Journal of Immunology and Infectious Diseases
Volume 12 | Issue 1
ISSN: 2394-6512

Research Article Open Access

Accurate prediction of Monkeypox Disease Cases by Analyzing Symptoms with Ensemble learning Boosting Methods and using ML Techniques

Mujeebu Rehman[1], Qinghua Liu[2,*], Ali Ghulam[2]

***Corresponding Author:** Qinghua Liu, School of Information and Communication Engineering, Guilin University of Electronic Technology, Guilin, China, Tel.: 13097730947, E-mail: qhil@gut.edu.cn

Citation: Mujeebu Rehman, Qinghua Liu, Ali Ghulam et al. (2025) Accurate prediction of Monkeypox Disease Cases by Analyzing Symptoms with Ensemble learning Boosting Methods and using ML Techniques, J Immunol Infect Dis 12(1): 102

Received Date: January 05, 2025 **Accepted Date:** February 05, 2025 **Published Date:** February 10, 2025

Abstract

The monkeypox virus is the virus that causes mpox, often known as monkeypox. It can lead to symptoms such as fever, enlarged lymph nodes, and a severe rash. The majority of patients experience full recovery; however a few may get seriously ill. Transmission of monkeypox from animals to humans happens by bites, scratches, or actions including hunting, skinning, trapping, cooking, handling carcasses, or consuming animals. Identifying and analyzing the monkeypox virus is of great significance. This study conducted a comparative analysis of ensemble learning techniques, specifically focusing on boosting models using datasets related to monkeypox. In this study, we conducted an analysis on a publically accessible dataset by subjecting it to evaluation using five pre-existing ensemble learning techniques, specifically focusing on boosting models. For the boosting classification of the monkey pox disease, five different classification models are used: Adaptative boosting (AdaBoost), Gradient Boosting Machines (GBM), Extreme Gradient Boosting Machine (XGBM), Light Gradient Boosting Machine (LGBM), Cat Boost Classifier (CBC). Four assessment measures were employed in this study to compute the classification accuracy, which involves F-Score, Accuracy, Precision, and Recall. Five ensemble learning boosting models were employed to classify the training model. Among them, the AdaBoost model demonstrated superior performance in terms of both time consumption and accuracy, with a score of 97.67%. GBM model obtain accuracy score 93.02%. The classification accuracy of the Extreme Gradient Boosting Machine (XGBM) was found to be 95.34%, Light Gradient Boosting Model (LGBM) (achieved accuracy of 93.02%), Cat Boost Model achieved accuracy of 93.02%. The AdaBoost method demonstrated the highest level of accuracy for this specific task, resulting in an approximate accuracy rate of 97.67%.

Keywords: Machine learning Ensemble Methods; Bioengineering; Monkeypox; ML Techniques; Viral Disease

Introduction

The monkeypox virus was initially identified in Denmark in 1958 in monkeys used for research. Despite the global disturbance caused by the outbreak of Covid-19 in 20020, the occurrence of Monkeypox in 2022 has brought to light the arrival of another highly prevalent virus. In order to distinguish Monkeypox disease from other der-matological conditions with similar characteristics, such as chicken pox and measles, it is imperative to accurately identify it. The utilization of artificial intelligence (AI) tools could aid in the identification of viruses through the application of virus image processing and analysis. Various diseases are currently posing a threat to public health, peace, and safety worldwide. Early diagnosis and detection of monkeypox are crucial for efficient treatment and prevention of further disease transmission [1, 2]. Since May 2022, there have been documented instances of massive monkeypox epidemics in many places worldwide [3, 4, 5]. The symptoms of the viral zoonotic disease monkeypox are comparable to those of smallpox patients [6]. The primary etiology of the disease is attributed to infection with the DNA viruses that cause orthopox and monkeypox [7]. The orthopoxvirus genus utilizes many techniques to circumvent the host's immune defenses. enabling the virus to infiltrate the host's systems without being identified or recognized [8]. Africa is home to two separate strains of the Monkeypox Virus (MPXV), namely clade I, which is prevalent in central Africa, and clade II, which is found in western Africa [9]. In contrast to smallpox and chickenpox viruses, which are transmitted solely through direct interpersonal contact with an affected individual, MPXV has the potential to be transmitted between animals and humans through the exchange of blood and other bodily fluids.

The transmission of Monkeypox primarily occurs through direct contact with infected individuals or con-taminated objects, and the spread is further exacerbated by factors such as high population density and unrestricted international travel [10-12]. The primary diagnostic methods, including the polymerase chain reaction (PCR) technique and electron microscopy examination of skin lesions, while accurate, are often time-consuming and resource-intensive [13]. These challenges highlight the urgent need for faster, more accessible diagnostic solutions. Recent research by Sitaula and Shahi (2022) [14] examined and compared 13 pre-trained deep learning models to identify the Monkeypox virus. However, these approaches heavily depend on high-quality patient photos, which may not always be available, especially in resource-limited settings. Similarly, Khafaga et al. [15] employed a deep convolutional neural network and achieved a high accuracy of 0.98, yet this method is also restricted by the need for extensive image datasets. Bala et al. [16] developed MonkeyNet using the MSID (Monkeypox Skin Images Dataset), building a modified DenseNet-201 deep neural network, while Ahsan et al. [17] created the GRA-TLA model, combining Transfer Learning techniques for classification tasks. Although these image-based models show promise, their reliance on visual data restricts their application in scenarios where rapid or early symptom-based diagnosis is critical. In contrast, predictive modeling approaches focus on forecasting case numbers but are less applicable to direct diagnosis. Marwa Eid et al. (2022) [18] introduced a novel approach using optimized Long Short-Term Memory (LSTM) models trained on tissue data to forecast confirmed monkeypox cases, which, although innovative, does not address immediate diagnostic needs. Iftikhar et al. [19] proposed a method for predicting monkeypox cases using machine learning models applied to trend and residual subseries of time series data. Similarly, Bhosale et al. [20] conducted a study using time-series data analysis for epidemic prediction. However, these methods prioritize trend analysis over direct symptom-based diagnostic capabilities, limiting their practical use in clinical settings. Kumar Mandal et al. [21] combined machine learning with Particle Swarm Opti-mization (PSO) to analyze monkeypox cases, demonstrating a computational algorithm inspired by biology to find optimal solutions [22]. Despite offering valuable insights, PSO and similar complex algorithms are computationally intensive and less suited for real-time or accessible diagnostics. These limitations underscore the critical need for novel approaches that can bypass the dependency on images and computationally demanding processes, paving the way for more accessible, symptom-based diagnostic models.

Furthermore, it is imperative to conduct thorough validation and testing of these algorithms to guarantee their reliability and accuracy prior to their use in real-world scenarios. This study conducted a comparative analysis of ensemble learning techniques, specifically focusing on boosting models using datasets related to monkeypox. Traditional image-based diagnostic meth-

ods often face challenges, such as the need for high-quality images, limited accessibility, and high computational costs, which can hinder timely and accurate diagnosis. To summarize, this research showcases the capability of employing ensemble learning boosting methods to identify monkeypox by analyzing symptoms rather than relying on photos. The XGBoost model and AdaBoost model have demonstrated favorable outcomes in terms of accuracy and possess the capability to be employed in practical scenarios to enhance the speed and precision of monkeypox diagnosis. Among the five models tested, AdaBoost showed superior performance with an accuracy rate of 97.67%. However, additional investigation and verification are necessary to guarantee the dependability and precision of this model prior to its practical application. As far as the author knows, it is the first method of diagnosing monkeypox illness based on symptoms.

The key contributions of this work are listed as follows:

1.Generating a dataset based on symptoms by utilizing published reports of monkeypox sickness;

2.Introducing the inaugural prototype for diagnosing monkeypox only relying on symptomatology;

3.Applying comparison to analyze the results of the ensemble learning boosting methods;

4.Assessing and contrasting various ensemble learning boosting methods, namely AdaBoost, GBM, XGBM, LGBM, and CBC.

For the boosting classification of the monkeypox disease, five different classification models are used: Adaptive boosting (AdaBoost), Gradient Boosting Machines (GBM), Extreme Gradient Boosting Machine (XGBM), Light Gradient Boosting Machine (LGBM), and Cat Boost Classifier (CBC). Four assessment measures were employed in this study to compute the classification accuracy, which involves F-Score, Accuracy, Precision, and Recall. These findings underscore the importance of leveraging ensemble learning boosting techniques in symp-tom-based classification of monkeypox, marking a step forward in non-image-based diagnostic approaches Five ensemble learning boosting models were employed to classify the training model.

Method and Data Source

Data Sets Collection

The dataset utilized in this research is publicly available on Kaggle, uploaded by the user 'Larxel', and is labeled "Global Monkeypox Cases (daily updated)" [23]. The data is gathered by the organization "Global Health" and utilized by the "World Health Organization". This dataset provides a chronological record of confirmed instances in relation to their corresponding dates. Additionally, it includes supplementary information for each reported instance [24]. The dataset exhibits a wide range of symptoms that lack a distinct and organized framework. Moreover, the sickness is not influenced by cities or countries. After analyzing the data, we extracted only two specific columns, namely "Symptoms" and "Status". Consequently, we created a new dataset using this selected information. This was accomplished by generating columns for each symptom present, assigning a value of 1 if the patient exhibited the symptom and 0 if not. Due to the absence of a standardized structure for documenting patient information, even identical symptoms were documented in varying ways, as the data was renewed on a daily basis. For instance, the term "Rash" is grouped together with other titles like "Rashes", "Rash on the skin", "skin rashes", and so on, due to their similarity, and so they were merged. Ultimately, there were about 46 columns total; one was for identification, 44 were for symptoms, and the last column was for the condition of the disease. Following the specified procedures for data cleaning, only 44 instances exhibited clear and distinguishable characteristics, thereby constituting our dataset. The resulting dataset exclusively consisted of binary values (0 and 1) and did not require any additional preprocessing.

Label Encoded Features Extraction Method

A method used in data analysis and machine learning to portray discrete classification as continuous measures is label encoding. Given that the majority of machine learning models exclusively operate with numerical data, it is highly advantageous to utilize methods that require numerical input. We have dissected the internal mechanisms of label encoding and will now demonstrate how to implement it in Python.

For a clear example, consider a dataset that has information about various diseases. In this dataset, there is a column called "disease 'monkeypox" which includes categorical values such as "positive monkeypox" and "negative monkeypox". Label encoding is a process that turns categorical data into a numerical format by assigning a unique numerical label to each discrete category [25]. The monkeypox virus dataset, used for validation and demonstration purposes, was acquired via Kaggle [26].

Extraction of Features: The monkeypox disease affected the significant characteristics or attributes that were used to differentiate between the two categories, which were extracted from the datasets. Given an input X with 47 independent test set features, only 47 of these features had an impact on the label or target monkeypox labels values. The remaining feature, "ID," is unimportant or uncorrelated. Therefore, we utilized these 10 features for model training [27].

Data Pre-processing is an essential step in transforming data into a useful and effective format for the machine learning algorithm. Data normalization is the initial approach employed for data pre-processing. Next, the employed pre-processing approach is label encoding. This technique is utilized to assess the dependent variable, specifically whether or not an individual has monkeypox. All string values in the output variable are substituted with 0 and 1, which determine the targeted class as shown in Table 1. The dataset included several values for many features, including HIV infection, swollen tonsils, oral lesions, solitary lesions, penile edema, sore throat, rectal pain, and systemic illness. The values that were missing were filled in by replacing them with the median value of a specific property. This data-preprocessing approach is alternatively referred to as median replacement.

Table 1: dataset used in our work 211 ID column, 47 columns of symptoms, labels outcome

ID	rash	skin lesions	headache	ulcerative lesions	blisters on	.	.	.	47	Outcome
1	1	0	0	0	0	0	0	0	0	1
2	1	0	0	0	0	0	0	1	1	1
3	1	0	0	0	0	1	1	0	0	1
4	1	0	0	0	0	0	0	0	0	1
5	1	0	0	0	0	1	0	1	1	1
6	1	0	0	0	0	0	0	0	0	0
.										
.										
209	1	0	0	0	0		0	1	0	0
210	0	1	0	0	0		1	0	1	0
211	0	1	1	0	0		1	0	1	1

Method

Proposed Method

This work aimed to boost, a machine learning method that can turn weak learners to a powerful classifier. An ensemble meta-algorithm is employed to mitigate both bias and variation. On one hand, Weak learners are classifiers that perform only slightly better than random guessing. On the other hand, strong learners are classifiers that achieve high accuracy and form the foundation of boosting ensemble methods [28]. The core concept of boosting entails repeatedly applying the underlying learning algorithm on modified versions of the input data Boosting techniques train a poor learner with input data, Identify the training samples that were categorized wrongly, calculate the predictions made by the weak learner, and then train the next weak learner using an updated training set that includes the previously misclassified cases. The proposed methodology has undergone empirical evaluation using advanced methodologies and base classifiers, including AdaBoost, GBC, LGBM, XGB and CatBoost. This research focuses on enhancing the outcomes and precision of monkey pox disease detection. We have suggested employing an ensemble of machine learning algorithms, utilizing a boosting classifier, to perform binary categorization of diseases as either positive or negative. Data a pretreatment procedure has been performed before to inputting it into the model, and this is subsequently followed by data augmentation.

The proposed framework of the suggested ensemble technique, which utilizes a boosting classifier, is depicted in Figure 1. "AdaBoost, GBC, LGBM, XGB and Cat Boost" are acronyms for boosting methods, which are algorithms used in machine learning. Boosting methods refer to a distributed gradient boosting library that prioritizes speed, adaptability, and user-friendliness. The Gradient Boosting framework is utilized for its execution. It provides a parallel tree boosting technique to efficiently and accurately tackle a broad spectrum of data science problems. Using a series of relatively weak individual classifiers, the ensemble learning technique known as "boosting" creates a final classifier that is both strong and dependable. Boosting algorithms excel in the bias-variance trade-off. Boosting algorithms are considered more efficient than other algorithms that only tackle high variation in a model. This is because boosting handles both bias and variance, which are the two drivers of mistake. Due to its capacity to easily adapt to different sizes and demands, it has recently experienced a surge in popularity and is now widely accepted as the norm for organizing and managing structured data. XGBoost is a faster and more efficient version of gradient boosted decision trees (AdaBoost) that prioritizes speed and efficiency. A predictive model for assessing the risk of Monkeypox virus in affected patients was developed using the XGBoost ensemble approach. This model was then compared to five boosting methods which commonly used boosting machine learning algorithms.

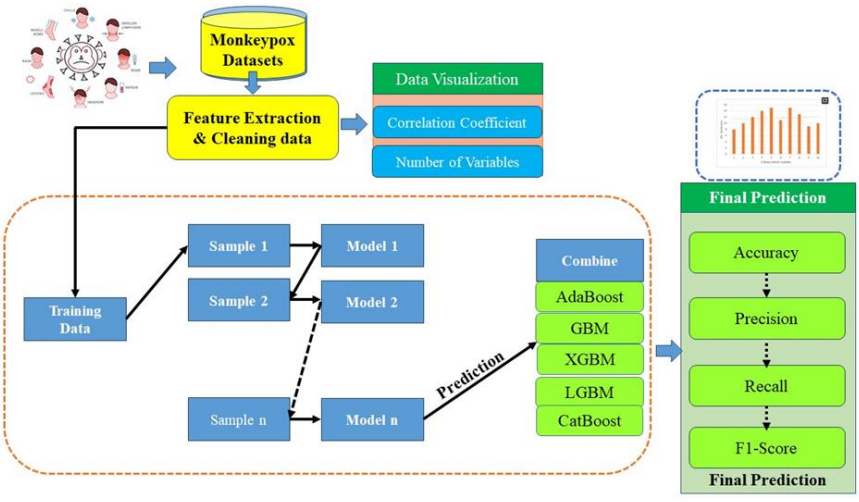


Figure 1: Proposed Framework Ensemble *Learning for Boosting method*

The machine learning technique known as ensemble learning, can be employed for the purpose of classifying monkeypox. This

study conducted a comparative analysis of 1 Ensemble Learning technique, specifically boosting approaches, using datasets related to monkey pox. The results of ensemble learning tests conducted on the monkey pox datasets revealed that the Boosting approach was effective. The AdaBoost model, GBM model, Extreme Gradient Boosting Machine (XGBM), (LGBM) models and Cat Boost Model achieved the greatest accuracy using datasets. This study involved a comparison of ensemble learning approaches, specifically the boosting method, using monkeypox datasets.

Boosting is an ensemble learning technique that utilizes a single base model type and employs adaptive sequential learning, where the results of each base model depend on the outcomes of the preceding base model. These results are then merged to achieve optimal performance. The selected model for boosting ensemble learning is:

AdaBoost (Adaptive Boosting)

Using a basis algorithm, usually a decision tree, a base classifier is trained as part of the AdaBoost ensemble learning technique. The sample weights are modified based on the classifier's predictions, and the revised samples are then used to train the next classifier. Consequently, the samples that were categorized incorrectly are given higher weights, while the samples that were classified correctly are given lower weights. This makes sure that samples that aren't correctly classified will be given more attention by subsequent classifiers.

This paper provides an overview of the critical factors influencing the AdaBoost algorithm gives weight to each item within training set. The weight $D_1(i)$ of each sample x_i and the weight update $D_{1t+1}(i)$ are computed for a collection of labeled training instances, represented as $S=\{(x_1, y_1)_{,...,} (x_i, y_j)_{,...,} (x_m, y_m)\}$, where y_i denotes the target label of sample x_i and y_i belongs to the set Y={-1, +1}. Multiple research have analyzed the consequences of this dataset.

$$D_1(i) = \frac{1}{n}, i = 1, 2, \ldots, m \quad (1)$$

$$D_{t+1}(i) = \frac{D_t(i)}{z_t} \exp(-a_t y_i h_t(x_i)),\ i = 1, 2, \ldots, \quad (2)$$

The function $h_t(x)$ denotes the fundamental classifier, where t=1, …, T represents the total number of iterations. Z_t is a factor used for normalization, the symbol α_t represents the allotted weight for the classifier $h_t(x)$. The weight quantifies the significance of the classifier $h_t(x)$ in determining the ultimate prediction of the classifier. The cases that are mispredicted in are given higher weights in the training cycle. Moreover, is chosen in a way that guarantees the transformation of into a distribution. and αt are derived using the following equations:

$$Z_t = \sum_{t+1}^{n} D_t(i)\ \exp(-a_t y_i h_t(x_i)) \quad (3)$$

$$a_t = \frac{1}{2} In(\frac{1 - \in_t}{\in_t}) \quad (4)$$

The classifier's error rate, represented as t, is obtained using the following method:

$$\in_t = P[h_t(x_i) \neq y_i] = \sum_{i=n}^{n} D_i(i) I[h_t(x_i) \neq y_i] \quad (5)$$

According to recent research, the last strong classifier is calculated using the following methods once the designated amount of iterations have been completed:

$$H(x) = sign(\sum_{t=1}^{T} a_t h_t(x)) \quad (6)$$

Empirical evidence appears to confirm the notion that algorithm 1 provides a concise summary of the AdaBoost method. Implementing AdaBoost is straightforward and requires minimal tuning of its hyperparameters [29]. In addition, AdaBoost is versatile and may utilize many algorithms as the underlying learner. As a result, the underlying learner can be any method that is appropriate for a given application, and AdaBoost can improve its performance. Due to the iterative nature of AdaBoost's learning process, overfitting may occur from noisy data and outliers.

GBM Stands for Gradient Boosting Machines

A machine learning technique called gradient boosting makes advantage of the boosting technique to build strong ensembles. The primary methodology primarily use decision trees as the foundational learner to construct a resilient ensemble classifier, commonly referred to as a gradient boosted decision tree (GBDT). The notion of gradient boosting was initially proposed by Breiman, who observed that boosting may be seen as an optimization method employed on a specific loss function.

Later on, Friedman [30] created an improved gradient boosting technique. The algorithm's learning procedure entails sequentially training new models in order to acquire a resilient classifier. The technique is created incrementally, following a similar approach to previous boosting approaches. However, Its primary goal, though, is to develop base learners with a significant link to the ensemble's loss function's negative gradient [31].

By utilizing a training set S={xi, yi} N1, the gradient boosting technique aims to minimize the given loss function L(y, F(x)) by approximating the function F.(x), which transfers the predictor variables x to corresponding responder variables y. The Gradient Boosted Decision Tree (GBDT) approach builds an additive estimation of the function F(x) by computing a weighted combination of several functions:

$$f_m(x) = F_{m-1}(x) + \rho_m h_m(x) \quad (7)$$

The mth function, $h_m(x)$, has a magnitude denoted by the symbol. Decision tree models within the ensemble are represented by these functions. The approximation is performed iteratively by the algorithm. Meanwhile, a continuous estimation of $F_0(x)$ is obtained using:

$$F_0(x) = argmin_a \sum_{i=1}^{N} L(y_i, a) \quad (8)$$

Subsequent base learners strive to minimize.

$$(\rho_m h_m(x)) = argmin_{p,h} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \rho h(x_i)) \quad (9)$$

Instead of directly solving the problem of optimization, every hm can be viewed as a step of a greedy gradient descent optimization for F*. Thus, each h_m undergoes training using a distinct training set D={xi, r_{mi}} Ni=1, where r_{mi} denotes the false residuals, this is the difference between the actual label and the output of a single base classifier. Pseudo residuals, another name for the term "false residual," are computed using the following formula:

$$r_{mi} = [\frac{\delta L(y_i, F(x))}{\delta F(x)}]_{F(x)=F_{m-1}(x)} \quad (10)$$

Afterwards, the value of is determined by the process of line search optimization. However, if the iterative job is not sufficiently regularized, this technique may suffer from overfitting. When it comes to certain loss functions, like the quadratic loss function, an early iteration termination will take place if the inaccurate residuals reach zero in the subsequent iteration, even if h_m accurately models the incorrect residuals.

In addition, much research has been conducted on various regularization hyperparameters to improve the optimization of the additive learning strategy used by the GBDT. Nevertheless, the inherent method of regularizing the Gradient Boosting Decision Tree (GBDT) involves utilizing shrinkage to restrict each gradient descent step $F_m(x) = F_{m-1}(x) + v\rho_m h_m(x)$, where 0.1 is typically assigned to the value of v [32]. Algorithm 2 provides a concise summary of the gradient boosting technique.

It is generally accepted wisdom that significant benefit of gradient boosting is its ability, similar to other methods for boosting algorithms, to effectively learn intricate patterns seen in the input data. In order to accomplish this, the model undergoes training to rectify errors generated by the preceding model. However, if there is noise in the input data, a model constructed using this technique has the potential to overfit and capture noise, as indicated by references [33] and For applications that use small datasets, this method works well .

XGBM, Short For Extreme Gradient Boosting Machine, Is a Powerful Machine Learning Technique

An ensemble model is produced by the machine learning algorithm known as XGBoost, which mixes decision trees with the gradient boosting framework. This approach is highly scalable and exceptionally precise, making it well-suited for both regression as well as classification applications. XGBoost has emerged as the dominant algorithm in the realm of applied machine learning and has achieved victory in numerous Kaggle tournaments. The approach was created in 2016 by Chen and Guestrin, and it offers significant improvements over the traditional gradient boosting algorithm. The loss function of XGBoost incorporates a regularization term to mitigate overfitting, distinguishing it from gradient boosting [34].

$$L_M(F(x_i)) = \sum_{i=1}^{n} L(y_i, F(x_i) + \sum_{m=1}^{M} \Omega(h_m) \quad (11))$$

The discrepancies between the target variable's actual class and its anticipated class are calculated using a loss function called L(*). While $F(x_i)$ is the forecast for the i-th occurrence during the M-th iteration, on the other hand. A regularization term is denoted by the symbol $\Omega(h_m)$ and is described as follows:

$$\Omega(h) = y^T + \frac{1}{2}\lambda \parallel w\parallel^2 \quad (12)$$

The complexity parameter, represented by the symbol ϒ, is the minimum amount of loss reduction gain needed to divide an internal node. Increasing the value of results in the generation of less complex trees. Furthermore, represents the leaf nodes' output, is the number of leaves in the tree, and is a penalty parameter. The objective function is approximated by a second-order Taylor algorithm in XGBoost, as opposed to first-order derivative employed in GBDT. Hence, Equation 13 is consequently altered as follows:

$$L_M \approx \sum_{i=1}^{n} [g_i f_m(x_i) + \frac{1}{2} h_i f_m^2(x_i)] + \Omega(h_m) \quad (13)$$

Here, gi and hi stand for the loss function's initial and secondary derivatives. To determine the final loss value, We can add together each leaf node's loss values, where Ij represents the leaf node j samples. Therefore, the objective function is expressed as:

$$L_M = \sum_{j=1}^{T} [(\sum_{i \in Ij} g_i) w_j + \frac{1}{2}(\sum_{i \in Ij} h_i + \lambda) w_j^2] + yT \quad (14)$$

In essence, the goal function is approximated quadratically through the optimization process. Furthermore, XGBoost is resistant to overfitting since it incorporates a regularization term [35]. The XGBoost approach, similar to the gradient boosting algorithm, utilizes maximum tree depth, learning rate, and subsampling to avoid overfitting the model.

LGBM Stands for Light Gradient Boosting Machine

Microsoft developers developed LightGBM, a very effective version of the gradient boosting technique, in 2017. [36]. This tool is versatile and may be applied to various machine learning tasks such as classification, ranking, and other similar challenges. To improve training speed and yield better accuracy, the LightGBM algorithm combines two cutting-edge methods: Exclusive Feature Bundling (EFB) and Gradient-based One-Sided Sampling (GOSS). GOSS is a modified gradient boosting technique that considers training examples with greater gradients, resulting in accelerated learning and reduced computational complexity in the model.

In particular, the GOSS technique [only computes the information gain using the remaining training instances after a significant number of cases with tiny gradients are removed. The rationale for removing samples with modest gradients is that examples with substantial gradients are more valuable in computing the information gain (IG). Therefore, the GOSS approach precisely computes the IG with a smaller sample size. The EFB technique decreases the number of characteristics by combining sparse features that are not compatible with each other, hence accomplishing a feature selecting task. There are certain qualities that are very rare in a feature space with few values, indicating that they seldom have simultaneous non-zero values. An exemplary illustration of distinctive characteristics is the utilization of One-hot encoded features. In addition, the EFB approach combines these features in order to decrease the size of the feature matrix.

LightGBM provides a notable benefit in terms of speed and generally yields an extremely efficient model. Moreover, because continuous data are converted into discrete bins, it exhibits a small memory footprint. Furthermore, it attains significantly greater precision in comparison to the majority of boosting algorithms, due to the integration of GOSS and EFB techniques. The LightGBM algorithm is particularly effective when trained with huge datasets, Due to its shorter training period in comparison to the XGBoost algorithm, it is faster. An inherent drawback of LightGBM is its proclivity to overfit small training datasets, as it is specifically designed to be optimized for bigger data sets. Moreover, partitioning the tree according to particular leaves can result in overfitting due to the creation of too complex trees. LightGBM has proven to be highly effective in several classification difficulties, producing exceptional results. The algorithmic technique for LightGBM is outlined in Algorithm 3. For a comprehensive understanding of the LightGBM approach, refer to the description provided in. In addition, a thorough mathematical analysis of the LightGBM method is provided in reference.

CatBoost is a Machine Learning Algorithm

The CatBoost algorithm, developed by Prokhorenkova et al. [37] in 2017, is a gradient boosting solution. The method efficiently manages categorical features throughout the training phase. The ability of CatBoost to do unbiased gradient estimation is a notable improvement as it substantially reduces overfitting. Thus, the CatBoost technique excludes each instance from being utilized to train current models to calculate the gradient during every iteration of boosting. A significant enhancement in the CatBoost method is its automatic conversion of categorical input into numerical values. Categorical characteristics are defined by a finite collection of values known as categories, which are generally not able to be compared. Therefore, these characteristics are currently unsuitable for constructing decision trees. During the preprocessing phase, categorical characteristics are commonly converted into numerical features by replacing them with numerical values. A common technique for handling categorical variables with few unique values is one-hot encoding. The process entails substituting the initial attribute with a binary variable. Nevertheless, CatBoost utilize an enhanced and resilient methodology. that prevents overfitting and ensures the use of all training set samples To train model. This method involves the random rearrangement of the training dataset. The average label value for a sample is calculated for each sample based on the sample that belongs to the same category and occurs earlier in the shuffled sequence. If $\sigma = (\sigma 1, \ldots, \sigma n)$ represents a permutation, then the value of $x\sigma p,k$ is substituted with a value determined by the prior value P and its weight a. Furthermore, it is important to note that the parameter an is greater than zero. In addition, incorporating the previous value and weight in the CatBoost algorithm effectively decreases the noise generated by cate-

gories with low frequency. The CatBoost approach exhibits outstanding performance and outperforms other machine learning algorithms in scenarios where the input comprises category data. Furthermore, it has the capacity to efficiently handle missing data. However, if the settings are not properly changed, the system's performance may be below average.

$$\frac{\sum_{j=1}^{p-1}[x_{\sigma j,k} = x_{\sigma p,k}]y_{\sigma j} + a.p}{\sum_{j=1}^{p-1}[x_{\sigma j,k} = x_{\sigma p,k}]y_{\sigma j} + a} \quad (15)$$

The Decision Tree model is chosen as the base model because to its consistently superior performance in earlier research studies. To divide the data set into training and test sets, a random allocation technique was applied.. ensuring an equal number of positive and negative cases in an 8:2 ratio. We conducted a 5-fold cross-validation on the training set to evaluate the effectiveness of the models and subsequently evaluated their performance on the test set. Five evaluation criteria were used to assess the models: F1-score, accuracy, recall, specificity, and area under the receiver operating characteristic curve (AUC). The continuous features were normalized using the mean and variance of each feature, and any missing values were replaced with the means of the respective features.

Data processing and modeling were conducted using the Python 3.6.5 kernel. These experiments were performed on an isolated intranet Linux server using the anaconda environment manager. Four algorithms were implemented using the Scikit-learn module and the Python programming language. Boosting method is part of ensemble learning techniques in machine learning. Ensemble learning is the process of algorithm for the prediction of monkeypox disease based on several models to enhance prediction accuracy. During boosting, a series of models are trained data. Every model is trained using a training set that has been assigned weights. We allocate weights according on the errors of the preceding models in the series. Samples training involves each model correcting the errors of the previous one. This process continues until the specified number of models has been trained or until other requirements are satisfied. Incorrectly classfied cases during training are given larger weights to prioritize them in the next model training.

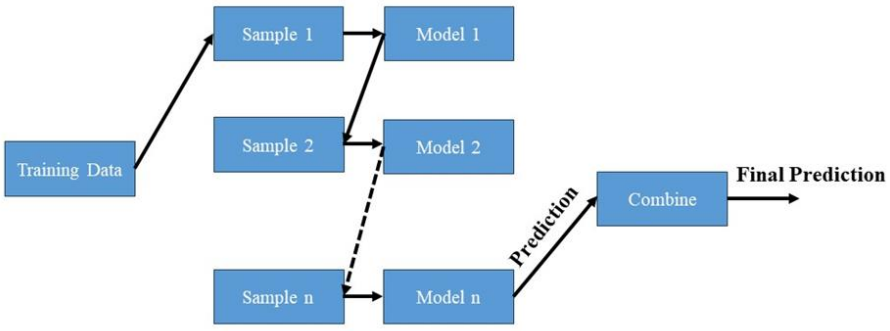


Figure 2: Ensemble Learning Boosting model how it works

Weaker models n are given lower weights compared to strong models when their predictions are merged to get the final result. We start by setting the data weights to a uniform value and then proceed with the following steps in a repetitive manner:

- Train a model using all occurrences.
- Determine the overall inaccuracy in the model's output across all instances.
- Assign a weight to the model based on its performance (high for good performance and low for poor performance).
- Revise data weights. Assign more weights to samples with significant mistakes.

- If the performance is unsatisfactory or other halting circumstances are encountered, repeat the previous procedures.

Ultimately, we merge the models to create the one we will utilize for making predictions.

Trained the Model

System training involves dividing the data into two parts, using 80% of the data is used for training and 20% is used for testing. During the training process, we provide both the input and output for 80% of the data in the training sets. The model exclusively utilizes training data for the purpose of learning. In order to construct our model, we utilized a diverse range of machine learning techniques (namely, the chosen model for boosting ensemble learning). Once our suggested boosting ensemble learning is implemented, it will be based on our test set datasets. During testing, the model is provided with the remaining 20% of the data that it has not before encountered. The boosting ensemble learning algorithm generated a predicted value, which we then compared to the actual output in order to assess its accuracy.

Supervised Learning Method

In the field of classification, a technique is employed to acquire the ability to predict the category-based output variable or class label, such as identifying whether a case is negative or positive for monkeypox [38].

Evaluations of Classification Models

The evaluation metrics employed for each model encompass Accuracy, Precision, and Recall. The calculation of these three metrics involves the utilization of variables from the confusion matrix, which include true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Accuracy is determined by dividing the sum of true positives and true negatives by the entire amount of data. An essential step in machine learning is assessing the performance and generalizability of a monkeypox disease prediction model by the evaluation of its classification capabilities on new and untested data. For the monkeypox infected human illness prediction problem and goals, a classification model can be assessed using several metrics and approaches. We utilized a roster of commonly employed assessment criteria for employees.

Classification Accuracy: refers to the proportion of instances correctly categorized as monkeypox patients out of all the examples in the test set. The metric is clear and comprehensible, but in datasets with imbalanced class distribution, Due to the prevailing dominance of the majority class, the accuracy score can be misleading.

Confusion matrix: A table employed to generate various assessment metrics, the output displays the number of true positives, true negatives, false positives, and false negatives for each class. Precision can be defined as the ratio of all expected positives to true positives. Whereas recall is the ratio of true positives to all real positives. These metrics are useful in situations where there is a trade-off between the occurrence of false positives and false negatives, or when one class holds greater importance than the other. The F1-Score is computed using the formula 2 x (precision x recall) / (precision + recall), and it represents the harmonic mean of precision and recall. In datasets with imbalanced class distribution, when both accuracy and recall are important, this statistic proves to be valuable.

ROC curve and AUC: The Receiver Operating Characteristic (ROC) curve is a graphical representation of the relationship between the true positive rate (recall) and the false positive rate (1-specificity) for different threshold values of the decision function of the classifier. The overall efficacy of the classifier is evaluated using the Area under the Curve (AUC), which can have values between 0.5 (representing random guessing) and 1 (indicating flawless categorization).

Cross-validation: is a technique used to provide a more precise evaluation of the model's performance. It involves dividing the

data into many folds, training the model on each fold, and subsequently testing it on the remaining folds. In order to address the issue of overfitting, it is crucial to select the most suitable evaluation metric(s) depending on the particular problem and needs, and to evaluate the model's performance using separate test data.

Results

The algorithms' performance has been assessed based on their accuracy percentage rate. The performance measurements of various machine learning algorithms on the monkey pox dataset are being proposed. This section presents the results obtained from the experimental study conducted on the proposed system. This study utilized experimental analysis by applying diverse machine learning algorithms to the monkey pox dataset for the purpose of label-based categorization. The initial step involves assessing the precision of the algorithms. Subsequently, a confusion matrix has been constructed for the high-precision classifier, utilizing the AdaBoost, GBM, XGBM, LGBM and CatBoost models achieved almost all classifiers score is more than above 93% and highest accuracy score is 97.67%. As shown in Table 2 the percentage accuracy achieved by different algorithms.

Table2*:* Boosting classification accuracy comparison performance

ML Classifiers	Accuracy
AdaBoost	97.67%
GBM	93.02%
XGBM	95.34%
LGBM	93.02%
Cat Boost	93.02%

Based on the data presented in the table above, It is clear that the AdaBoost, GBM, XGBM, LGBM and CatBoost model had the best accuracy percentage among the other classifiers. The accuracy rate obtained is illustrated by the curve displayed in Figure 3.

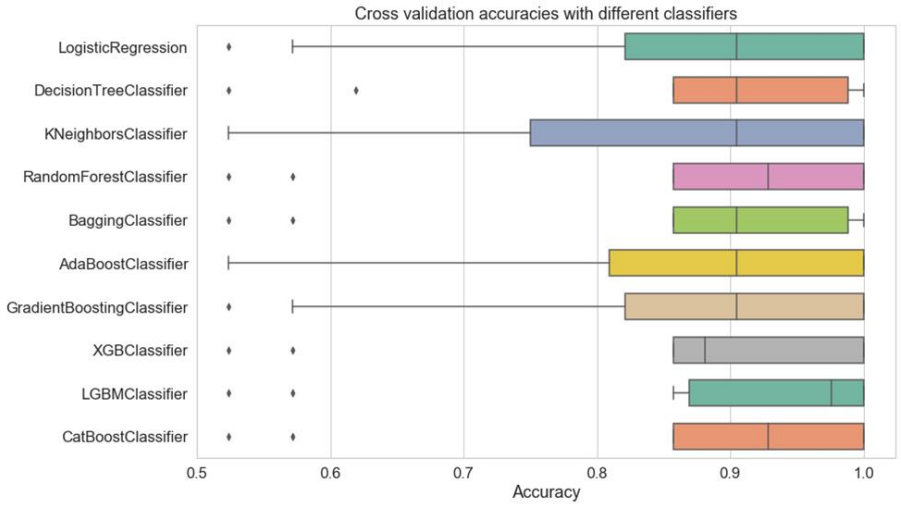
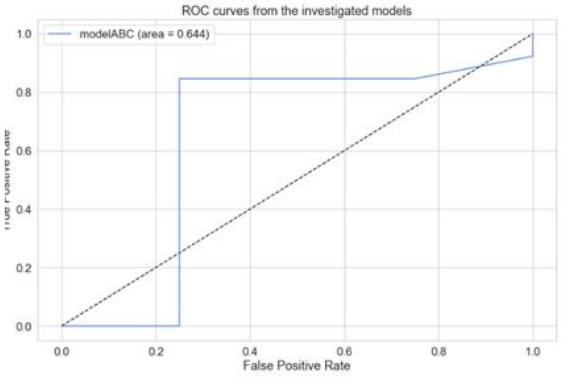


Figure 3: Accuracy performance basis on comparison of different classifiers

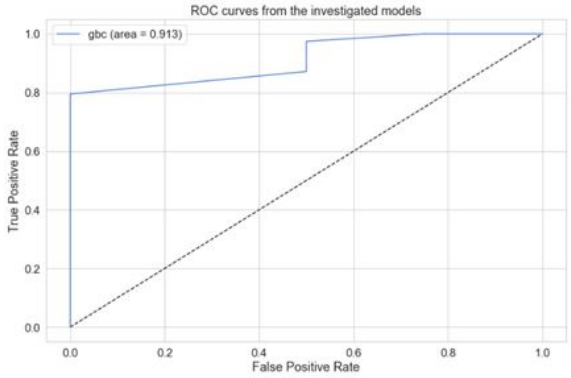
The proposed boosting classifier achieves the highest accuracy on the dataset in question due to its dependency on proximity, which enhances the quality of predictions. The AdaBoost, GBM, XGBM, LGBM and CatBoost models demonstrate high efficiency and yield competitive results on a multiclass dataset. Consequently, based on the proposed model, the AdaBoost and XGBoost classifier achieves the highest level of accuracy.

Performance Based on ROC(AUC)

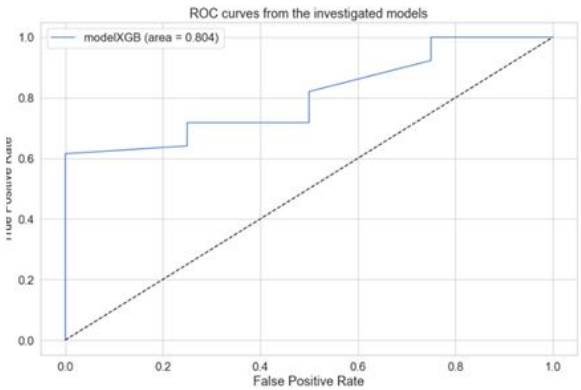
The renowned statistical approaches, including accuracy, precision, recall, and F1-score, are employed to measure and demonstrate the overall experimental results. The following are the definitions attributed to them: We employed 5-Fold cross validation on the preexisting building design models equipped with the mechanisms to differentiate monkeypox from other related diseases, such as monkeypox or non-monkeypox. The AdaBoost, GBM, XGBM, LGBM and CatBoost models demonstrate superior performance in a two-class method, as seen by its exemplary performance in a 5-fold-wise evaluation, as depicted in Figure 4. Displays the performance of the model on the training and testing sets for each epoch in every study. Figure 4(b) shows that the model's performance in Study Two reached its maximum at 100 epochs for both the training and validation dataset. Figure 4 displays the AUC-ROC curves for study one and study two. In study one, TPR represents the true positive rate and FPR represents the false positive rate. Similarly, in study two, TPR represents the true positive rate and FPR represents the false positive rate. The Figure 4 (a) denoted the achieved score of AdaBoost algorithm, Figure 4 (b) GBC ROC(AUC) Score C., XGB ROC(AUC) Score ,D. LGBM ROC(AUC) Score, and E. CBC ROC(AUC) Score. The AUC-ROC plot displays the performance of the proposed model using ensemble feature, sequence-based feature, and graphlet feature, as measured by the AUC-ROC score.



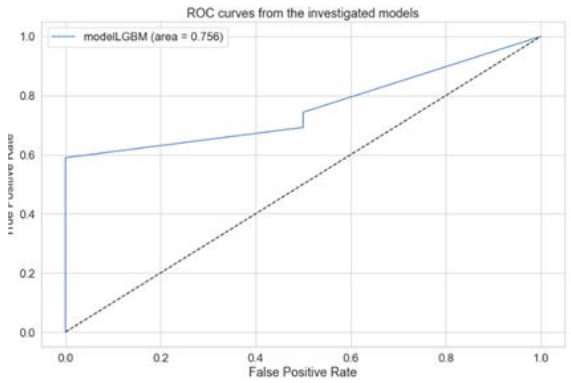
ABC ROC(AUC) Score



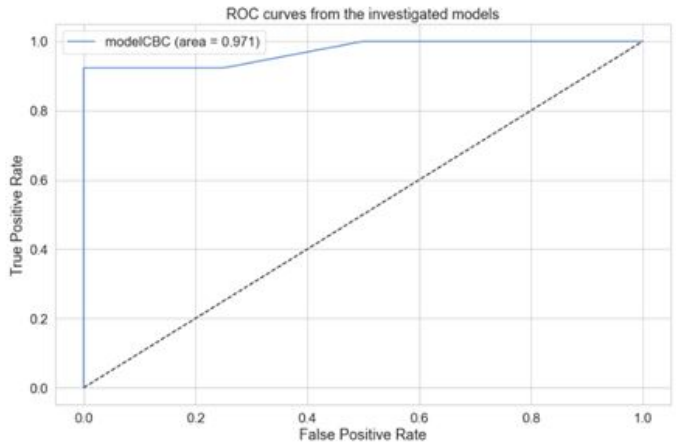
B. GBC ROC(AUC) Score



C. XGB ROC(AUC) Score



D. LGBM ROC(AUC) Score



E. CBC ROC(AUC) Score

Figure 4: Performance based on ROC(AUC)

Comparison Analysis Results Based on ROC(AUC)

In the section 3.2 comparison analysis results of result of AdaBoost, GBC, XGB, LGBM and CatBoost. Best models validation based on 5-fold-wise performance in a two-class approach. The optimal splitting method was determined to be 5-fold cross validation, while the optimal optimization approach was found to be CatBoost a. This conclusion was based on performance measures including accuracy, specificity, sensitivity, precision, G-Mean, F1-score, and AUC (Area Under Curve). AdaBoost AUC= 0.644, GBC AUC= 0.910, XGB AUC= 0.804, LGBM AUC= 0.756, and CatBoost AUC= 0.971 as shown in figure 5. Decision tree AUC= 0.734, KNN AUC= 0.628, and RF AUC= 0.862 obtained score as shown in figure 6. The findings demonstrated that the ensemble feature outperformed the other characteristics, attaining the greatest Area Under the Receiver Operating Characteristic (AUC-ROC) score of 0.971. On the one hand, the sequence-based feature showed a decent performance, with an AUC-ROC score of 0.974. The optimal splitting method was determined to be 10-fold cross validation, whereas the optimal optimization strategy was found to be CatBoost classifiers, based on the performance of AUC (Area Under Curve).

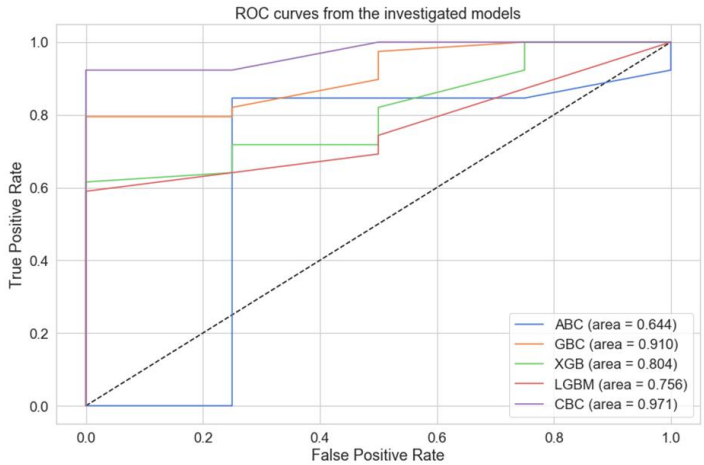


Figure 5: Best models based on 5-fold-wise performance in a two-class approach.

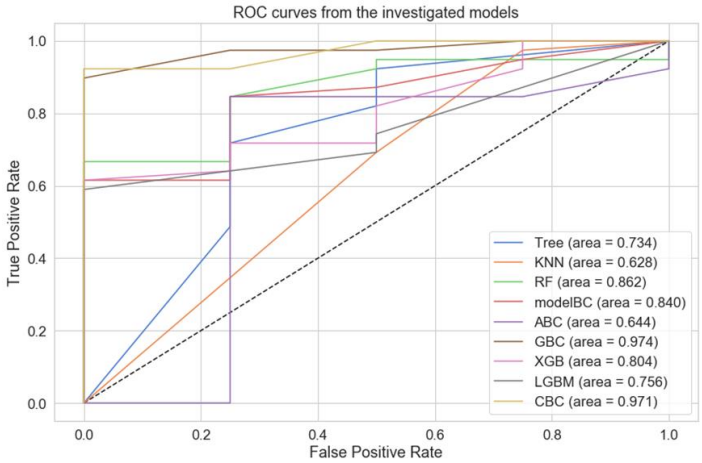


Figure 6: Best models based on 5-fold-wise performance with other ML classifiers.

Comparison Analysis of Machine Learning Models Based on Evaluation Parameters

Furthermore, Table 3 presents a comparison of the performance measures of the mentioned architectures. These metrics include Precision, Recall, F-Score, and test accuracy, which are evaluated on the selected datasets. The issue of classifying monkey pox is more or less effectively addressed by the monkey pox disease compared to the other two models. The reason for this is the utilization of auxiliary classifiers in addition to the trunk classifier. Moreover, due to its architecture having a reduced number of parameters to be learned compared to the boosting architecture, the computational cost is also enhanced and superior to other models. Table 3 reveals that AdaBoost, GBM, XGBM, LGBM and CatBoost models can be combined with other machine learning models or incorporated into suggested models.

Table 3: Performance Comparison evaluation parameters

Boosting ML classifiers	accuracy	precision	recall	f1-score
AdaBoost Classifier	97.67	91.62	98.71	93.79
GBM Classifier	93.02	80.83	73.71	76.67
XGBM Classifier	95.34	97.56	75.88	82.08
LGBM Classifier	93.02	78.57	96.15	84.36
Cat Boost Classifier	93.02	96.42	62.51	68.14

Accuracy Comparison Based on Boosting Ensemble Learning Method

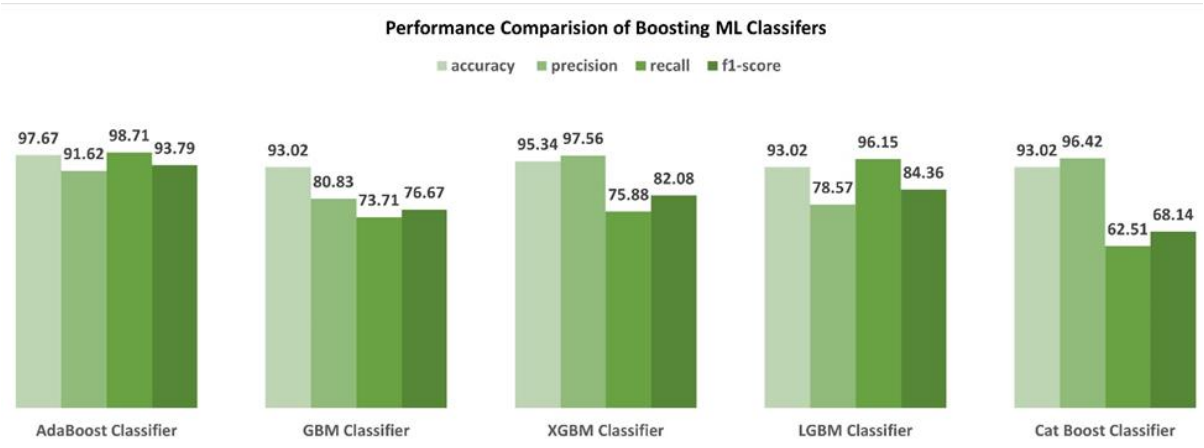


Figure 7: Performance comparison of boosting ML Classifiers

The AdaBoost algorithm obtained best accuracy score as shown in figure 7 first blue bar which denoted AdaBoost classifiers. Figure 7 reveals a significant disparity, as the number of variables in monkeypox positive individuals surpasses that in monkeypox negative patients. AdaBoost classifier accuracy score 97.67%, precision received score 91.62%, and recall received score 98.71%, GBM classifier received accuracy score 93.02%, XGBM classifier received accuracy score 95.34%, LGBM classifier received accuracy score 93.02% and Cat Boost classifier received accuracy score 93.02% performance based on boosting method.

Accuracy Comparison Based on Boosting Ensemble Learning Method

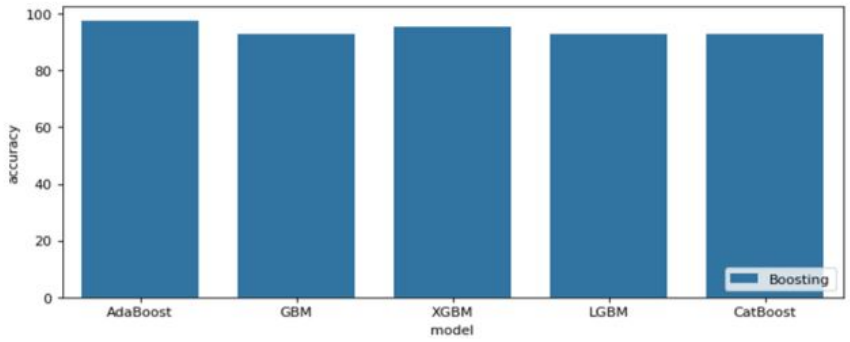


Figure 8: Accuracy comparison based on Boosting ensemble learning method

Figure 8 displays the classification outcomes of the Gradient Boosting algorithm, with a precision rate of 97.67%, for positive findings and 64% for negative results. AdaBoost classifier accuracy score 97.67%, GBM classifier received accuracy score 93.02%, XGBM classifier received accuracy score 95.34%, LGBM classifier received accuracy score 93.02% and Cat Boost classifier received accuracy score 93.02% performance based on boosting method. The precision of the AdaBoost algorithm classification is 97.67% for positive results and 48% for negative outcomes, as shown in Figure 8. This finding suggests that ensemble boosting method learning techniques is a superior predictor compared to the other models employed in this study.

Comparative Confusion Results Based on Boosting Ensemble Learning Method

The metrics for XGBoost model attain a perfect score of 97.67% in both the train and test set split, as evidenced in Table 4, hence surpassing the metrics of any other approach. Consequently, the XGBoost model outperforms the other models. In order to provide a clearer representation of the models' performance, we have included the confusion matrix for each technique in Figure 9. AdaBoost model demonstrated superior performance compared to the other four machine learning algorithms. The AdaBoost model attained a perfect score of 95.34% score. XGBoost demonstrated superior performance 97.67% score compared to the other four machine learning algorithms. This is a result of multiple variables. XGBoost is a technique for ensemble learning that integrates many decision trees, enabling it to capture intricate patterns in the data.

Table 4: Comparison machine learning models based on 5-Fold CV evaluation parameters

Boosting ML classifiers	accuracy	precision	recall	f1-score
AdaBoost model	95.34	75.62	97.56	82.08
GBM model	93.02	72.83	96.34	76.67
XGBM model	97.67	97.72	97.67	97.30
LGBM model	83,33	69.44	83.33	75.75
Cat Boost model	85.71	87.80	85.71	80.92

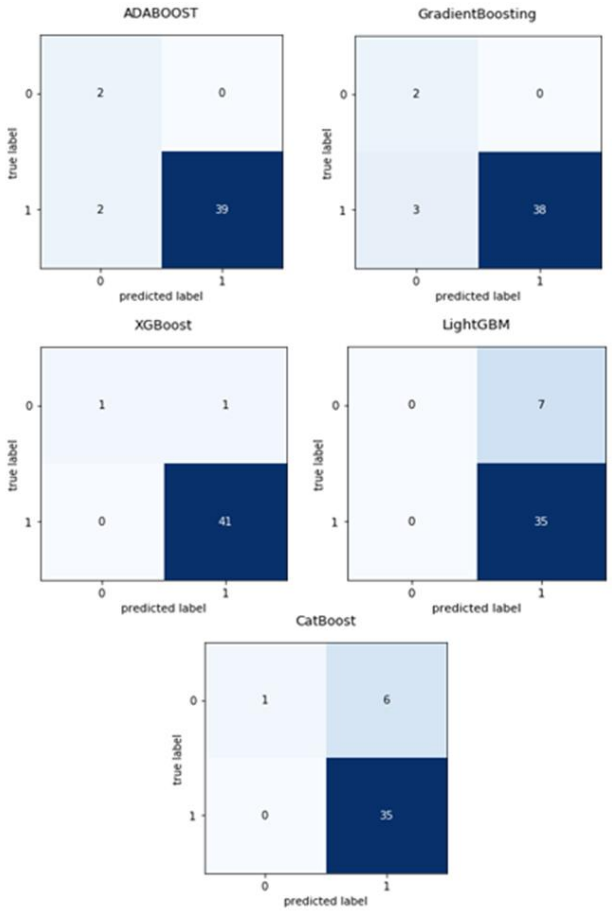


Figure 9: Confusion matrix for each ML method

Future Work

In future work, additional techniques and strategies will be employed to enhance the performance of the model, such as employing advanced methods like word embedding (e.g., doc2Vec) and text labeling (e.g., Azure Machine Learning). Moreover, our intention is to utilize deep learning and transformer algorithms to enhance the accuracy of sentiment analysis and emotion prediction. We obtained the dataset and conducted training on it using classifiers. We used text normalization before we trained and tested our models. Afterwards, we trained the model using 80% of the dataset for training and 20% for testing.

Discussion

This study presents and assesses six unique deep learning models that have been enhanced to distinguish between patients displaying signs of monkeypox and those who do not. This study employed a range of physiological variables and applied machine learning methods, including AdaBoost model demonstrated superior performance in terms of both time consumption and accuracy, with a score of 97.67%. GBM model obtain accuracy score 93.02%. The classification accuracy of the Extreme Gradient Boosting Machine (XGBM) was found to be 95.34%, Light Gradient Boosting Model (LGBM) (achieved accuracy of 93.02%), Cat Boost Model achieved accuracy of 93.02%. The AdaBoost method demonstrated the highest level of accuracy for this specific task, resulting in an approximate accuracy rate of 97.67%. Among them, the AdaBoost model demonstrated superior performance in terms of both time consumption and accuracy, with a score of 97.67%. GBM model obtain accuracy score 93.02%. The classification accuracy of the Extreme Gradient Boosting Machine (XGBM) was found to be 95.34%, Light Gradient Boosting Model (LGBM) (achieved accuracy of 93.02%), Cat Boost Model achieved accuracy of 93.02%. The AdaBoost method demonstrated the highest level of accuracy for this specific task, resulting in an approximate accuracy rate of 97.67%.

Conclusions

Originally, there was a belief that machine learning (ML) approaches would act as the foundation for automating data mining and classification. In the area of medical science, numerous AI models have been developed utilizing image analysis to detect various viruses, indicating the significant growth in the adoption of AI models across diverse domains. The accuracy values acquired from the experimental findings of ensemble learning methods, specifically Boosting, varied across the datasets. The current monkeypox outbreak is a matter of concern worldwide. It would be prudent to prepare for improving outcomes notwithstanding the relatively less severe nature of the 2019 coronavirus illness. The utilization of artificial intelligence in health science applications and research has experienced a significant surge in recent years. In this study, we performed a comprehensive assessment of the latest artificial intelligence strategies employed to combat the monkeypox virus.

We employed ensemble learning methods, specifically Boosting, a subdivision of artificial intelligence, to predict the outcome of an event using modern techniques. In this paper, the rapid miner tool was utilized to construct a ensemble learning methods, specifically Boosting model that accurately forecasted the fate of the monkey pox outbreak based on the recorded cases. The experiment included a diverse range of supervised learning classifiers, and the results were showcased through comparative analysis. In the future, we can extend the application of this model to additional datasets. By increasing the amount of data and incorporating other performance measures, we may enhance the model's accuracy rates and expand its capabilities. Based on the outcomes of the monkey pox datasets, it is evident that the Boosting approach consistently achieves the most favorable results. However, Boosting still outperforms in terms of Precision and Recall measurements. Hence, the Boosting technique has the potential to surpass the performance of approaches. Light Gradient Boosting is a boosting algorithm that achieves the highest accuracy results on monkeypox datasets. However, the inclusion of Adaptative Boost in the boosting approach consistently yields the lowest accuracy results compared to other models. We introduced a new ensemble learning boosting model in this study that significantly enhances the prediction of monkey pox disease identification. Additionally, we conducted separate validation on four cross-species monkey pox csv format datasets. Our method has been shown to be successful and resilient in predicting monkey pox disease identification based on experimental data from cross validations and comparisons.

Institutional Review Board Statement

Not applicable

Informed Consent Statement

Not applicable

Data Availability Statement

The data presented in this study are available in this article

Acknowledgments

The authors would like to express their sincere gratitude to Guilin University of Electronic Technology for its invaluable support throughout this research

Conflicts of Interest

The authors declare no conflicts of interest

References

1. Petersen E, Ntoumi F, Hui DS, Abubakar A, Kramer LD et al. (2022) Emergence of new SARS-CoV-2 Variant of Concern Omicron (B.1.1.529)-highlights Africa's research capabilities, but exposes major knowledge gaps, inequities of vaccine distribution, inadequacies in global COVID-19 response and control efforts. Int. J. Infect. Dis, 114: 268–72.

2. Habib MA, Rahman MT (2020) Enhancing Monkeypox Detection: A Fusion of Machine Learning and Transfer Learning.

3. Zumla A, Valdoleiros SR, Haider N, Asogun D, Ntoumi F, Petersen E, Kock R (2022)Monkeypox Outbreaks Outside Endemic Regions: Scientific and Social Priorities. Lancet Infect. Dis, 22: 929–31.

4. Orviz E, Negredo A, Ayerdi O, Vázquez A, Muñoz-Gomez A, Monzón S, Clavo P, Zaballos A, Vera M., Sánchez P et al. (2022) Monkeypox Outbreak in Madrid (Spain): Clinical and Virological Aspects. J. Infect, 85: 412–7.

5. Tarín-Vicente EJ, Alemany A, Agud-Dios M, Ubals M, Suñer C et al. (2022) Clinical Presentation and Virological Assessment of Confirmed Human Monkeypox Virus Cases in Spain: A Prospective Observational Cohort Study. Lancet, 400: 661–669.

6. Wei F, Peng Z, Jin Z, Wang J, Xu X et al. (2022) Study and Prediction of the 2022 Global Monkeypox Epidemic. J. Biosaf. Biosecur, 4: 158–62.

7. Huang Y, Mu L, Wang W (2022) Monkeypox: Epidemiology, Pathogenesis, Treatment and Prevention. Signal Transduct. Target. Ther, 7: 373.

8. Harapan H, Ophinni Y, Megawati D, Frediansyah A, Mamada SS et al. (2022) Monkeypox: A Comprehensive Review. Viruses, 14: 2155.

9. Fink DL, Callaby H, Luintel A, Beynon W, Bond H et al. (2022) Clinical Features and Management of Individuals Admitted to Hospital with Monkeypox and Associated Complications across the UK: A Retrospective Cohort Study. Lancet. Infect. Dis, 3099: 6-14.

10. E Tuba et al. (2019) Classification and feature selection method for medical datasets by brain storm optimization algorithm and support vector machine Procedia Comput Sci, 6

11. H Li et al. (2022) The evolving epidemiology of monkeypox virus Cytokine Growth Factor Rev, 7.

12. S Nolasco et al. (2023) First case of monkeypox virus, SARS-CoV-2 and HIV co-infection J Infect, 8.

13. Reynolds MG, Emerson GL, Pukuta E, Karhemere S, Muyembe JJ et al. (2013) Detection of human monkeypox in the republic of the congo following intensive community education. The American Journal of Tropical Medicine and Hygiene, 88: 982-4.

14. Ahsan MM, Uddin MR, Farjana M, Sakib AN, Momin KA et al. (2022) Image data collection and implementation of deep learning-based model in detecting monkeypox disease using modified vgg16, arXiv, 2

15. Khafaga DS, Ibrahim A, El-Kenawy ESM, Abdelhamid AA, Karim FK et al. (2022) An Al-Biruni Earth Radius Optimization-Based Deep Convolutional Neural Network for Classifying Monkeypox Disease. Diagnostics, 12: 2892.

16. Bala D, Hossain MS, Hossain MA, Abdullah MI, Rahman MM et al. (2023) MonkeyNet: A Robust Deep Convolutional Neural Network for Monkeypox Disease Detection and Classification. Neural Netw, 161: 757–75.

17. Ahsan M, Ramiz M, Ali S, Islam K, Farjana M et al. (2023) Deep Transfer Learning Approaches for Monkeypox Disease Diagnosis. Expert Syst. Appl, 216: 119483.

18. Eid MM, El-Kenawy ESM, Khodadadi N, Mirjalili S, Khodadadi E et al. (2022) Meta-Heuristic Optimization of LSTM-Based Deep Network for Boosting the Prediction of Monkeypox Cases. Mathematics, 10: 3845.

19. Iftikhar H, Khan M, Khan MS, Khan M (2023) Short-Term Forecasting of Monkeypox Cases Using a Novel Filtering and Combining Technique. Diagnostics, 13: 1923.

20. Bhosale YH, Zanwar SR, Jadhav AT, Ahmed Z, Gaikwad VS et al. (2022) Virus: Machine Learning Prediction Model, Outbreak Forecasting, Visualization with Time-Series Exploratory Data Analysis. In Proceedings of the 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Virtual, 2022: 1–6.

21. Mandal AK, Sarma PKD, Dehuri S (2023) Machine Learning Approaches and Particle Swarm Optimization Based Clustering for the Human Monkeypox Viruses: A Study. In Proceedings of the Innovations in Intelligent Computing and Communication: First International Conference, ICIICC 2022, Bhubaneswar, India, 16–17 December 2022; Springer: Berlin/Heidelberg, Germany, 2023: 313–332.

22. Kennedy J, Eberhart R (1995) Particle Swarm Optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 4: 1942–8.

23. Maranhão A (2022) Global Monkeypox Cases (Daily Updated).

24. Multi-Country Monkeypox Outbreak in Non-Endemic Countries (2022)

25. C Farabet, C Couprie, L Najman, Y LeCun (2013) Learning Hierarchical Features for Scene Labeling," in IEEE Transactions on Pattern Analysis and Machine Intelligence, 35: 1915-29.

26. https://www.kaggle.com/code/prashanththangavel/advanced-feature-engineering-feature-encoding

27. https://www.kaggle.com/code/teowaihong/monkeypox-classification-using-sklearn

28. Y Sun, Z Li, X Li, J Zhang (2021) Classifier selection and ensemble model for multi-class imbalance learning in education grants prediction, Appl. Artif. Intell, 35: 290–03.

29. P Wu, H Zhao (2011) Some analysis and research of the AdaBoost algorithm, in Intelligent Computing and Information Science. Berlin, Germany: Springer, 2011: 1–5.

30. JH Friedman (2001) Greedy function approximation: A gradient boosting machine, Ann. Statist, 5: 1189–232.

31. A Natekin, A Knoll (2022) Gradient boosting machines, a tutorial, Frontiers Neurorobot, 7: 21.

32. JH Friedman (2002) Stochastic gradient boosting, Comput. Statist. Data Anal, 38: 367–78.

33. J Jiang, R Wang, M Wang, K Gao, DD Nguyen et al. (2020) Boosting tree-assisted multitask deep learning for small scientific datasets, J. Chem. Inf. Model, 60: 1235–44.

34. Y Li, W Chen (2020) A comparative performance assessment of ensemble learning for credit scoring," Mathematics, 8: 1756.

35. W Liang, S Luo, G Zhao, H Wu (2020) Predicting hard rock pillar stability using GBDT, XGBoost, and LightGBM algorithms," Mathematics, 8: 765.

36. G Ke (2017) LightGBM: A highly efficient gradient boosting decision tree," in Proc. Adv. Neural Inf. Process. Syst, 30: 3149–57.

37. L Prokhorenkova, G Gusev, A Vorobev, A V Dorogush, A Gulin (2018) CatBoost: Unbiased boosting with categorical features," in Proc. Adv.Neural Inf. Process. Syst, 31: 6639–49.

38. https://www.kaggle.com/code/vuppalaadithyasairam/svm-with-rbf-kernel-for-monkeypox-prediction

39. Daanouni O, Cherradi B, Tmiri A (2019) Predicting diabetes diseases using mixed data and supervised machine learning algorithms. In Proceedings of the 4th International Conference on Smart City Applications, 1-6.

40. Nai-Arun N, Sittidech P (2014) Ensemble learning model for diabetes classification. In Advanced Materials Research, 931: 1427-31.

41. Mung PS, Phyu S (2020) Ensemble Learning Method for Enhancing Healthcare Classification. Proceedings of the 10th International Workshop on Computer Science and Engineering (WCSE 2020).

42. https://www.mygreatlearning.com/blog/label-encoding-in-python/