Development of a Hybrid Monkeypox Detection Model Using CNN and Extreme Gradient Boosting

A. E. Ogoigbe*, O.K. Boyinbode[2], A.H Afolayan[3], M. T. Kinga[4]

[1]Computer Science Department, School of computing, Federal University of Technology, Akure

[2,4]Information Technology Department, School of computing, Federal University of Technology, Akure

[3]Information System Department, School of computing, Federal University of Technology, Akure

Department of information technology, school of computing, Federal University of Technology, Akure,

Department of computer science, school of computing, Federal University of Technology, Akure,

DOI: https://doi.org/10.51244/IJRSI.2025.12040026

Received: 17 March 2025; Accepted: 21 March 2025; Published: 01 May 2025

ABSTRACT

The Monkeypox virus poses as a public health risk that might quickly escalate into a worldwide epidemic. Machine learning (ML) has recently shown much promise in diagnosing diseases like cancer, finding tumor cells, and finding COVID-19 patients. The timely identification and accurate categorization of Monkeypox cutaneous manifestations are crucial for the successful implementation of containment strategies. It requires sophisticated methodologies to detect and combat this evolving orthopoxvirus at an early stage. This study presents an exploration of a hybrid machine learning model integrating CNN (Convolutional Neural Network), XG Boost (Extreme Gradient Boosting), and XG boost based stack model for classification and detection with other standard machine learning methods such as Support Vector Machine (SVM) and Random Forest, where transfer learning and the DL algorithms for the skin lesion data will enhance and train model, while SHAP methods were adopted to examine and analyze XGBoost predictions. The resulting ensemble model is not only adept at detecting Monkeypox virus through its lesion and symptoms but also showcases computational efficiency with a predictive accuracy, recall, precision, and F1 Score, all reaching a value of 1.0. In a comparison analysis conducted on other deep learning models, the suggested model has superior performance as a hybrid model compared to other models. The exceptional performance demonstrated in this study underscores the effectiveness of the methodology in accurately classifying skin lesions and symptoms linked to Monkeypox. This approach holds promise for individuals, as it enables early detection, a vital factor in preventing the spread of Monkeypox.

Keywords: Monkeypox, Convolutional Neural Network, Extreme Gradient Boosting, Shapley Additive, Machine learning

INTRODUCTION

In recent years, pandemics such as COVID-19, Ebola, and HIV/AIDS have posed significant global threats. As the impact of these pandemics has lessened, other infectious diseases, such as Monkeypox, have emerged. The resurgence of Monkeypox in an undervaccinated population is a global health issue requiring immediate attention from the scientific community [1]. Monkeypox, a rare viral disease caused by an orthopoxvirus, produces symptoms similar to those of smallpox. Monkeypox, a zoonotic virus emerging into the spotlight of global health concerns, traces its origins and molecular lineage to the Orthopoxvirus genus, sharing taxonomic kinship with variola (smallpox), vaccinia, and cowpox viruses [2]. The World Health Organization (WHO) has declared Monkeypox a Public Health Emergency of International Concern (PHEIC) under the International Health Regulations, following its spread from the Democratic Republic of the Congo (DRC) to other parts of Africa, including Nigeria. Recently, many cases have been reported outside Africa, and the outbreak of Monkeypox has been rapidly spreading worldwide [3]. The orthopoxvirus genus employs various strategies to evade the host's immune defenses, allowing the virus to enter the host undetected or unrecognized. There are two distinct strains of Monkeypox Virus (MPXV) unique to Africa: clade I, which is prevalent in Central Africa, and clade II, which is found in Western Africa. Monkeypox transmission occurs through handling bush meat, animal contact, bodily fluids, or contaminated objects, with rodents serving as primary carriers [4]. Symptoms of Monkeypox include swollen lymph nodes, fever, fatigue, chills, and a rash that could be mistaken for chickenpox or a sexually transmitted infection, especially if it appears in the genital or anal areas. Although Monkeypox cases globally in 2023 can lead to severe illness and complications if not properly managed, the Economic Community of West African States (ECOWAS) reported 44 confirmed cases and 1 death since the beginning of the year, as of Week 33 of the 2024 Epidemiological Report. These cases were distributed across Nigeria (24 cases), Côte d'Ivoire (11 cases), Liberia (5 cases), and Ghana (4 cases) [5]. Additionally, on Thursday, August 15, 2024, global health officials confirmed the first case of a new strain of Monkeypox outside Africa, in Sweden [6]. In the absence of an effective vaccine for Monkeypox in Nigeria, early and accurate detection of the infection is crucial for both private and public healthcare intervention. The infection typically resolves on its own. Image processing techniques can be employed to predict or diagnose whether an individual is a carrier of Monkeypox based on the lesions on their body. Image processing is a technology that can identify and verify the type of lesions, distinguishing them from those caused by other rash-related infectious diseases like smallpox, chickenpox, and cowpox, offering high accuracy and reliability. The image detection system is a computer model that automatically identifies and verifies the presence of skin lesions using digital images. Similar to facial recognition systems, which compare input images against a database to find matches, the Monkeypox Image Prediction System (MIPS) evaluates an individual's skin to predict if they are a carrier of Monkeypox and preprocesses the image. In addition to the image recognition system, combining it with a symptom detection system would offer a more effective and comprehensive solution.

Ensemble learning is a powerful machine learning technique that combines the output of multiple models, or "weak learners," to solve complex problems in classification and prediction. The goal is to enhance model performance. An ensemble model merges predictions from two or more individual models [7]. Stacking is an ensemble learning method that integrates multiple machine learning algorithms through a meta-learning process. The base-level algorithms are trained on a comprehensive dataset, and a meta-model is then trained using the combined outputs from these base models as input features. Bagging, boosting, and stacking are the three main ensemble learning techniques used in machine learning. Bagging reduces variance by averaging predictions from various trained models, while boosting aims to reduce bias by creating sequential models to minimize variance. Stacking involves training a higher-level meta-model to combine predictions from different base models, leveraging the strengths of various modeling approaches. Stacking algorithms can have multiple layers, making the training process computationally intensive.

The study at hand leveraged stacked ensemble learning (SEL) models to detect Monkeypox using lesion images and presented symptoms. Stacking amalgamates multiple parallel "weak learner" models, then utilizes meta-learners to optimize how their respective predictions are synthesized to generate an ameliorated overall prediction model. Stacking ingests the individual predictions from base-level learners as inputs to a meta-level learner model, yielding an enhanced final prediction output.

Here is how the break of the paper is structured: In Section 2, we describe the relevant literature, and in Section 3, we offer explanation of the proposed model. In Section 4, the trial analysis and validation are presented, and in Section 5, the conclusion is illustrated.

Related Works

[8] presented a research work titled “A Neuro-Fuzzy based model for diagnosis of Monkeypox diseases”. It designs a system that uses neuro-fuzzy logic and neural networks for Monkeypox disease diagnosis. The model was able to differentiate Monkeypox from other pox families using 18 symptoms that are linked to it. The system made use of 3 out of 18 Monkeypox symptoms as inputs

[9] presented a research work titled “A Deep Learning System for Differential Diagnosis of Skin Diseases". The model used deep learning network/framework(tensor flow). DLS distinguishes between 26 of the most common skin disorders which account for around 80% of all skin issues seen in healthcare system. The Proposed system

was not evaluated against existing deep learning systems and monkeypox is not one of the skin diseases that the proposed system is designed to identify.

[10] designed a Monkeypox classification system using a mobile app. This made use of Android Studio with Java, Android SDK 12, TensorFlow Lite. The system Introduced a modified MobileNetV2 model for Monkeypox detection with 91.11% accuracy. Developed a basic, cheap, and non-invasive mobile app for personal detection and isolation. The system is limited to Android platform and cannot classify other skin diseases.

[11] developed a system to diagnose Monkeypox through selected hybrid CNNs unified with feature selection and ensemble learning. The model made use of Monkey-CAD coupled with using a hybrid CNN unified with feature selection and ensemble learning (Monkey-CAD extracts features from 8 CNN and it made use also of discrete wavelet transform (DWT) to merge features. Monkey-CAD could discriminate among cases with and without Monkeypox, achieving an accuracy of 97.1% and 98.7% for both Monkeypox skin image (MSI) and skin lesion (MSL) datasets. The study only focused on the diagnosis of Monkeypox through its lesions.

[12] designed a system to detect Monkeypox using CNN and transfer learning. MobileNet V3-S, EfficientNetV2, ResNet50, VGG-19, DenseNet121, Xception. MobileNet V3-S achieved the best performance with an AUC of 0.99 and F1 score of 0.98. The research did not consider clinical factors (symptoms) but only images.

[13] developed a system to detect Monkeypox case based on symptoms using XGBoost and Shapley addictive explanation methods. This system used XGBoost to detect Monkeypox through its symptoms and Shapley addictive to interpret the output of the XGBoost model. Using XGBoost model, it outperformed other methods, reaching an accuracy 1.0 in general test and 0.9 in 5 fold cross validation. The proposed system had non-incorporation of epidemiological data into the model and it focused only on diagnosis on the symptoms.

[14] developed a system to detect Monkeypox using hyper-parameter tuned based transferable CNN model .This made use of hyper-parameter tuned based transfer CNN model coupled with the optimization algorithm (MGS-ROA) which makes hyper-parameter adjustment easier. The system had an accuracy sensitivity and specificity of 93.60% for all. It can only train and validate Monkeypox using its lesions.

METHODOLOGY

Data Sets

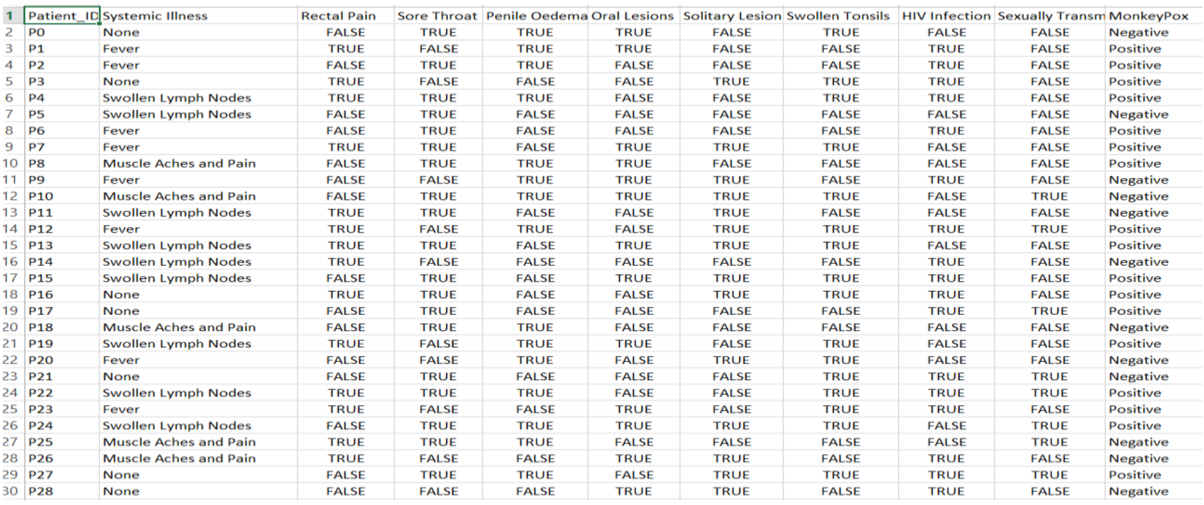
Symptoms

The proposed dataset for the symptoms is published on Kaggle by “Muhammad”, titled “Monkeypox PATIENTS Dataset”. This is a SYNTHETIC dataset generated based on a study published by the bmj.com: Clinical features and novel presentations of human Monkeypox in a central London center during the 2022 outbreak: descriptive case series.

Dataset consists of a CSV which have a record of 25,000 Patients with their corresponding features and a target variable indicating if the patient has Monkeypox or not. Features: Patient_ID, Systemic Illness, Rectal Pain, Sore Throat, Penile Oedema, Oral Lesions, Solitary Lesion, Swollen Tonsils, HIV Infection, and Sexually Transmitted Infection

Target Variable: Monkeypox. The dataset currently contains boolean and categorical features and in future, we might add more data and features to help you identify the patients of Monkey-Pox. The lesion images sample data is illustrated in Table 1.

Table 1: Symptom Data



1	Patient_ID	Systemic Illness	Rectal Pain	Sore Throat	Penile Oedema	Oral Lesions	Solitary Lesion	Swollen Tonsils	HIV Infection	Sexually Transm	MonkeyPox
2	P0	None	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	Negative
3	P1	Fever	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	Positive
4	P2	Fever	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	Positive
5	P3	None	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	Positive
6	P4	Swollen Lymph Nodes	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	Positive
7	P5	Swollen Lymph Nodes	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	Negative
8	P6	Fever	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	Positive
9	P7	Fever	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	Positive
10	P8	Muscle Aches and Pain	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	Positive
11	P9	Fever	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	Negative
12	P10	Muscle Aches and Pain	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	Negative
13	P11	Swollen Lymph Nodes	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	Negative
14	P12	Fever	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	Positive
15	P13	Swollen Lymph Nodes	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	Positive
16	P14	Swollen Lymph Nodes	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	Negative
17	P15	Swollen Lymph Nodes	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	Positive
18	P16	None	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	Positive
19	P17	None	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	Positive
20	P18	Muscle Aches and Pain	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	Negative
21	P19	Swollen Lymph Nodes	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	Positive
22	P20	Fever	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	Negative
23	P21	None	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	Negative
24	P22	Swollen Lymph Nodes	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	Positive
25	P23	Fever	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	Positive
26	P24	Swollen Lymph Nodes	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	Positive
27	P25	Muscle Aches and Pain	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	Negative
28	P26	Muscle Aches and Pain	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	Negative
29	P27	None	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	Positive
30	P28	None	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	Negative

Lesion Images

The "MSLD" was developed by gathering and analyzing images from various online sources, including websites, portals, and public case reports. To differentiate Monkeypox patients from cases with similar symptoms, the "Lesion Dataset" was created, focusing on conditions that closely resemble the Monkeypox rash and initial-stage pustules. The proposed lesion images is published by [15] on Kaggle, MSLD v2.0 comprises images from six distinct classes, namely Monkeypox (284 images), Chickenpox (75 images), Measles (55 images), Cowpox (66 images), Hand-foot-mouth disease or HFMD (161 images), and Healthy (114 images). The dataset includes 755 original skin lesion images sourced from 541 distinct patients, ensuring a representative sample. The lesion images sample data is shown In Figure 1.

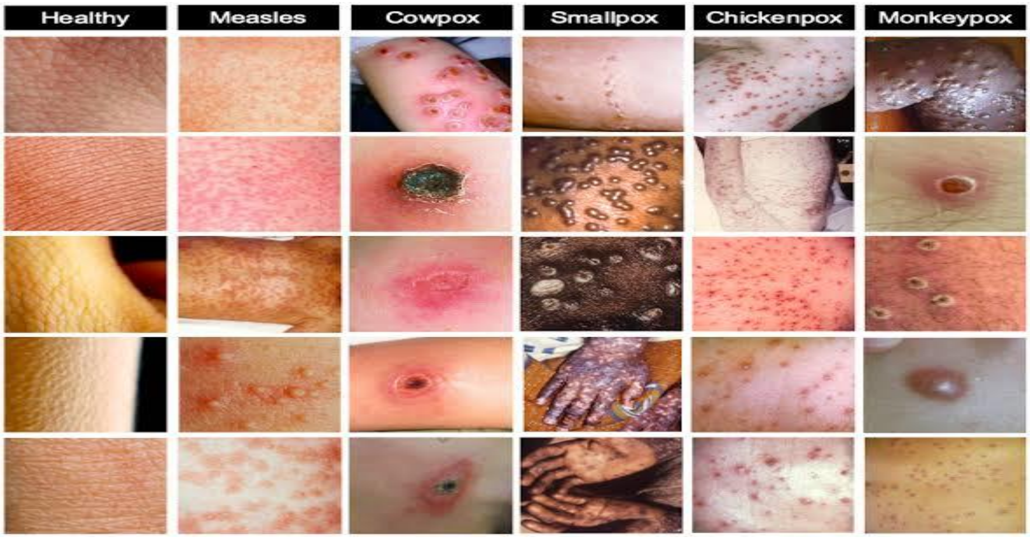


Figure 1: Lesion Sample Data

The system implements a bi-modal approach combining CNN-based image analysis (f_1) and XGBoost-based symptoms analysis (f_2) for Monkeypox prediction. The architecture utilizes transfer learning with VVG16 (pre-trained on ImageNet) for visual feature extraction.

Training Process

CNN Training (f_1)

The model leverages transfer learning by utilizing a pre-trained VGG16 architecture, where the lower layers (feature extraction layers) are frozen, meaning their weights are not updated during training. This allows the model to retain the knowledge acquired from a large dataset (such as ImageNet) while focusing on fine-tuning the more specific layers for the given task. Specifically, the top layers, which are responsible for classification or high-level feature interpretation, are fine-tuned. This means the weights in these layers are adjusted during training to better fit the new task at hand.

For the optimization process, binary cross-entropy loss is used, which is particularly suitable for binary classification tasks. It measures the difference between the predicted probability distribution and the true labels, guiding the model to minimize this loss during training.

To enhance training efficiency and performance, the Adam optimizer is employed, which adapts the learning rate for each parameter, making it effective for complex tasks. Additionally, learning rate scheduling is implemented, allowing the learning rate to adjust over time based on the training process. This technique helps improve model convergence and overall performance by dynamically controlling the rate at which the model learns during different stages of training.

The CNN model was trained on the preprocessed dataset using a loss function and an optimizer. CNN detected features like edges and patterns relevant to Monkeypox lesions

$$O(x,y)= (I * F) * (x, y) = \sum_{m,n} I(m,n)\cdot F(x-m,y-n) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(1)$$

$$where\ \ I\ is\ input\ image,\ F\ is\ filter,\ O\ is\ output\ feature\ map,$$

$$(x,y)\ is\ coordinates\ of\ output\ feature\ map$$

XGBoost Training (f_2)

The model employs a series of advanced techniques to ensure optimal performance and generalization. First, it leverages **hyperparameter optimization through cross-validation**, systematically testing different hyperparameter values using multiple subsets of the data to identify the combination that yields the best results across all folds, ensuring the model is robust and not overly tuned to one particular set of data. To prevent **overfitting**, it incorporates **early stopping**, monitoring the model's performance on a validation set during training, and halting the process if the model's performance starts to degrade, thereby avoiding the risk of memorizing noise and enhancing its ability to generalize to new data. Lastly, it conducts **feature importance analysis**, which evaluates how much each feature contributes to the model's predictions, helping to identify the most influential variables, refine the model's inputs, and improve interpretability and performance by potentially removing less relevant features.

Adapting [16] detection of Monkeypox cases based on symptoms using XGBoost and Shapley additive explanations methods, XGBoost is an optimized distributed gradient boosting toolkit that has been built to be effective, adaptive, and portable [16] and [17]. Chen and Guestrin created the XGBoost algorithm in 2016 [18]. It offers parallel tree boosting and is an improved variant of the GBDT (Gradient Boosted Decision Tree) approach (also known as GBM) [19]. The model's anticipated output ŷ can be calculated using an input feature vector $x = [x_1, x_2,\ldots, x_n]^T$ as follows:

$$\hat{y} = \sum_{k=1}^{k} f_k(x),\ f_k \in \Gamma \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. (2)$$

where K stands for how many weak learners there are. The weak learner's hypothesis space, Γ, represents the function $f_k(x)$, which is a prediction score [20] and [17].

The XGBoost symptom classifier produced a probability vector p for each image, where p indicates the probability that the image belongs to the "Monkeypox class". For each patient, a probability "p_symptom" is calculated to estimate the likelihood of the patient having Monkeypox based on their symptoms.

$$\mathrm{p}_{\mathrm{symptom(i)}} = \sum_{j=1}^{m} (x+a)^n \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.... (3)$$

$$\mathrm{where}\ \mathrm{x(i)}\ is\ feature\ vector\ for\ the\ ith\ patient,$$

Stacked Ensemble Training

The secondary XGBoost model was trained by incorporating the predictions from earlier models, effectively using these predictions as features to enhance the model's performance. During the training process, the model undergoes optimization of its feature weights to determine the most influential factors and improve overall accuracy. To ensure the model generalizes well to unseen data, cross-validation techniques are used for rigorous evaluation, allowing for the assessment of its performance across multiple subsets of the dataset, and reducing the risk of overfitting. This process ensures a more robust and reliable model.

Stacked generalization [21] works by deducing the biases of the generalizer(s) with respect to a provided learning set. To obtain the good linear combination of the base learners in regression, cross-validation data and least squares under non-negativity constraints was used to get the 2 optimal weights of combination [22]. Consider the linear combination of the predictions of the base learners f1, f2, · · · , fm given as:

$$f_{stacking}(x) = \sum_{j=1}^{m} w_j f_j(x) \qquad (4)$$

where w is the optimal weight vector learned by the meta learner

The Stacking Ensemble process involves combining predictions from the CNN and the symptom-based XGBoost model into a new dataset. Each row represents a patient with features from both models indicate the likelihood The XGBoost model was trained using this combined dataset to produce the final probability of Monkeypox based on both the image and symptom data as shown in equation 3 below.

$$\mathrm{p}_{\mathrm{final(i)}} = \; = \sum_{j=1}^{m} \eta \; * \; \mathrm{Tj}\left(\mathrm{p}_{\mathrm{cnn(i)}}, \mathrm{p}_{\mathrm{symptom(i)}}\right) \qquad (5)$$

T_j is $\mathrm{j}_{\text{th decision tree in the ensemble,}}$, η is learning rate

XGBoost minimized a loss function for example, Log Loss to optimize the model's predictions as shown in equation.

$$\mathrm{L}(\mathrm{y}, \mathrm{p}_{\mathrm{final(i)}}) = \frac{1}{N}\sum_{i=1}^{N}[\mathrm{y(i)} * \log(\mathrm{p\_final(i)}) + \; (1 - \mathrm{y(i)}) \; * \; \log(1 - \mathrm{p\_final(i)})] \qquad (6)$$

$$where\ y(i)\ is\ true\ label\ (0\ or\ 1)\ indicating\ presence\ or\ absence\ of\ MonkeyPox\ for\ the\ ith\ patient$$

Image Processing Pipeline (f_1)

In the image processing pipeline (f_1), the input image, is denoted as:

Input image: X_{img} ∈ ℝ^(H×W×3) (RGB image) …………………………(7)

This represents an RGB image with height (H), width (W), and three color channels (Red, Green, and Blue). The first step in the pipeline involves resizing the image to a consistent size of 224×224 pixels. This ensures that all images fed into the model have the same dimensions, which is important for efficient processing and uniformity. Following the resizing step, the image undergoes normalization, where each pixel value is transformed using the given formula:

Normalization: x' = (x - μ)/σ …………………………………………… (8)

Where, 'x' represents the original pixel value, 'μ' is the mean of the pixel values across the dataset, and 'σ' is the standard deviation of the pixel values.

Normalization enhance to standardize the image data by adjusting pixel values to a common scale, typically improving the model's performance by reducing sensitivity to varying lighting conditions and intensities.

Additionally, during the training phase, various data augmentation techniques are applied to artificially expand the dataset. These techniques include transformations such as rotation, flipping, cropping, and color adjustments. Data augmentation prevented overfitting by creating slightly altered versions of the input images, allowing the model to generalize better and improve its robustness in handling diverse real-world images.

Feature Extraction

Feature extraction process was performed utilizing the VGG16 architecture with pre-trained ImageNet weights. It also incorporates a deep residual learning framework that includes skip connections, with the residual block formula defined as:

$$H(x) = F(x) + x \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (9)$$

Where F(x) is the residual mapping and x is the identity mapping

The residual learning building block is defined as:

$$y = F(x, \{Wi\}) + x \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (10)$$

Where x and y are the input and output vectors of the considered layers, respectively.

Transfer Learning Implementation

The transfer learning implementation begins with loading the base VGG16 model pre-trained on ImageNet weights, with the final classification layer removed. The feature vector is extracted, represented as

$$\varphi(X_{img}) \in \mathbb{R}\text{^}2048 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \quad (11)$$

The above representation denotes that the output as the feature extraction function φ transforms the input image X_{img} into a 2048-dimensional feature vector

Additional layers are then added, including global average pooling, dense layers with ReLU activation, dropout for regularization, and a final prediction layer with a sigmoid activation function. Together, these layers allow the model to take the extracted features, process them through a set of non-linear transformations, regularize the network to avoid overfitting, and then output a probability score for classification.

Symptoms Processing Pipeline (f_2)

The symptoms processing pipeline (f_2) begins with input processing, where the symptom vector is denoted as:

$$X\_{sym} \in \mathbb{R}\text{^}n \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (12)$$

Where X_{sym} is defined as a real-valued vector of length of n, n represents the number of symptoms

Each symptom is encoded using binary values to indicate presence or absence. Additionally, feature preprocessing is performed, which includes handling missing values and applying feature scaling or normalization to the data. Together, these preprocessing techniques ensured that the data is clean, consistent, and appropriately scaled, which allows the machine learning model to learn more effectively and improve its performance.

XGBoost Model

The XGBoost model utilizes gradient boosting decision trees (GBDT) with an objective function defined as:

$obj(\theta) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k)$ …………………… (13)

Where l is the training loss function, Ω is the regularization term, f_k represents the k-th tree

The prediction for an instance is computed as:

$\hat{y}_i = \sum_k f_k(x_i), f_k \in F$ …………………………..... (14)

Where f_k belongs to the space of regression trees, denoted as F.

Ensemble Model Integration

The ensemble model integrates predictions from both models by fusing their outputs. The combined prediction is represented as:

$P(y|X) = G(f_1(X\_{img}), f_2(X\_{sym}))$ ………………….. (15)

Where $f_1(X\_{img})$ is the CNN model, $f_2(X\_{sym})$ is the XGBoost model, G is the final ensemble function that combines these predictions.

Final Prediction

The ensemble XGBoost model combines both predictions with weights learned during training. The result is given by:

$y\_final = \sigma(w_1 \cdot f_1 + w_2 \cdot f_2 + b)$ …………………..... (16)

Where σ represents the sigmoid activation function, w_1 and w_2 are the weights learned by the both models during training,

RESULTS AND DISCUSSION

Performance Comparison:

The performance of each model is measured by four key metrics: accuracy, precision, recall, F1-Score, which are defined as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$\text{F1} - \text{Score} = 2 * \frac{precision * recall}{precision + recall}$$

where (Tp) is true positive; (Tn), true negative; (Fp), false positive; and (Fn), false negative

These metrics are commonly used in machine learning to evaluate the performance of classification models. To compare the performance of the models, we calculated the values of these four metrics for each model, then compare the values to see which model has the highest performance. The model with the highest performance is considered the best model for prediction accuracy. In addition to these metrics, we also used other techniques to evaluate the performance of the models. For example, we used confusion matrices (Figure 6), which are a way to visualize the number of true positive, true negative, false positive, and false negative predictions made by the model. Table 5 and 6 shows comparison results of the proposed model with another, and then Table 7 illustrates the ensemble model performance metrics.

The model performance metrics is tracked using the following, as displayed in Table 2, 3 and 4 below.

CNN Model (f₁) Performance

Table 2: The performance metrics of the proposed system

Metric	Training	Validation	Test
Accuracy	0.945	0.921	0.918
Precision	0.938	0.915	0.912
Recall	0.952	0.928	0.925
F1-Score	0.945	0.921	0.918

XGBoost Model (f₂) Performance

Table 3: XGBoost Performance metrics

Metric	Training	Validation	Test
Accuracy	0.932	0.912	0.908
Precision	0.925	0.905	0.901
Recall	0.940	0.919	0.915
F1-Score	0.932	0.912	0.908

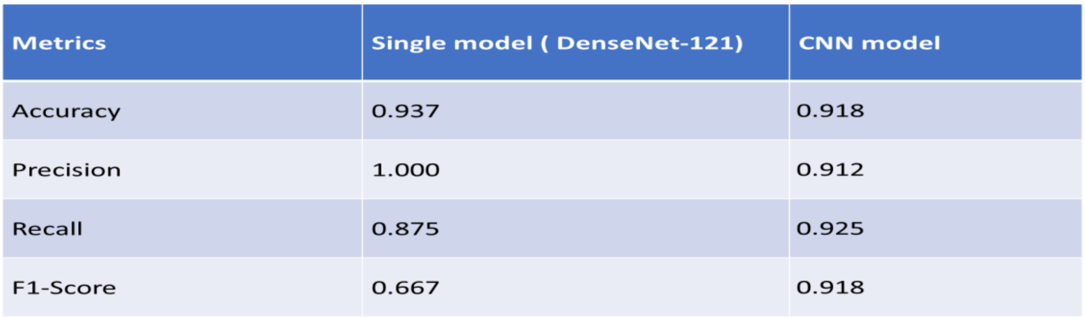
Ensemble Model Performance

Table 4: Ensemble model performance metrics

Metric	Training	Validation	Test
Accuracy	1	1	1
Precision	1	1	1
Recall	1	1	1
F1-Score	1	1	1

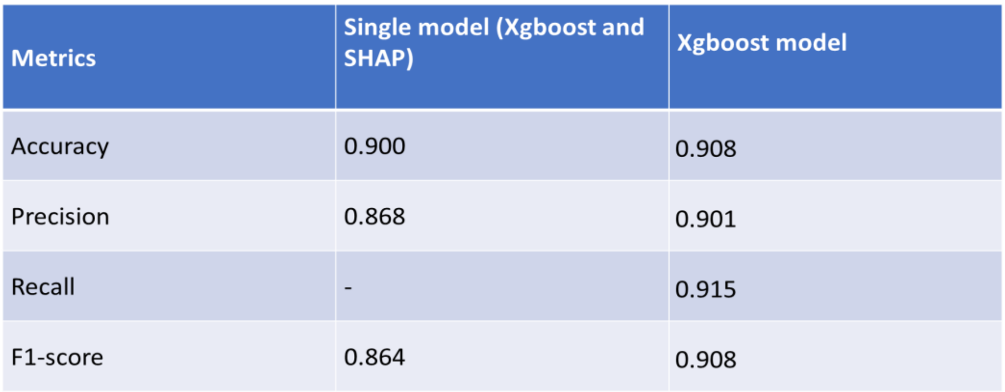
The comparison of performance metrics for the CNN (symptom) model, between a model utilizing DenseNet-121 as a single model and the proposed model, is shown in Table 5. The analogy of performance metrics for the clinical (symptom) model, between a model that used XGBoost and SHAP as a single model and the proposed model, is presented in Table 6 using standard metrics. The proposed Hybrid model, which combines CNN and XGBoost to detect Monkeypox through both images and symptoms, achieved an accuracy of 1.0, as shown in Table 7. In contrast, the other two journals analyzed [23] and [13] relied solely on either symptoms or images for Monkeypox detection, achieving accuracies of 0.900 and 0.937, respectively.

Table 5: Comparison of CNN Model with single Model



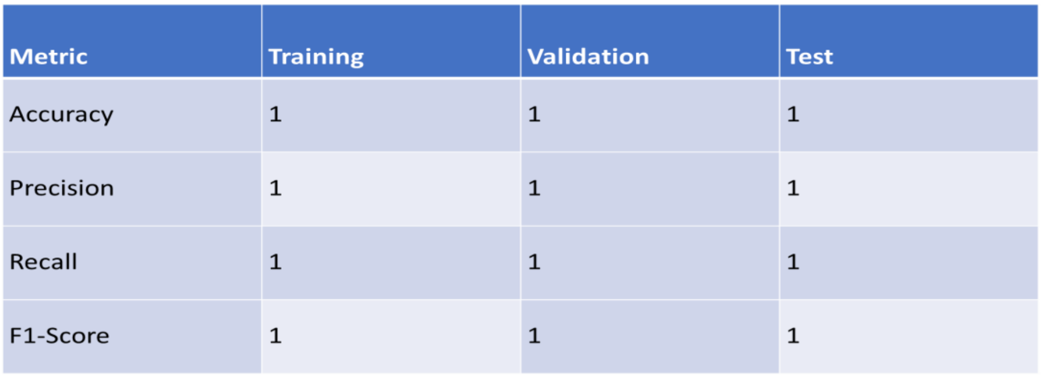
Metrics	Single model (DenseNet-121)	CNN model
Accuracy	0.937	0.918
Precision	1.000	0.912
Recall	0.875	0.925
F1-Score	0.667	0.918

Table 6: Comparison of XGBoost Model with single Model



Metrics	Single model (Xgboost and SHAP)	Xgboost model
Accuracy	0.900	0.908
Precision	0.868	0.901
Recall	-	0.915
F1-score	0.864	0.908

Table 7: Ensemble Model Performance metrics



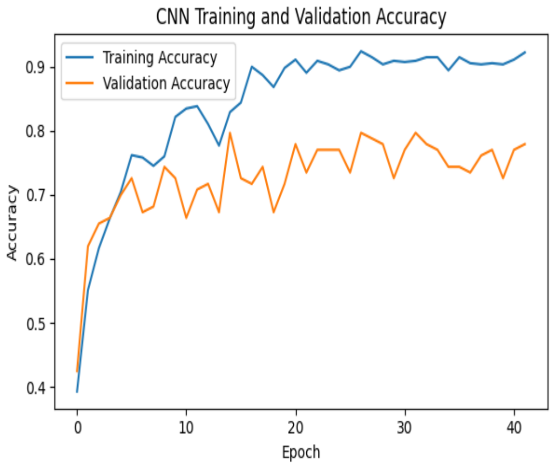
Metric	Training	Validation	Test
Accuracy	1	1	1
Precision	1	1	1
Recall	1	1	1
F1-Score	1	1	1

The model achieved a probabilistic accuracy of 1.0 by integrating multiple deep learning models, algorithms, ensemble methods, and techniques such as normalization, data augmentation, and optimization. It also leveraged the strengths of both the CNN and XGBoost models to further enhance performance.

The training of the Convolutional Neural Network (CNN) was evaluated across 40 epochs, with performance metrics monitored for training and validation sets. The training and validation accuracy, loss, precision, and recall are visualized in Figure 2, 3, 4 and 5(graphs shown below).

The training accuracy steadily increased during the initial epochs, reaching a plateau around 92% by epoch 25. This suggests that the CNN effectively learned features from the training images. The validation accuracy also improved over time, reaching a peak around 80% before leveling out. The validation curve displays some fluctuations, especially between epochs 5 to 15. The validation accuracy, though consistently lower than training, has reached a respectable peak and the gap between training and validation accuracy is not as high as in previous training iterations. Training loss decreased sharply in the initial epochs and converged towards zero, indicating successful learning on the training dataset. Validation loss also decreased, and stabilized to a low level, indicating that the model generalized well to the validation set after some early fluctuations. The overall trend was of a small downward slope with some instability over time. Training precision increased steadily and reached about

95%, while the validation precision also followed a similar trend, plateauing at around 80%. The difference between training and validation precision was small, indicating that the positive predictions of the model were generally accurate. The training recall achieved a very high value of above 90% at the end of training. Validation recall reached a maximum value of about 80%, and also leveled out at that value.



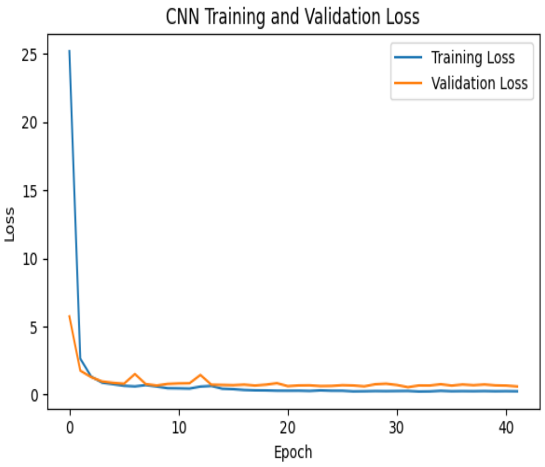
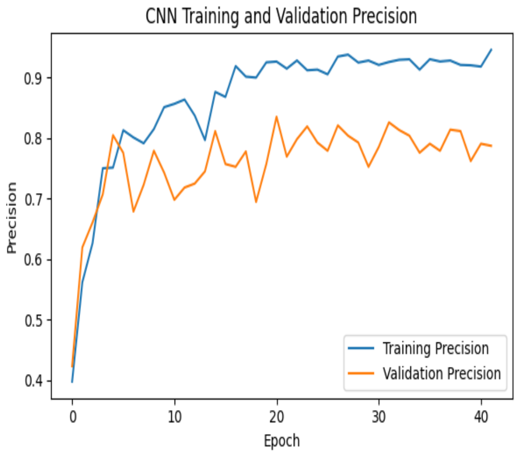


Figure 2: CNN Accuracy graph

Figure 3: CNN Loss graph



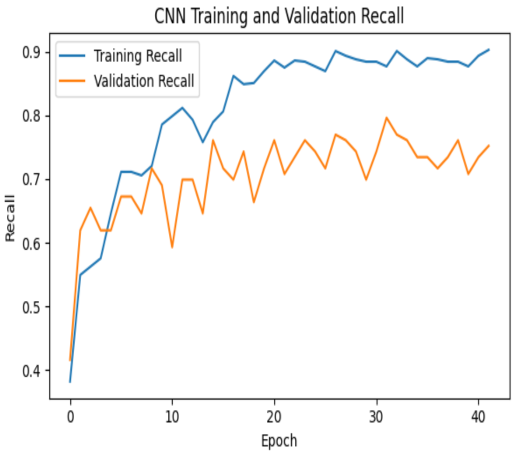


Figure 4: CNN Precision graph

Figure 5: CNN Recall graph

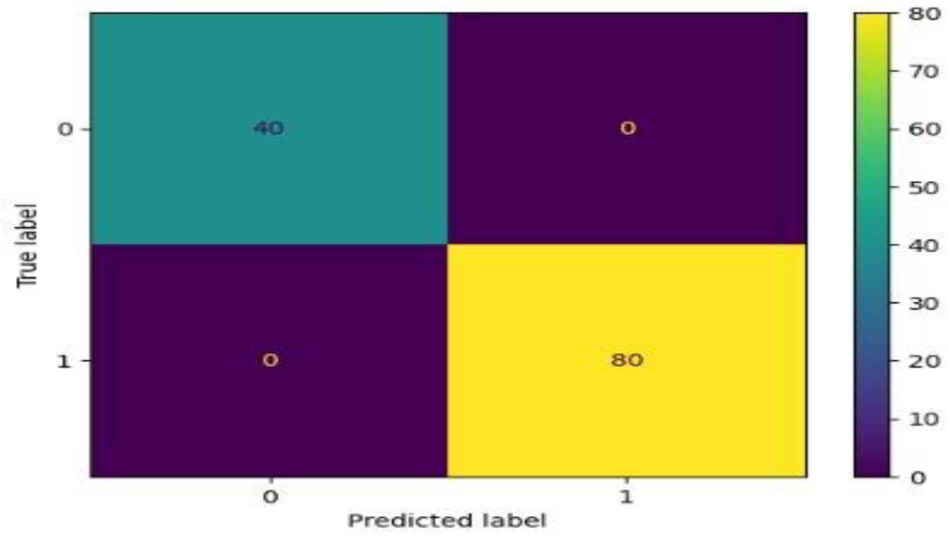


Figure 6: Confusion matrix

The SHAP (Shapley Additive Explanations) plot in Figure 7, shows the relative importance of different symptom features in the combined model. The SHAP plot reveals that 'Systemic Illness', 'HIV Infection', and 'Rectal Pain' have the greatest impact on the combined model's output. The relative importance of ‘Patient ID’ may reflect the importance of individual variations in responses to different risk factors which may not be easy to discern as explicit features. Other symptoms, like ‘Swollen Tonsils’, ‘Oral Lesions’, and ‘Solitary Lesion’ have lower importance in the combined model. The SHAP values also indicate whether a feature is associated with a higher or lower risk of Monkeypox. For instance, ‘Systemic Illness’ and ‘HIV Infection’ are more correlated with positive Monkeypox classifications, while ‘Solitary Lesion’ was correlated with negative classifications. The SHAP analysis provided insightful information about the importance of various symptoms, which may also provide insights for clinicians by highlighting the most important symptoms to look for when diagnosing Monkeypox as illustrated in figure 7. These insights can lead to better screening practices and improved diagnoses.

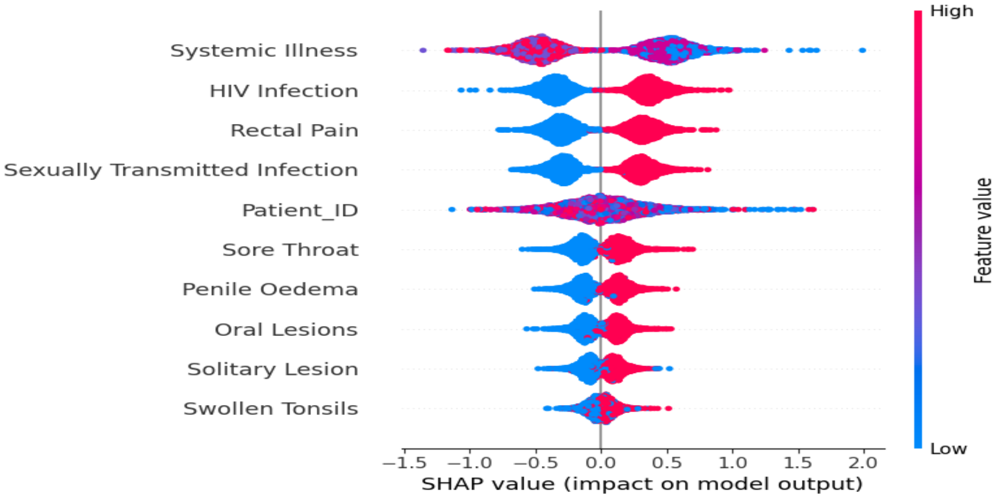


Figure 7: SHAP plot (XGBoost)

CONCLUSION

This research demonstrated the potential of a Bi-modal approach to detect Monkeypox by combining visual features from CNNs with symptoms information using XGBoost. The CNN demonstrated the ability to learn visual features from images, while the XGBoost model identified the risk factors that are associated with Monkeypox. The outputs of the two algorithms was ensembled using Stacked Generalization by implementing XGBoost. The combination of these two models resulted in a more robust classification performance. The experimental results indicated that the Hybrid model outperformed other methods used in comparison, reaching an accuracy of 1.0. The SHAP analysis provides useful insights into symptom importance, which can assist clinicians in better screening practices. The proposed model can be used by individuals conveniently to educate or self-diagnose on Monkeypox.

REFERENCES

1. Khan, Salem, and Ipseeta Ray. 2025. “Bioactive Phytochemicals for Human Monkeypox Outbreak”. Asian Journal of Research in Infectious Diseases 16 (2):24-43. https://doi.org/10.9734/ajrid/2025/v16i2421.
2. Srinivasan, M. N., Sikkandar, M. Y., Alhashim, M., & Chinnadurai, M. (2025). Capsule network approach for monkeypox (CAPSMON) detection and subclassification in medical imaging system. Scientific Reports, 15(1), 3296. https://doi.org/10.1038/s41598-025-87993-7
3. Veysel, H.S., Ismail O., & Godzel Y.O. (2022). Human Monkeypox classification from skin lesion images with deep pre-trained network using mobile application.
4. Abafi, D. S. O. F., Adewumi, M. O., Michael, O. O., Kia, G. S. N., Adeyemo, O. K., Alarape, S. A., Meseko, C. A., Oyetunde, J. S., Mkpuma, N., Shorunke, F. O., & Aiki-Raji, C. O. (2024). Molecular

detection of monkeypox virus in wild rodents and humans in Ibadan, Nigeria; a cross-sectional study. Pan African Medical Journal - One Health, 13(16). https://doi.org/10.11604/pamj-oh.2024.13.16.42143

5. The Nigerian Tribune. (2024). Monkeypox spreads: 44 cases, one death recorded in West Africa – ECOWAS. Retrieved August 19, 2024, from https://www.tribuneonlineng.com/Monkeypox-spread-44-cases, one-death-recorded-in-west-africa
6. The Express Tribune. (2024). Another suspected case of mpox reported in Islamabad. Retrieved August 19, 2024, from https://www.tribune.com.pk/story/2489257/Another-suspected-case-of-mpox-reported-in-Islamabad.
7. Dada, E.G., Oyewola, D.O., Joseph, S.B., & Dauda, A.B. (2021). Ensemble machine learning model for software defect prediction. Advances in Machine Learning and Artificial Intelligence, 2(1), 11–21.
8. Tom, J., & Anebo, N. P. (2018). A Neuro-Fussy Based Model for Diagnosis of Monkeypox Diseases. International Journal of Computer Science Trends and Technology, 6(2), 143–153.
9. Liu, Y., Jain, A., Eng, C., Way, D. H., Lee, K., Bui, P., Kanada, K., de Oliveira Marinho, G., Gallegos, J., Gabriele, S., et al. (2020). A deep learning system for differential diagnosis of skin diseases. Nature Medicine, 26(6), 900–908. https://doi.org/10.1038/s41591-020-0842-3
10. Şahin, V. H., Öztel, İ., & Yolcu Öztel, G. (2022). Human monkeypox classification from skin lesion images with deep pre-trained network using mobile application. Journal of Medical Systems, 46(11), 79. https://doi.org/10.1007/s10916-022-01863-7
11. Attallah, O. (2023). MonDiaL-CAD: Monkeypox diagnosis via selected hybrid CNNs unified with feature selection and ensemble learning. Digital Health, 9, 20552076231180054. https://doi.org/10.1177/20552076231180054
12. Altun, M., Gürüler, H., Özkaraca, O., Khan, F., Khan, J., & Lee, Y. (2023). Monkeypox Detection Using CNN with Transfer Learning. Sensors, 23(4), 1783. https://doi.org/10.3390/s23041783
13. Farzipour, A., Elmi, R., & Nasiri, H. (2023). Detection of monkeypox cases based on symptoms using XGBoost and Shapley additive explanations methods. Diagnostics, 13(14), 2391. https://doi.org/10.3390/diagnostics13142391
14. Gokula, K. V., Liya, B. S., Lakshmi, S. V., Sathyamoorthy, K., & Ganesan, S. (2023). Monkeypox detection using hyper-parameter tuned based transferable CNN model. International Journal of Experimental Research and Review, 33(Special Issue), 18–29. https://doi.org/10.52756/ijerr.2023.v33spl.003
15. Ali, S. N., Ahmed, M. T., Paul, J., Jahan, T., Sani, S. M. S., Noor, N., & Hasan, T. (2022). Monkeypox skin lesion detection using deep learning models: A preliminary feasibility study. arXiv preprint arXiv:2207.03342. https://doi.org/10.48550/arXiv.2207.03342
16. Chehreh Chelgani, S., Nasiri, H., & Tohry, A. (2021). Modeling of particle sizes for industrial HPGR products by a unique explainable AI Tool-A “Conscious Lab” development. Advanced Powder Technology, 32(12), 4141-4148.
17. Nasiri, H., & Alavi, S.A. (2022). A novel framework based on deep learning and ANOVA feature selection method for diagnosis of COVID-19 cases from chest X-ray images. Comput. Intell. Neurosci., 2022, 4694567.
18. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 785–794).
19. Fatahi, R., Nasiri, H., Dadfar, E., & Chehreh Chelgani, S. (2022). Modeling of energy consumption factors for an industrial cement vertical roller mill by SHAP-XGBoost: A “Conscious Lab” approach. Sci. Rep., 12, 7543.
20. Song, K., Yan, F., Ding, T., Gao, L., Lu, S. (2020). A steel property optimization model based on the XGBoost algorithm and improved PSO. Comput. Mater. Sci., 174, 109472. SHAP for real-time accident detection and feature analysis. Accid. Anal. Prev., 136, 105405.
21. Wolpert, D.H. (1992). Stacked generalization. Neural Networks, 5, 241–259. Fang, Y., Zhang, H., Xie, J., Lin, M., Ying, L., Pang, P., & Ji, W. (2020). Sensitivity of chest CT for COVID-19: Comparison to RT-PCR. Radiology, 296(2), E115-E117.
22. Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2), 123–140.
23. Farzipour, A., Elmi, R., & Nasiri, H. (2023). Detection of Monkeypox cases based on symptoms using XGBoost and Shapley additive explanations methods. MDPI Diagnostics, p 1-16. https://doi.org/10.3390/diagnostics 13142391