frequency of each class in the training set :

$$P(C) = (\text{Count of Class C} / \text{Total Cases})$$

3. Create Frequency Distribution Tables:

- Build frequency distribution tables for each feature across the classes. This involves calculating how often each feature value occurs in relation to each class.

4. Calculate Conditional Probabilities P(Y|C):

- For each class C and each feature Y, determine the conditional probabilities P (Y|C). These probabilities represent the likelihood of observing feature given class C. This is done using the frequency distribution tables.

$$P (Y|C) = (\text{Count of feature value in class C} / \text{Total instances in class C})$$

❖ Testing Phase

6. Input Testing Data:

- Input a testing set, which consists of a feature vector Y_{test} , for which the class label is unknown.

7. Compute Posterior Probabilities:

- For each class C, calculate the posterior probability $P(C|Y_{test})$ using Bayes' theorem:

$$P (C | Y_{test}) = P(Y_{test}|C) * P(C)$$

8. Prediction:

- Choose the class with the highest posterior probability as the predicted class for the test instance.

9. Accuracy Calculation:

- Calculate the model's accuracy on the testing set by comparing the predicted class labels with the true labels.

❖ Incremental Learning Process

10. Set The incremental update trigger value (IUT):

- Define the value of IUT, which determines how the frequently updates of the model are performed based on new incoming data.

11. Wait until the no.of new incoming cases equals (IUT)value :

- Wait until (IUT) new cases (incoming data points) are observed before performing an update.

12. Class Prediction for Incoming Cases:

- For each incoming case, use the current model to predict the class label.

13. Update Frequency Distribution Tables:

- After predicting the class label for each incoming case, update the frequency distribution tables for each feature and class.

14. Update Conditional Probabilities:

- Update the conditional probabilities P (Y|C) for each feature in each class based on the updated frequency distributions tables.

15. Update Prior Probabilities:

- Update the prior probabilities P(C) for each class, incorporating the newly predicted class labels from the incoming cases.

16. Recalculate Accuracy:

- After updating the model, recalculate the accuracy of the model using the original testing set to evaluate the impact of the updates.

❖ Selection of Data Addition

16. Compare New Accuracy with Previous Accuracy:

- Compare the accuracy obtained after the update with the previous accuracy.

17. Decision on Data Addition:

- If the accuracy has increased, add the new incoming cases to the training set to improve the model, and use this model to classify the cases that will come up again.
- If the accuracy has decreased or remained the same, do not add the new incoming cases to the training set and retain the previous model as the current best model.

Figure 12. The pseudocode of the EINB Classifier

Table 5 shows that the model classified 2 cases correctly, but 3 cases were classified incorrectly. Table 6 shows the confusion matrix where TP is the truly positive case, TN is the truly negative case, FP is the false positive case, and FN is the false negative case .By using (21), the initial accuracy of the model is calculated and equals 0.4.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (21)$$

Table 2 . The dataset with nominal values.

Patient #	Features of dataset						Diagnosis
	PC	WBC	MC	AST	BC	LDH	
Initial training data (15 Patients)							
1	Low	Low	Low	High	Normal	Normal	Healthy
2	Low	Low	Normal	High	Normal	High	Healthy
3	Low	High	Normal	High	Normal	Normal	Infected
4	Low	High	Normal	High	High	Normal	Healthy
5	Low	Normal	High	High	Normal	Normal	Infected
6	Low	Normal	Normal	High	Normal	High	Healthy
7	Normal	Low	Low	High	Normal	Normal	Healthy
8	Normal	High	Normal	High	Normal	Normal	Infected
9	Normal	High	Normal	High	High	High	Healthy
10	Normal	Normal	High	High	Normal	Normal	Infected
11	Normal	Normal	High	High	Normal	High	Healthy
12	High	Low	Low	Normal	Normal	Normal	Healthy
13	High	Normal	High	Normal	Normal	Normal	Infected
14	High	Normal	High	Normal	High	High	Healthy
15	High	High	Normal	Normal	Normal	High	Healthy
Testing data (5 Patients)							
16	Low	Normal	High	High	High	Normal	Infected
17	Normal	Normal	High	High	High	Normal	Infected
18	High	Low	Low	Normal	Normal	High	Healthy
19	Normal	Normal	Normal	High	Normal	Normal	Infected
20	High	High	Normal	Normal	Normal	Normal	Healthy

Table 3. Nominal values of each feature.

Feature of dataset	Corresponding Nominal Values
PC	Low
	Normal
	High
WBC	Low
	Normal
	High
MC	Low
	Normal
	High
AST	High
	Normal
BC	High
	Normal
LDH	High
	Normal

Table 4. The frequency distribution tables compiled from the dataset.

(a)Platelet Count (PC).

Feature	Diagnoses	Σ	Probability in Class			
			Healthy	Infected		
PC	Low	4	2	6	4/10	2/5
	Normal	3	2	5	3/10	2/5
	High	3	1	4	3/10	1/5
Σ		10	5	15		

(b) White Blood cell (WBC).

Feature	Diagnoses	Σ	Probability in Class			
			Healthy	Infected		
WBC	Low	4	0	4	4/10	0/5
	Normal	3	3	6	3/10	3/5
	High	3	2	5	3/10	2/5
Σ		10	5	15		

(c) Monocytes Count (MC).

Feature	Diagnoses	Σ	Probability in Class			
			Healthy	Infected		
MC	Low	3	0	3	3/10	0/5
	Normal	5	2	7	5/10	2/5
	High	2	3	5	2/10	3/5
Σ		10	5	15		

(d) Aspartate aminotransferase (AST).

Feature	Diagnoses	Σ	Probability in Class			
			Healthy	Infected		
AST	High	7	4	11	7/10	4/5
	Normal	3	1	4	3/10	1/5
Σ		10	5	15		

(e) Basophils Count (BC).

Feature	Diagnoses	Σ	Probability in Class			
			Healthy	Infected		
BC	High	3	0	3	3/10	0/5
	Normal	7	5	12	7/10	5/5
Σ		10	5	15		

(f)Lactate Dehydrogenase (LDH).

Feature	Diagnoses	Σ	Probability in Class			
			Healthy	Infected		
LDH	High	6	0	6	6/10	0/5
	Normal	4	5	9	4/10	5/5
Σ		10	5	15		

Table 5. Testing the data to predict the accuracy of the initial model

Patient #	P(Healthy)	P(Infected)	Diagnoses	
Patient16	0.0006	0	Healthy	x
Patient 17	0.001	0	Healthy	x
Patient 18	0.003	0	Healthy	✓
Patient 19	0.0058	0.0512	infected	✓
Patient 20	0.0025	0.0042	infected	x

Table 6. Confusion Matrix

Actual Diagnoses	Predicted Diagnoses		
	Healthy	infected	
Healthy	TP=1	FN=1	2
infected	FP=2	TN=1	3
Σ		3	5

The previous steps are used to initially train the EINB classifier on a training dataset to establish baseline probabilities for classes and features. According to no.of incremental update trigger (IUT) value, the model updates incrementally by updating the prior probabilities for each class based on the new data. In this example, it is assumed that IUT =5. Then update the conditional probabilities of features within each class based on the occurrences of that feature in the new data and the existing data. Table 7 shows the five new incoming cases and their diagnosis by the model. The probability of class healthy is updated to 13/20, and the probability of class infected will be 7/20, as shown in Table 8. The updated frequency distribution tables for each feature based on new incoming cases are shown in Table 9 (a → f). Table 10 shows the diagnosis of the five test cases by the new model. If the accuracy of the updated model increases, the newly arrived cases will be added to the training set. If the accuracy not increase or remain as the previous value, the new cases will be deleted and not added to the training set. The Accuracy of the updated model = (TP + TN) / (TP + TN + FP + FN) = 0.6. The accuracy increases so that the five cases will be added to the training set, as shown in Table 10.

Table 7. The five new incoming cases to predict the diagnosis

1-The five incoming new cases								Diagnosis by the model
Case 1	Normal	Low	Normal	High	Normal	High	result	
P(Yes)	0.3	0.4	0.5	0.7	0.7	0.6	0.01176	Healthy
P(No)	0.4	0	0.4	0.8	1	0	0	
case 2	Low	High	Normal	High	High	High	result	Diagnosis
P(Yes)	0.4	0.3	0.5	0.7	0.3	0.6	0.00504	Healthy
P(No)	0.4	0	0.4	0.8	0	0	0	
case 3	Low	Normal	Normal	High	Normal	Normal	result	Diagnosis
P(Yes)	0.4	0.3	0.5	0.7	0.7	0.4	0.00784	Infected
P(No)	0.4	0.6	0.4	0.8	1	1	0.0256	
case 4	High	High	Normal	Normal	Normal	Normal	result	Diagnosis
P(Yes)	0.3	0.3	0.5	0.3	0.7	0.4	0.00252	Healthy
P(No)	0.2	0.4	0.4	0.2	1	1	0.00213	
case 5	High	Normal	High	Normal	Normal	Normal	result	Diagnosis
P(Yes)	0.3	0.3	0.3	0.3	0.7	0.4	0.00151	Infected
P(No)	0.2	0.4	0.4	0.2	1	1	0.00213	

Table 8. The first update of probability of classes

Class	Probability of the classes
Healthy	13/20
Infected	7/20

Table 9. The updated frequency distribution tables compiled from the dataset.

(a)Platelet Count (PC).

Feature	Diagnoses		Σ	Probability in Class		
	Healthy	Infected		Healthy	Infected	
PC	Low	5	3	8	5/13	3/7
	Normal	4	2	6	4/13	2/7
	High	4	2	6	4/13	2/7
Σ	13	7	20			

(b)White Blood cell (WBC).

Feature	Diagnoses		Σ	Probability in Class		
	Healthy	Infected		Healthy	Infected	
WBC	Low	5	0	5	5/13	0/7
	Normal	3	5	8	3/13	5/7
	High	5	2	7	5/13	2/7
Σ	13	7	20			

(c) Monocytes Count (MC).

Feature	Diagnoses		Σ	Probability in Class		
	Healthy	Infected		Healthy	Infected	
MC	Low	3	0	3	3/13	0/7
	Normal	8	3	11	8/13	3/7
	High	2	4	6	2/13	4/7
Σ	13	7	20			

(d) Aspartate aminotransferase (AST).

Feature	Diagnoses		Σ	Probability in Class		
	Healthy	Infected		Healthy	Infected	
AST	High	9	5	14	9/13	5/7
	Normal	4	2	6	4/13	2/7
Σ	13	7	20			

(e)Basophils Count (BC).

Feature	Diagnoses		Σ	Probability in Class		
	Healthy	Infected		Healthy	Infected	
BC	High	4	0	4	4/13	0/5
	Normal	9	7	16	9/13	7/7
Σ	13	7	20			

(f)Lactate Dehydrogenase

Feature	Diagnoses		Σ	Probability in Class		
	Healthy	Infected		Healthy	Infected	
LDH	High	8	0	8	8/13	0/7
	Normal	5	7	12	5/13	7/7
Σ	13	7	20			

Table 10. The five test cases to predict the accuracy

2- The testing cases									
Case 1	Low	Normal	High	High	High	Normal	result	Diagnosis	Actual diagnosis
P(Yes)	0.384	0.23	0.15	0.692	0.307	0.384	0.000705	Healthy	x
P(No)	0.428	0.71	0.57	0.714	0	1	0		
case 2	Normal	Normal	High	High	High	Normal	result	Diagnosis	Actual diagnosis
P(Yes)	0.31	0.23	0.15	0.692	0.307	0.384	0.000565	Healthy	x
P(No)	0.29	0.71	0.57	0.714	0	1	0		
case 3	High	Low	Low	Normal	Normal	High	result	Diagnosis	Actual diagnosis
P(Yes)	0.31	0.38	0.23	0.307	0.692	0.615	0.002308	Healthy	√
P(No)	0.29	0	0	0.285	1	0	0		
case 4	Normal	Normal	Normal	High	Normal	Normal	result	Diagnosis	Actual diagnosis
P(Yes)	0.31	0.23	0.615	0.692	0.692	0.384	0.005219	Infected	√
P(No)	0.29	0.71	0.428	0.714	1	1	0.02183		
case 5	High	High	Normal	Normal	Normal	Normal	result	Diagnosis	Actual diagnosis
P(Yes)	0.31	0.38	0.615	0.307	0.692	0.384	0.00381	Healthy	√
P(No)	0.29	0.28	0.428	0.285	1	1	0.00342		

Table 11. The new training dataset after adding the new arrived cases

Patient #	Features of dataset						Diagnosis
	PC	WBC	MC	AST	BC	LDH	
Training dataset after adding the five cases (20 Patients)							
1	Low	Low	Low	High	Normal	Normal	Healthy
2	Low	Low	Normal	High	Normal	High	Healthy
3	Low	High	Normal	High	Normal	Normal	Infected
4	Low	High	Normal	High	High	Normal	Healthy
5	Low	Normal	High	High	Normal	Normal	Infected
6	Low	Normal	Normal	High	Normal	High	Healthy
7	Normal	Low	Low	High	Normal	Normal	Healthy
8	Normal	High	Normal	High	Normal	Normal	Infected
9	Normal	High	Normal	High	High	High	Healthy
10	Normal	Normal	High	High	Normal	Normal	Infected
11	Normal	Normal	High	High	Normal	High	Healthy
12	High	Low	Low	Normal	Normal	Normal	Healthy
13	High	Normal	High	Normal	Normal	Normal	Infected
14	High	Normal	High	Normal	High	High	Healthy
15	High	High	Normal	Normal	Normal	High	Healthy
16	Normal	Low	Normal	High	Normal	High	Healthy
17	Low	High	Normal	High	High	High	Healthy
18	Low	Normal	Normal	High	Normal	Normal	Infected
19	High	High	Normal	Normal	Normal	Normal	Healthy
20	High	Normal	High	Normal	Normal	Normal	Infected

Table 12 shows another five new incoming cases and their classification with the updated model. According to this, the probability of healthy and infected classes are updated to be 17/25 and 8/25, respectively, as shown in Table 13. Table 14 shows the changes made to the frequency distribution tables for the features according to the five additional incoming cases. The model classified 3 cases correctly, and 2 cases were classified incorrectly. The accuracy of the updated model according to these five cases as shown in Table 15 is 0.6. The accuracy doesn't increase, so the five cases will not be added to the training set. The training set will remain as Table 11.

Table 16 shows an additional five cases that needed to be classified by the model. There are 4 cases classified as infected and one case classified as healthy, so the probability of these two classes is updated as shown in Table 17. Table 18(a→f) shows the third update of the frequency distribution tables compiled from the dataset. The accuracy of the updated model is measured by classifying the initial test cases to calculate the values of TP, TN, FP, and FN. As shown in Table 19, the model classified 2 cases correctly and 3 cases incorrectly so that the accuracy of the model will be 0.4. The accuracy decreases so that, the five cases will not be added to the training set.

In summary, this updating strategy ensures that the model learns efficiently by incorporating only beneficial data, avoids unnecessary re-training, and maintains its ability to recall prior knowledge, all while minimizing computational resources. Figure 8 shows the advantages of ADDS. Figure 13 shows the flowchart of EINB.

Table 12. The five new incoming cases to predict the diagnosis

The another five cases								
Case 1	Normal	Low	Normal	High	High	Normal	result	Diagnosis
P(Yes)	0.31	0.38	0.615	0.692	0.31	0.385	0.003878	Healthy
P(No)	0.286	0.00	0.4286	0.714	0	1	0	
case 2	Low	High	High	High	High	Normal	result	Diagnosis
P(Yes)	0.38	0.38	0.15	0.692	0.308	0.385	0.001212	Healthy
P(No)	0.43	0.29	0.571	0.714	0	1	0	
case 3	High	Low	Low	Normal	Normal	High	result	Diagnosis
P(Yes)	0.31	0.38	0.231	0.308	0.692	0.615	0.002327	Healthy
P(No)	0.29	0	0	0.286	1	0	0	
case 4	Normal	Normal	High	High	Normal	Normal	result	Diagnosis
P(Yes)	0.31	0.23	0.154	0.692	0.6923	0.385	0.001309	Infected
P(No)	0.29	0.71	0.571	0.714	1	1	0.02915	
case 5	High	High	Normal	Normal	Normal	Normal	result	Diagnosis
P(Yes)	0.31	0.385	0.615	0.308	0.692	0.385	0.00388	Healthy
P(No)	0.29	0.286	0.429	0.286	1	1	0.00350	

Table 13. The second update of probability of classes

Class	Probability of the class
Healthy	11/20
Infected	9/20

Table 14: The updated frequency distribution tables compiled from the dataset.

(a) Platelet Count (PC).

Feature	Diagnoses		Σ	Probability in Class	
	Healthy	Infected		Healthy	Infected
PC	Low	1	3	1/11	3/9
	Normal	0	2	0/11	2/9
	High	1	2	1/11	2/9
Σ	11	9	20		

(b) White Blood cell (WBC).

Feature	Diagnoses		Σ	Probability in Class	
	Healthy	Infected		Healthy	Infected
WBC	Low	1	0	1/11	0/9
	Normal	3	1	3/11	1/9
	High	1	2	1/11	2/9
Σ	11	9	20		

(c) Monocytes Count (MC).

Feature	Diagnoses		Σ	Probability in Class	
	Healthy	Infected		Healthy	Infected
MC	Low	1	0	1/11	0/9
	Normal	1	3	1/11	3/9
	High	2	0	2/11	0/9
Σ	11	9	20		

(d) Aspartate aminotransferase (AST).

Feature	Diagnoses		Σ	Probability in Class	
	Healthy	Infected		Healthy	Infected
AST	High	1	1	1/11	1/9
	Normal	1	2	1/11	2/9
Σ	11	9	20		

(e) Basophils Count (BC).

Feature	Diagnoses		Σ	Probability in Class	
	Healthy	Infected		Healthy	Infected
BC	High	1	0	1/11	0/9
	Normal	1	1	1/11	1/9
Σ	11	9	20		

(f) Lactate Dehydrogenase (LDH).

Feature	Diagnoses		Σ	Probability in Class	
	Healthy	Infected		Healthy	Infected
LDH	High	1	0	1/11	0/9
	Normal	1	1	1/11	1/9
Σ	11	9	20		

Table 15. The diagnosis of the testing set after update the model.

3-The testing cases									
Case 1	Low	Normal	High	High	High	Normal	result	Diagnosis	Actual diagnosis
P(Yes)	0.353	0.18	0.176	0.647	0.353	0.471	0.000803	Healthy	x
P(No)	0.375	0.75	0.625	0.75	0	1	0		
case 2	Normal	Normal	High	High	High	Normal	result	Diagnosis	Actual diagnosis
P(Yes)	0.29	0.18	0.176	0.647	0.353	0.471	0.000669	Healthy	x
P(No)	0.38	0.75	0.625	0.75	0	1	0		
case 3	High	Low	Low	Normal	Normal	High	result	Diagnosis	Actual diagnosis
P(Yes)	0.35	0.41	0.235	0.353	0.647	0.529	0.002811	Healthy	√
P(No)	0.25	0	0	0.25	1	0	0		
case 4	Normal	Normal	Normal	High	Normal	Normal	result	Diagnosis	Actual diagnosis
P(Yes)	0.29	0.18	0.588	0.647	0.647	0.471	0.004091	Infected	√
P(No)	0.38	0.75	0.375	0.75	1	1	0.02531		
case 5	High	High	Normal	Normal	Normal	Normal	result	Diagnosis	Actual diagnosis
P(Yes)	0.35	0.412	0.588	0.353	0.647	0.471	0.00625	Healthy	√
P(No)	0.25	0.25	0.375	0.25	1	1	0.00188		

Table 16. The five new incoming cases to predict the diagnosis

The another five cases							result	Diagnosis
Case 1	Normal	Normal	Normal	High	Normal	Normal		
P(Yes)	0.31	0.23	0.615	0.692	0.69	0.385	0.005236	Infected
P(No)	0.286	0.71	0.4286	0.714	1	1	0.021866	
case 2	Low	High	High	High	High	Normal		Diagnosis
P(Yes)	0.38	0.38	0.15	0.692	0.308	0.385	0.001212	Healthy
P(No)	0.43	0.29	0.571	0.714	0	1	0	
case 3	High	Normal	Normal	Normal	Normal	Normal		Diagnosis
P(Yes)	0.31	0.23	0.615	0.308	0.692	0.385	0.002327	Infected
P(No)	0.29	0.714	0.429	0.286	1	1	0.008746	
case 4	Normal	Normal	High	High	Normal	Normal		Diagnosis
P(Yes)	0.31	0.23	0.154	0.692	0.6923	0.385	0.001309	Infected
P(No)	0.29	0.71	0.571	0.714	1	1	0.02915	
case 5	High	High	Normal	High	Normal	Normal		Diagnosis
P(Yes)	0.31	0.385	0.615	0.692	0.692	0.385	0.00873	Infected
P(No)	0.29	0.286	0.429	0.714	1	1	0.00875	

Table 17. The third update of probability of classes

Class	Probability of the class
Healthy	14/25
Infected	11/25

Table 18. The frequency distribution tables compiled from the dataset.

(a) Platelet Count (PC).

Feature	Diagnoses		Σ	Probability in Class	
	Healthy	Infected		Healthy	Infected
PC	Low	3	3	3/15	0/11
	Normal	4	4	4/15	4/11
	High	4	4	4/15	4/11
Σ	11	11	22		

(b) White Blood cell (WBC).

Feature	Diagnoses		Σ	Probability in Class	
	Healthy	Infected		Healthy	Infected
WBC	Low	5	5	5/15	0/11
	Normal	3	3	3/15	3/11
	High	3	3	3/15	3/11
Σ	11	11	22		

(c) Monocytes Count (MC).

Feature	Diagnoses		Σ	Probability in Class	
	Healthy	Infected		Healthy	Infected
MC	Low	3	3	3/15	0/11
	Normal	8	1	8/15	1/11
	High	2	1	3/15	1/11
Σ	13	2	15		

(d) Aspartate aminotransferase (AST).

Feature	Diagnoses		Σ	Probability in Class	
	Healthy	Infected		Healthy	Infected
AST	High	1	1	1/15	1/11
	Normal	4	3	4/15	3/11
Σ	5	4	9		

(e) Basophils Count (BC).

Feature	Diagnoses		Σ	Probability in Class	
	Healthy	Infected		Healthy	Infected
BC	High	0	0	0/15	0/11
	Normal	9	1	9/15	1/11
Σ	9	1	10		

(f) Lactate Dehydrogenase

Feature	Diagnoses		Σ	Probability in Class	
	Healthy	Infected		Healthy	Infected
LDH	High	8	8	8/15	0/11
	Normal	3	1	4/15	1/11
Σ	11	1	12		

Table 19. The diagnosis of the test cases of the updated model

4- The testing cases									
Case 1	Low	Normal	High	High	High	Normal	result	Diagnosis	Actual diagnosis
P(Yes)	0.429	0.21	0.143	0.714	0.357	0.429	0.000803	Healthy	x
P(No)	0.273	0.73	0.545	0.727	0	1	0		
case 2	Normal	Normal	High	High	High	Normal	result	Diagnosis	Actual diagnosis
P(Yes)	0.29	0.21	0.143	0.714	0.357	0.429	0.000535	Healthy	x
P(No)	0.36	0.73	0.545	0.727	0	1	0		
case 3	High	Low	Low	Normal	Normal	High	result	Diagnosis	Actual diagnosis
P(Yes)	0.29	0.36	0.214	0.286	0.643	0.571	0.001285	Healthy	√
P(No)	0.36	0	0	0.182	1	0	0		
case 4	Normal	Normal	Normal	High	Normal	Normal	result	Diagnosis	Actual diagnosis
P(Yes)	0.29	0.21	0.571	0.714	0.643	0.429	0.003856	Infected	√
P(No)	0.36	0.73	0.545	0.727	1	1	0.04616		
case 5	High	High	Normal	Normal	Normal	Normal	result	Diagnosis	Actual diagnosis
P(Yes)	0.29	0.429	0.571	0.286	0.643	0.429	0.00308	Infected	x
P(No)	0.36	0.273	0.545	0.273	1	1	0.00649		

EINB has many advantages to address the challenges of IAI can be listed as follow;

- Handling Catastrophic Forgetting: It preserves knowledge acquired during previous iterations, ensuring that crucial information learned earlier is not discarded or lost.
- Ensuring the New Data Quality: The procedure for augmenting the training set with new data depends on assessing the accuracy of the model after new data arrives, so ensuring the model does not integrate data that may decrease its performance or offer no further benefit.
- Handling Concept Drift: The model's parameters are updated following each IUT value rather than being updated for every new data point. This enables the model to process more data simultaneously, facilitating a more precise representation of the latest data distribution. This is essential for efficiently adjusting the model to the new concept. It also helps the model mitigate the noise and fluctuations arising from only examining individual cases.
- Less sensitivity to noise: Modifying the model incrementally based on individual points might result in erratic adjustments, mainly when outliers are present. Batch processing of data diminishes the influence of noise, as model updates rely on aggregated information, resulting in more stable and dependable model modifications.
- Minimizing overfitting: By updating the model after the IUT value of new data, the risk of overfitting to limited or noisy input samples is mitigated, which is more probable in online learning or when processing individual data points sequentially.
- Better Resource Utilization: Rather than maintaining the entire dataset, it preserves and updates only the feature statistics and class counts upon the arrival of new data.
- Recoverability: it may restore its performance following a decline, allowing it to revert to its prior optimal performance despite a reduction in accuracy during specific learning periods.
- The model can generate precise predictions even in a lack of data. It can initially be trained on a limited dataset and then progressively enhance its predictions when new data points are incorporated, rendering it beneficial for rare or emerging diseases such as MPXV.

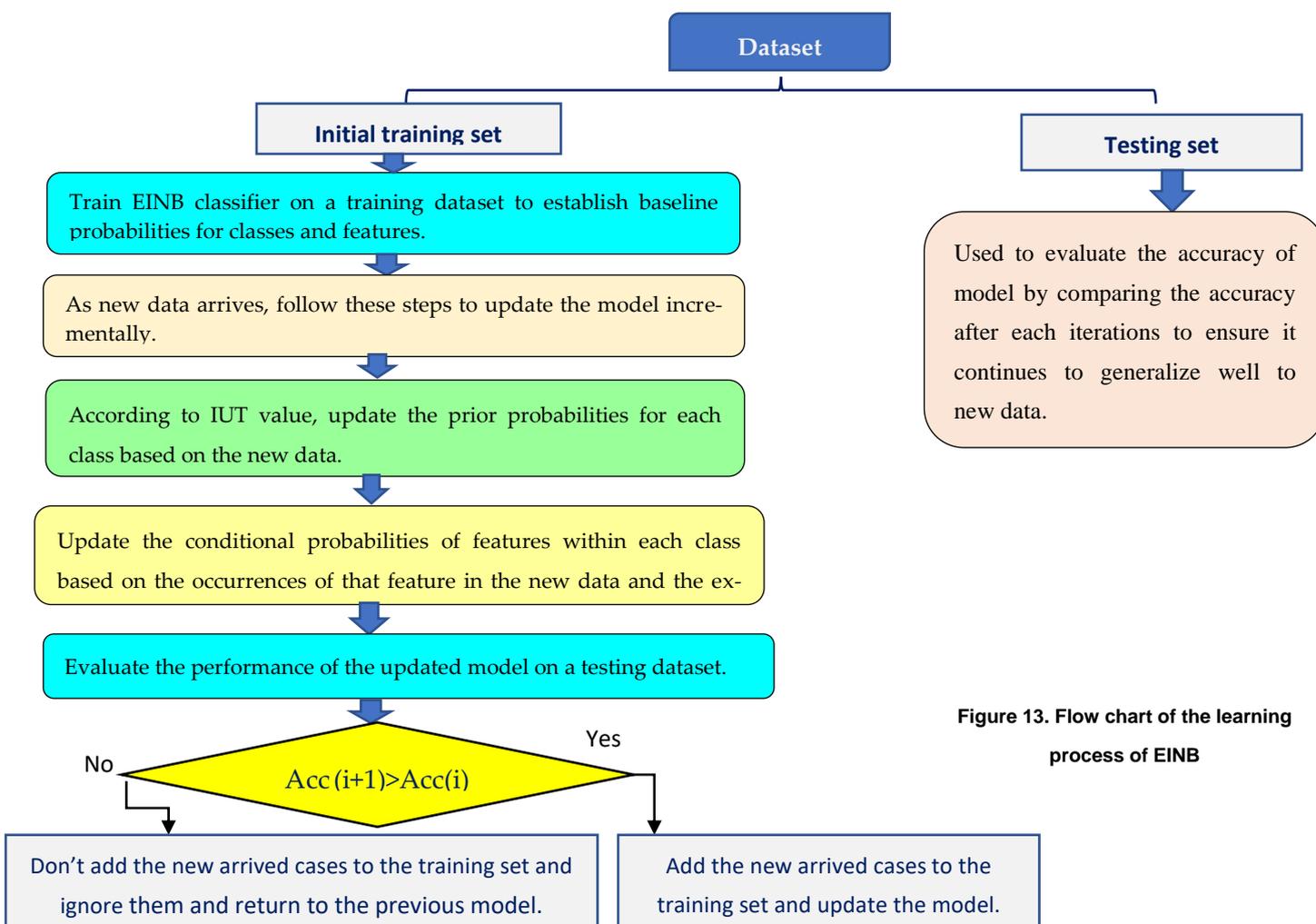


Figure 13. Flow chart of the learning process of EINB

5. Experimental Results

This section provides a detailed discussion of the experimental results, encompassing a description of the dataset and assessment measures employed in the proposed strategy.

5.1. Dataset Description and Experimental Environment

The dataset for this research study was obtained from a publicly accessible ML data repository and used in many researches [21], [26], [28], [31]. The dataset was artificially generated from a report published in the British Medical Journal (The BMJ) titled "Clinical Features and Novel Presentations of Human Monkeypox in a Central London Centre during the 2022 Outbreak: Descriptive Case Series". The dataset comprised 25,000 participants, encompassing 10 boolean and categorical features, along with 1 target variable. The 8 boolean features comprise Rectal Pain, Penile edema, Sore Throat, Solitary Lesion, Oral Lesions, HIV Infection, Swollen Tonsils, and Sexually Transmitted Infections. The two categorical attributes were the patient ID and systemic illness, while the target variable, MonkeyPox, indicated the presence or absence of MPXV in the patient. To ensure that the execution environment of the comparison experiment is standardized, All simulation results in this paper are obtained by Python programming language under the environment of an Intel (R) Core (TM) i7-8550U CPU @ 1.80GHz - 2.00 GHz with 16 GB RAM running Windows 11.

One of the powerful approaches for addressing the class imbalance in the datasets is Synthetic Minority Over-sampling Technique (SMOTE). It operates by creating synthetic samples of the minority class, as opposed to simply replicating existing ones as shown in Figure 14. This enhances the model's performance, particularly in classification tasks where imbalanced data can result in biases. In this paper, SMOTE is used to address the class imbalance in the MPXV dataset.

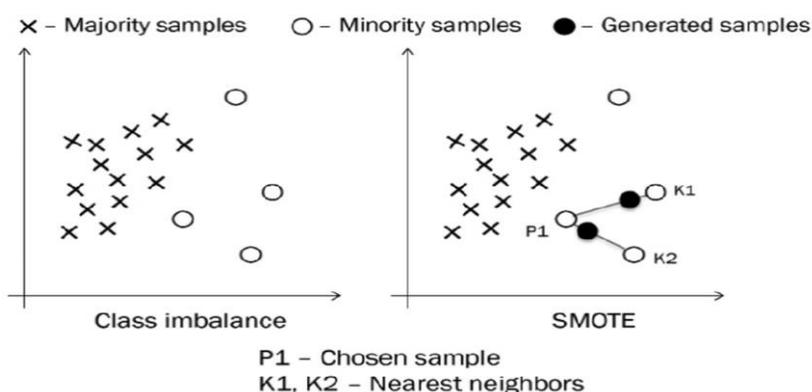


Figure 14. The class balance using SMOTE

5.2. Evaluation metrics

Evaluation metrics are important for evaluating the effectiveness of machine ML algorithms. These quantitative measurements, which include precision, recall, accuracy, and F1-score, are provided to researchers. They enable researchers to identify the most appropriate approach for their tasks by facilitating meaningful comparisons between various models. ADDS is assessed by employing a variety of conventional evaluation metrics, including precision, sensitivity, accuracy, and the F1-score. TP, TN, FP, and FN define these metrics. These metrics are based on the classification confusion matrix of binary decision-making [32].

The fundamental parameter of accuracy quantifies the overall correctness of an ML model. It determines the ratio of cases correctly predicted to the total dataset, as shown in (21).

The proportion of true positive cases that the model correctly identifies is known as recall, which is also known as sensitivity or the true positive rate. Equation (22) illustrates its definition.

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (22)$$

Precision is the proportion of accurate positive predictions to all positive predictions generated by the model, and it can be calculated using (23).

$$\text{Precision} = \text{TN} / (\text{TN} + \text{FP}) \quad (23)$$

The F1 score is a metric that incorporates precision and recall scores and can be calculated using (24).

$$\text{F1 Score} = (2 \times (\text{Precision} \times \text{Recall})) / (\text{Precision} + \text{Recall}) \quad (24)$$

5.3. Testing Adaptive Disease Diagnosis Strategy (ADDS)

This section presents and analyzes the results of the proposed strategy's experiment. The proposed strategy will be evaluated comprehensively through the implementation of three scenarios. At the outset, we will evaluate the performance of conventional classifiers, including NB and K-Nearest Neighbors (KNN), on the dataset regarding critical metrics such as precision, recall, accuracy, and F1 score. In the second scenario, the results will be compared to state-of-the-art methods that utilize advanced models of ML or frameworks that reflect the most recent advancements in the field. The third scenario is using another MPXV dataset to test ADDS.

The most significant features of MPXV are HIV infection, rectal pain, and sexually transmitted infection, which were identified by applying the BGME as a feature selection method. In the first scenario, the NB and KNN will be used as classifiers, and their results will be compared with EINB. Figure 14 illustrates the accuracy, precision, recall, and F1 score of applying the two classifiers to the dataset.

5.3.1. Ablation study of ADDS

This section will involve the removal of some components of the ADDS to assess their impact on the overall performance strategy. Three rounds will be conducted; in the first, remove the proposed EINB and employ traditional NB and KNN to evaluate its impact on the overall performance while retaining the BGME as it is. In the second round, the feature selector (BGME) will be eliminated, permitting all features to participate in the classification process while retaining all components as is. The third round is to remove the feature selector and the EINB and use the classical NB only.

As shown in Figure 15, the EINB model proves to be more effective than KNN and NB in all of the metrics that were assessed in the comparative analysis of model performance. EINB outperforms NB (67%) and KNN (64%), achieving the maximum accuracy at 99.48%. In terms of precision, EINB achieves a value of 99.74%, surpassing KNN at 69% and NB at 68%. In terms of recall, EINB surpasses NB's 92% and KNN's 80%, achieving 99.4% performance. EINB's dominance is further underscored by the F1-score, which achieves a value of 99.57%, in contrast to 78% for NB and 74% for KNN. This score is a delicate equilibrium between precision and recall. Although the NB model exhibits marginally better performance than the KNN model in recall and F1-score, the EINB model consistently outperforms both by a substantial margin across all metrics, highlighting its superior reliability and efficacy for the classification task. The BGME will be removed in the second round and remain the EINB as it is. As shown in Table 20, the value of accuracy, precision, recall, and f1-score are 96.49%, 94.7%, 95.68%, and 95.4% respectively. In the third round both BGME and EINB are eliminated and the values of the evaluation metrics are 61%, 45%, 41.9%, 43.3% for accuracy, precision, recall, and f1-score respectively.

The EINB exhibits exceptional performance in key metrics, including a precision of 99.74%, recall of 99.40%, and an F1-score of 99.57%. This reflects its exceptional capacity to classify positive cases while minimizing errors accurately. The model's high precision suggests that it is highly conservative, effectively preventing false positives. Additionally, the strong recall guarantees that the majority of true positives are detected, thereby minimizing false negatives. The F1-score, which is the harmonic mean of precision and recall, emphasizes a balanced performance, rendering the model dependable for tasks that require consideration of both categories of errors. Furthermore, the model accurately classifies the majority of cases, with only 0.52% of cases misclassified.

fied, with an accuracy of 99.48%. The model's minor preference for precision over recall indicates that it is designed to prevent false positives, which is advantageous in situations where such errors have substantial implications. In general, the findings suggest that the model is highly effective and robust, making it appropriate for applications that necessitate high reliability and minimal error rates.

Table 20. Ablation study of ADDS

Scenario	Accuracy	Precision	Recall	F1-score
BGME and traditional NB	67%	68%	92%	74%
EINB without the BGME	96.49%	94.7%	95.68%	95.4%
Without BGME and EINB	61%	45%	41.9%	43.3%
With BGME and with EINB	99.48%	99.74%	92%	99.57%

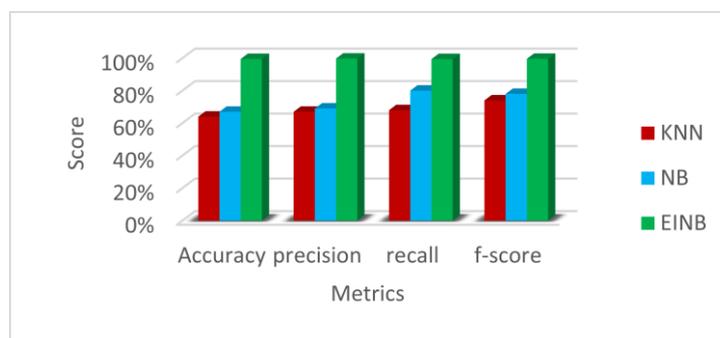


Figure 15. The accuracy, precision, recall, f1-score by applying NB, KNN, and EINB on the dataset

5.3.2. Choosing the proper value of IUT

IUT is a problem-specific parameter, meaning that its optimal value varies based on the dataset and the specific objectives of the analysis. Ideally, the value of IUT should be set high enough to prevent noise (outliers) from significantly affecting the prediction, while also ensuring smoother decision boundaries. A low IUT value could lead to unstable decision boundaries. Conversely, the IUT value should not be set too high, as this could cause one factor to dominate others, giving an unfair advantage to major classes over minor ones. In this section, a sensitivity analysis will be conducted to tune the IUT, a key parameter in the proposed ADDS. The analysis involves testing various IUT values, calculating the classification error for each, and using cross-validation to identify the IUT value that minimizes the error. We claim that (IUT = 130) is the most suitable value for our dataset we are using, as it is derived the minimum value of error in the simulation experiments as shown in Figure 16. Consequently, it will be employed in the experiments that are included in the subsequent subsections, where the performance of the proposed strategy will be assessed.

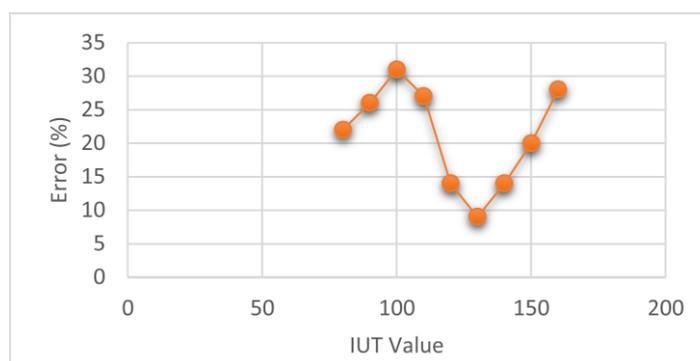


Figure 16. Diagnosis error of ADDS against the different values of IUT.

5.3.3. Testing the ADDS against state-of-the-art MPXV diagnostics methods.

As shown in Figure 17, The ADDS technique is more robust and effective than other methods, as evidenced by its superior accuracy across all training set sizes. At a 70% training size, ADDS obtains an accuracy of 99.48%, which is significantly higher than EMLDL's 85.20%. Accuracy rates of 82.05%, 94.50%, 92.80%, 91.65%, and 86.22% are achieved by other techniques, such as ILA, SEIRD, TL, FLA, and DLF, for the same conditions. ADDS maintains an accuracy of 97.50% when the training set size is reduced by 50%, while EMLDL experiences a significant decrease to 69.70%. The accuracies of ILA, SEIRD, TL, FLA, and DLF also decreased, reaching 77.20%, 96.30%, 91.10%, 92.75%, and 86.22%, respectively. Despite a further reduction to 30% of the training set, ADDS still obtains 96.00% accuracy, which is higher than EMLDL's 96.10%. However, it remains significantly higher than ILA (60.87%), SEIRD (69%), TL (82.60%), FLA (90.6), and DLF (84.07%). ADDS consistently outperforms all techniques, as evidenced by these results, which demonstrate its capacity to maintain high predictive accuracy across varied training set sizes. This emphasizes the adaptability and dependability of ADDS in managing a wide range of datasets.

ADDS's performance is characterized by a remarkable level of precision, which is a critical metric for assessing the proportion of positively identified instances that are accurately identified. ADDS obtains an impressive precision of 99.47% with a 70% training size, significantly surpassing competitive models such as EMLDL (91.3%), ILA (88.1%), SEIRD (78.2%), TL (94.3%), FLA (93.3%), and DLF (89.2%) as shown in Figure 18. EMLDL's precision is reduced to 71.60%. In comparison, ADDS maintains a robust precision of 97%, even when the training size is reduced to 50%. The other models—ILA (83.16%), SEIRD (71.9%), TL (94.87%), FLA (93.3%), and DLF (85.6%)—are unable to keep up. At a training size of 30%, ADDS maintains its lead with 95%, while EMLDL is at 71.3%. The values of the remaining techniques are ILA (60.76%), SEIRD (70.8%), TL (79.5%), FLA (82.2%), and DLF (81%). These results illustrate ADDS's remarkable capacity to reduce false positives, rendering it an optimal selection for applications requiring elevated precision and accuracy.

Figure 19 and Figure 20 show the efficacy of various techniques regarding recall and F-score over varying training set sizes, providing insights into their prediction capacities under different data constraints. Recall, or sensitivity, measures a model's capacity to detect all positive instances. In contrast, the F-score is the harmonic mean of precision and recall and offers a balanced evaluation of these metrics. At a training set size of 70%, ADDS attains near-optimal performance (Recall = 99.4%, F-score = 99.75%), indicating remarkable prediction dependability. Likewise, FLA (Recall = 92.78%, F-score = 91.3%) and TL (Recall = 91.56%, F-score = 93.16%) exhibit high performance, while DLF also demonstrates (Recall = 87.65%, F-score = 90.40%). EMLDL demonstrates moderate efficiency with a Recall of 84.30% and an F-score of 87.70%; ILA shows a Recall of 82.30% and an F-score of 85.10%; SEIRD exhibits a Recall of 84.70% and an F-score of 81.20%. At a training set size of 50%, ADDS has the highest reliability (Recall = 96.00%, F-score = 96.39%), succeeded by FLA (Recall = 92.80%, F-score = 90.80%) and TL (Recall = 90.60%, F-score = 92.80%). DLF has robust performance (Recall = 87.42%, F-score = 85.90%), whereas EMLDL (Recall = 78.30%, F-score = 78.70%), ILA (Recall = 79.70%, F-score = 81.39%), and SEIRD (Recall = 85.60%, F-score = 78.20%) demonstrate moderate efficacy. At a training set size of 30%, ADDS consistently surpasses its competitors (Recall = 94%, F-score = 94.49%), while FLA (Recall = 90.60%, F-score = 86.20%) and TL (Recall = 82.50%, F-score = 81%) exhibit comparable efficacy. DLF exhibits stability (Recall = 85.30%, F-score = 83.11%). Conversely, EMLDL has variable performance (Recall = 87%, F-score = 78.40%), but ILA (Recall = 61.42%, F-score = 61.09%) and SEIRD (Recall = 86.80%, F-score = 78.02%) demonstrate considerable decreases, underscoring their diminished efficacy with fewer datasets.

As shown in Figure 21, ADDS consistently demonstrates superior performance across all dataset sizes, attaining near-optimal metrics even with diminished training data. This dependability renders it a good option for applications necessitating elevated sensitivity and equitable prediction accuracy. Techniques such as FLA and TL demonstrate efficacy, though with slight reductions under data limitations. The performance variability of EMLDL, ILA, and SEIRD highlights their reliance on extensive datasets for optimal operation, rendering them

less appropriate for contexts with restricted data availability. Table 21 summarizes the values of evaluation metrics for all techniques according to different sizes of training sets.



Figure 17. The accuracy results for different techniques according to different training sets



Figure18. The precision results for different techniques according to different training sets



Figure 19. The recall results for different techniques according to different training sets

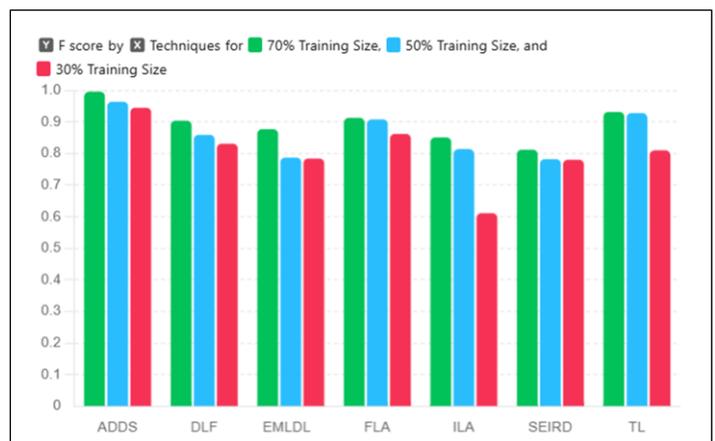


Figure 20. The F-score results for different techniques according to different training sets

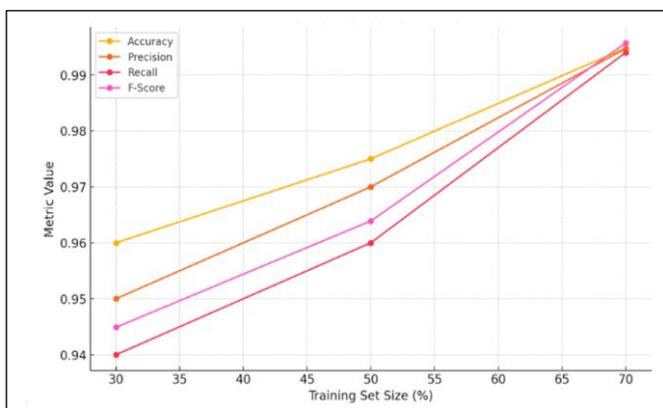


Figure 21. The values of evaluation metrics of ADDS according to different training sets

Table 21. The values of evaluation metrics for all techniques according to different size of training sets

Technique	Training set (%)	Accuracy	Precision	Recall	F score
ADDS	70	0.9948	0.9947	0.9940	0.9957
	50	0.9750	0.9700	0.9600	0.9639
	30	0.9600	0.9500	0.9400	0.9449
EMLDL	70	0.8520	0.9130	0.8430	0.8770
	50	0.6970	0.7160	0.7830	0.7870
	30	0.6910	0.7130	0.8700	0.7840
ILA	70	0.8205	0.8810	0.8230	0.8510
	50	0.7720	0.8316	0.7970	0.8139
	30	0.6087	0.6076	0.6142	0.6109
SEIRD	70	0.7590	0.7820	0.8470	0.8120
	50	0.6930	0.7190	0.8560	0.7820
	30	0.6900	0.7080	0.8680	0.7802
TL	70	0.9450	0.9483	0.9156	0.9316
	50	0.9110	0.9487	0.9060	0.9280
	30	0.8260	0.7950	0.8250	0.8100
FLA	70	0.9280	0.9330	0.9278	0.9130
	50	0.9275	0.9330	0.9280	0.9080
	30	0.9060	0.8220	0.9060	0.8620
DLF	70	0.9165	0.8920	0.8765	0.9040
	50	0.8622	0.8560	0.8742	0.8590
	30	0.8407	0.8100	0.8530	0.8311

5.3.4. Computational Cost of ADDS

The computational cost of the ADDS is primarily determined by four factors: the number of search agents (N), the dimensionality of the problem (D), the number of iterations (p), and the number of hops (h) conducted within each iteration. The computational complexity can be expressed by (25);

$$\text{ADDS's computational cost} = O(N * D * P * h) \quad (25)$$

It is crucial to achieve high accuracy in medical diagnosis, particularly in sensitive applications like MPXV. Although computational efficiency remains valuable, the primary concern is the accuracy and reliability of the results, rather than the execution time. Table 22 shows the values of hyperparameter used in ADDS.

Table 22. The hyperparameter values of ADDS

Parameter	Description	Implemented Value
N	Number of BGME'S agents	50
P	Total number of Iterations in BGME	100
hop (h)	Number of hops	2
K	Value defines the shape of spiral	Each agent has its value
ξ	Factor between zero and one	[0,1]
α	Random value	0.4
Θ	Angle from 0 to 2π .	0 to 2π
Φ	Angle from 0 to π .	0 to π
μ	Shrinking ratio to reduce the radius (R)	0.3
IUT	Incremental Update Trigger	130

5.3.5. Testing the BGME against other feature selection methods.

In this section, the BGME is compared with other feature selection techniques such as; Grey Wolf Optimizer(GWO) [33], Particle Swarm Optimization(PSO)[34], Red Piranha Optimization(RPO)[35], Leopard Seal Optimization(LSO)[36].At training size of 70% of dataset ,the accuracy of KNN in (6) is used to evaluate the performance. Table 23 shows the values of accuracy, recall, precision, f1-score for BGME and these algorithms.

As shown in Table 23, the BGME is outperformed the GWO, PSO, RPO, and LSO in terms of accuracy, precision, recall, and F1-Score. It achieves 96.4% for accuracy, 94.76% for precision, 95.41% and 95.08% for recall and f1-score respectively.

Table 23. The values of evaluation metrics for BGME and different optimization algorithms

Technique	Accuracy	Precision	Recall	F1- score
BGME	96.4%	94.76%	95.41%	95.08%
GWO	78%	81%	84%	76%
PSO	71%	69%	67%	72%
RPO	82%	83.33%	87%	81.64%
LSO	86%	89%	91%	90%

5.3.6. Testing the performance of ADDS in another dataset.

The dataset used in [25],is used to evaluate the performance of ADDS and compare it with other MPXV diagnosis strategies. The blood test dataset was gathered from patients of varying ages and genders in various regions of various countries, including the United Kingdom, Spain, and Nigeria. The MPXV dataset comprises 500 cases that were categorized into two different categories: "Positive" and "Negative." Patients who have the virus are classified as positive cases, while those who do not have it are classified as negative cases.

The following Table 24 illustrates the performance comparison of various techniques based on their accuracy, precision, recall, and F1-score.The efficacy levels of the techniques varied across various evaluation metrics. The ADDS method achieved the highest scores, demonstrating remarkable efficacy with an accu-

racy of 99.6%, precision of 99.4%, recall of 99.3%, and an F1-score of 99.35%. This indicates an excellent balance between precision and recall. In contrast, the SEIRD technique demonstrated the lowest accuracy (79.43%), precision (81.04%), recall (83.4%), and F1-score (82.22%), indicating that there is space for improvement in the accurate capture of relevant cases. Other methods, such as EMLDL and ILA, demonstrated intermediate performance. EMLDL exhibited an accuracy (92.63%) but a lower recall (82.9%), while ILA maintained a higher recall (91.7%) but a lower accuracy (86.77%), underscoring its ability to identify pertinent instances. TL and FLA techniques consistently exhibited robust performance, with accuracy scores of approximately 96.7% and 94.85%, respectively, indicating a high level of overall reliability. The DLF method also demonstrated strong performance, obtaining an F1-score of 93% and an accuracy of 93.78%, despite being slightly behind TL and FLA. HMD provides 98.48%, 91.1%, 88.9%, and 90% for accuracy, precision, recall, and F1-score respectively and the values of these metrics for ADMS are 99%, 91.65%, 91%, and 91.32% which indicates a good results of ADMS. In general, ADDS is the most effective and well-balanced technique that has been assessed.

Table 24. The values of performance metrics of ADDS and different MPXV diagnosis Strategies.

Technique	Accuracy	Precision	Recall	F1- score
ADDS	99.5%	99.4%	99.3%	99.35%
EMLDL	92.63%	88.1%	82.9%	85.5%
ILA	86.77%	90.3%	91.7%	91%
SEIRD	79.43%	81.04%	83.4%	82.22%
TL	96.7%	95.32%	94.82%	95.07%
FLA	94.85%	93.6%	94.2%	93.9%
DLF	93.78%	91.6%	94.4%	93%
HMD	98.48%	91.1%	88.9%	90%
ADMS	99%	91.65%	91%	91.32%

6. Conclusion and Future Work

Classical ML models are trained on static, well-labeled data, but real-world environments are constantly changing. In healthcare, databases are updated daily, making it difficult to process all data at once. Even with traditional algorithms, managing continuously changing data is a challenge. Continual learning, which allows models to incrementally acquire information from non-stationary data streams, is crucial for handling large datasets, adapting over time, and reducing retraining costs. This paper presents ADDS, a strategy for early MPXV detection. ADDS uses the GME optimization algorithm to select significant features, improving diagnostic efficiency. The EINB algorithm processes continuous data, learns incrementally, and adapts over time. The dynamic checkpoint value, ξ , determines the amount of new data integrated, updating the training set only if accuracy improves, and ensuring adaptive learning. ADDS achieves 99.46% accuracy in MPXV detection, offering an efficient solution for evolving medical datasets. Future work could explore integrating GME with other feature selection methods, validating the model with larger datasets, and investigating ensemble incremental learning classifiers to improve diagnostic accuracy.

References

- [1] A. Bamaqa, W. M. Bahgat, Y. AbdulAzeem, H. M. Balaha, M. Badawy, and M. A. Elhosseini, "Early detection of monkeypox: Analysis and optimization of pretrained deep learning models using the Sparrow Search Algorithm," *Results Eng.*, vol. 24, no. September, p. 102985, 2024, doi: 10.1016/j.rineng.2024.102985.

- [2] M. M. Ahsan, S. A. Luna, and Z. Siddique, "Machine-Learning-Based Disease Diagnosis : A," *Healthcare*, pp. 1–30, 2022.
- [3] J. G. Richens, C. M. Lee, and S. Johri, "with causal machine learning," *Nat. Commun.*, no. 2020, pp. 1–9, doi: 10.1038/s41467-020-17419-7.
- [4] J. Zhong, Z. Liu, Y. Zeng, L. Cui, and Z. Ji, "A Survey on Incremental Learning," no. Cape, pp. 166–174, 2017.
- [5] D. Appasani, C. S. Bokkissam, and S. Surendran, "An Incremental Naive Bayes Learner for Real-time Health Prediction," *Procedia Comput. Sci.*, vol. 235, pp. 2942–2954, 2024, doi: 10.1016/j.procs.2024.04.278.
- [6] H. He, S. Chen, K. Li, and X. Xu, "Incremental learning from stream data," *IEEE Trans. Neural Networks*, vol. 22, no. 12 PART 1, pp. 1901–1914, 2011, doi: 10.1109/TNN.2011.2171713.
- [7] K. Raghuraman, M. Senthurpandian, M. Shanmugasundaram, Bhargavi, and V. Vaidehi, "Online Incremental Learning Algorithm for anomaly detection and prediction in health care," *2014 Int. Conf. Recent Trends Inf. Technol. ICRTIT 2014*, 2014, doi: 10.1109/ICRTIT.2014.6996092.
- [8] Y. Luo, L. Yin, W. Bai, and K. Mao, "An Appraisal of Incremental Learning Methods," 2020, doi: 10.3390/e22111190.
- [9] N. A. Mansour, M. S. Saraya, and A. I. Saleh, "Groupers and moray eels (GME) optimization: a nature-inspired metaheuristic algorithm for solving complex engineering problems," *Neural Comput. Appl.*, vol. 0123456789, 2024, doi: 10.1007/s00521-024-10384-y.
- [10] Z. Huo, B. Gu, and H. Huang, "Large Batch Optimization for Deep Learning Using New Complete Layer-Wise Adaptive Rate Scaling," *35th AAAI Conf. Artif. Intell. AAAI 2021*, vol. 9A, pp. 7883–7890, 2021, doi: 10.1609/aaai.v35i9.16962.
- [11] K. M. Amekoe, M. Lebbah, G. Jaffre, H. Azzag, and Z. C. Dagdia, "Evaluating the Efficacy of Instance Incremental vs. Batch Learning in Delayed Label Environments: An Empirical Study on Tabular Data Streaming for Fraud Detection," 2024, [Online]. Available: <https://arxiv.org/abs/2409.10111v1>
- [12] S. I. Hossain *et al.*, "Exploring convolutional neural networks with transfer learning for diagnosing Lyme disease from skin lesion images," *Comput. Methods Programs Biomed.*, vol. 215, pp. 1–27, 2022, doi: 10.1016/j.cmpb.2022.106624.
- [13] M. Altun, H. Gürüler, O. Özkaraca, F. Khan, J. Khan, and Y. Lee, "Monkeypox Detection Using CNN with Transfer Learning," *Sensors*, vol. 23, no. 4, 2023, doi: 10.3390/s23041783.
- [14] A. Alqahtani, S. Alsubai, M. Sha, L. Vilcekova, and T. Javed, "Cardiovascular Disease Detection using Ensemble Learning," *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/5267498.
- [15] Q. Dou *et al.*, "Federated deep learning for detecting COVID-19 lung abnormalities in CT: a privacy-preserving multinational validation study," *npj Digit. Med.*, vol. 4, no. 1, 2021, doi: 10.1038/s41746-021-00431-6.
- [16] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, and Á. Fernández-Leal, "Human-in-the-loop machine learning: a state of the art," *Artificial Intelligence Review*, vol. 56, no. 4. pp. 3005–3054, 2023. doi: 10.1007/s10462-022-10246-w.
- [17] E. Camargo, J. Aguilar, Y. Quintero, F. Rivas, and D. Ardila, "An incremental learning approach to prediction models of SEIRD variables in the context of the COVID-19 pandemic," *Health Technol. (Berl.)*, vol. 12, no. 4, pp. 867–877, 2022, doi: 10.1007/s12553-022-00668-5.
- [18] M. Sirshar, T. Hassan, M. U. Akram, and S. A. Khan, "An incremental learning approach to automatically recognize pulmonary diseases from the multi-vendor chest radiographs," *Comput. Biol. Med.*, vol. 134, no. May, p. 104435, 2021, doi: 10.1016/j.combiomed.2021.104435.
- [19] A. Gepperth and B. Hammer, "Incremental learning algorithms and applications," *ESANN 2016 - 24th Eur. Symp. Artif. Neural Networks*, pp. 357–368, 2016.
- [20] Z. Ahmad, S. Rahim, M. Zubair, and J. Abdul-Ghafar, "Artificial intelligence (AI) in medicine, current applications and future role with special emphasis on its potential and promise in pathology: present and future impact, obstacles including costs and acceptance among pathologists, practical and philosoph," *Diagn. Pathol.*, vol. 16, no. 1, pp. 1–16,

- 2021, doi: 10.1186/s13000-021-01085-4.
- [21] F. N. F. R. Dea Louisa B. Magsino, Russel Lenard O. Mercado and Vincent P. C. MaSheila A. Magboo, "Enhancing Monkeypox Detection: A Machine Learning Approach to Symptom Analysis and Disease Prediction," in *Artificial Intelligence Applications and Innovations*, 2024, pp. 57–67. doi: https://doi.org/10.1007/978-3-031-63211-2_5.
- [22] D. Kundu *et al.*, "Federated Deep Learning for Monkeypox Disease Detection on GAN-Augmented Dataset," *IEEE Access*, vol. 12, no. March, pp. 32819–32829, 2024, doi: 10.1109/ACCESS.2024.3370838.
- [23] R. Pramanik, B. Banerjee, G. Efimenko, D. Kaplun, and R. Sarkar, "Monkeypox detection from skin lesion images using an amalgamation of CNN models aided with Beta function-based normalization scheme," *PLoS One*, vol. 18, no. 4 April, pp. 1–21, 2023, doi: 10.1371/journal.pone.0281815.
- [24] A. I. Saleh, A. H. Rabie, S. E. ElSayyad, A. E. Takieldeem, and F. Khalifa, "An optimized ensemble grey wolf-based pipeline for monkeypox diagnosis," *Sci. Rep.*, vol. 15, no. 1, p. 3819, 2025, doi: 10.1038/s41598-025-87455-0.
- [25] A. I. Saleh and A. H. Rabie, "Human monkeypox diagnose (HMD) strategy based on data mining and artificial intelligence techniques," *Comput. Biol. Med.*, vol. 152, no. November 2022, p. 106383, 2023, doi: 10.1016/j.compbiomed.2022.106383.
- [26] A. Muhammed Kalo Hamdan and D. Ekmekci, "Prediction of monkeypox infection from clinical symptoms with adaptive artificial bee colony-based artificial neural network," *Neural Comput. Appl.*, vol. 36, no. 22, pp. 13715–13730, 2024, doi: 10.1007/s00521-024-09782-z.
- [27] A. Farzipour, R. Elmi, and H. Nasiri, "Detection of Monkeypox Cases Based on Symptoms Using XGBoost and Shapley Additive Explanations Methods," *Diagnostics*, vol. 13, no. 14, 2023, doi: 10.3390/diagnostics13142391.
- [28] E. H. I. Eliwa, A. M. El Koshiry, T. Abd El-Hafeez, and H. M. Farghaly, "Utilizing convolutional neural networks to classify monkeypox skin lesions," *Sci. Rep.*, vol. 13, no. 1, pp. 1–20, 2023, doi: 10.1038/s41598-023-41545-z.
- [29] M. M. Eid *et al.*, "Meta-Heuristic Optimization of LSTM-Based Deep Network for Boosting the Prediction of Monkeypox Cases," *Mathematics*, vol. 10, no. 20, pp. 1–20, 2022, doi: 10.3390/math10203845.
- [30] N. A. Mansour, A. I. Saleh, M. Badawy, and H. A. Ali, "Accurate detection of Covid-19 patients based on Feature Correlated Naïve Bayes (FCNB) classification strategy," *J. Ambient Intell. Humaniz. Comput.*, vol. 13, no. 1, pp. 41–73, Jan. 2022, doi: 10.1007/s12652-020-02883-2.
- [31] S. J. R. da Silva, A. Kohl, L. Pena, and K. Pardee, "Clinical and laboratory diagnosis of monkeypox (mpox): Current status and future directions," *iScience*, vol. 26, no. 6, p. 106759, 2023, doi: 10.1016/j.isci.2023.106759.
- [32] H. A. Sakr, M. M. Fouda, A. F. Ashour, A. Abdelhafeez, M. I. El-Affifi, and M. Refaat Abdellah, "Machine learning-based detection of DDoS attacks on IoT devices in multi-energy systems," *Egypt. Informatics J.*, vol. 28, no. August, p. 100540, 2024, doi: 10.1016/j.eij.2024.100540.
- [33] B. Tu, F. Wang, Y. Huo, and X. Wang, "A hybrid algorithm of grey wolf optimizer and harris hawks optimization for solving global optimization problems with improved convergence performance," *Sci. Rep.*, vol. 13, no. 1, pp. 1–26, 2023, doi: 10.1038/s41598-023-49754-2.
- [34] A. G. Gad, *Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review*, vol. 29, no. 5. Springer Netherlands, 2022. doi: 10.1007/s11831-021-09694-4.
- [35] A. H. Rabie, A. I. Saleh, and N. A. Mansour, "Red piranha optimization (RPO): a natural inspired meta-heuristic algorithm for solving complex optimization problems," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 6, pp. 7621–7648, 2023, doi: 10.1007/s12652-023-04573-1.
- [36] A. H. Rabie, N. A. Mansour, and A. I. Saleh, "Leopard seal optimization (LSO): A natural inspired meta-heuristic algorithm," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 125, no. June, p. 107338, 2023, doi: 10.1016/j.cnsns.2023.107338.