

Multi-Disease Prediction Website and Android Application Using Explainable Machine Learning and Deep Learning Techniques

Abu Tareq

North South University
Dhaka, Bangladesh
abu.tareq
@northsouth.edu

Abdullah Al Mahfug

North South University
Dhaka, Bangladesh
abdullah.mahfug
@northsouth.edu

Mohammad Imtiaz Faisal

North South University
Dhaka, Bangladesh
imtiaz.faisal03
@northsouth.edu

Tanvir Al Mahmud

North South University
Dhaka, Bangladesh
tanvir.mahmud15
@northsouth.edu

Dr. Riasat Khan

North South University
Dhaka, Bangladesh
riasat.khan
@northsouth.edu

Abstract—As the world advances towards a more computerized future, disruptive technologies are arriving at a pace second to no other innovation in history. The rise of computer vision and artificial intelligence techniques empowered innovations in the healthcare sector is helping to save lives, detect diseases and extend life expectancy. This paper uses machine learning and deep learning techniques that fall into supervised learning to detect five diseases, e.g., diabetes, coronavirus, cardiac disorder, liver illness, and chronic kidney disease. Consequently, the automatic disease prediction system has been deployed on a website and an Android smartphone application. Initially, we enter necessary values from the pathology report or insert X-ray images into the website or the Android application to predict whether the person suffers from these diseases or not. The prediction is made in real-time using machine learning models trained on various open-source datasets of different illnesses. We have used Random Forest as the supervised machine learning model as it produced an accuracy of 85%, 97%, 75%, and 85% to predict diabetes, cardiac disorder, liver illness, and chronic kidney disease, respectively. We have employed an attention-based CNN model which produces a validation accuracy of 92% to predict coronavirus. Explainable AI models were used to interpret them.

Index Terms—CNN, COVID-19, Dataset, Disease, KNN, Logistic Regression, Decision Tree, Pre-Processing, Random Forest, Attention module, SVM.

I. INTRODUCTION

Healthcare in Bangladesh suffers from a scarcity of doctors/nurses, absenteeism, inefficiency, and mismanagement according to WHO. [1]. Heart disease, stroke, diabetes, and other chronic diseases are expected to account for half of all deaths in the WHO Southeast Asian countries, among which Bangladesh is a constituent. [2]. The Kidney Foundation expressed that 35000-40000 out of 18 million get chronic kidney failure in Bangladesh yearly [3]. Nonalcoholic Fatty Liver Disease prevalence in Bangladesh's general population ranges from 4 to 18.4%, while it rises to 49.8% in diabetic patients. [4]. As of May 2022, 1,953,379 confirmed COVID-19 cases with 29,130 deaths in Bangladesh [5]. In Bangladesh, coronary heart disease is the second biggest cause of mortality, followed by diabetes in fifth place and liver and chronic kidney

diseases in eighth and ninth place. Early detection of these diseases can lead to a lower mortality rate for Bangladeshis, but the ineffectual healthcare system of Bangladesh does not make it easier to do so. This complex scenario has propelled us to create a website and an Android application that can predict these diseases from the inputs given by the user. As medical consultation is expensive in Bangladesh, users can predict whether they are affected by these diseases from their homes.

As the numbers suggest, Diabetes, Coronavirus, Heart disease, Liver illness, and Chronic Kidney disease are the leading causes of death among Bangladeshis. With the availability of medical data in the public domain and the parallel growth of AI, serious efforts have been made to make early predictions of the mentioned diseases. To address the current difficulties with disease prediction, the researchers used cutting-edge machine learning and deep learning technology. Some existing solutions that refer to solving these issues have been reviewed below. The drawbacks of the existing solutions are also briefly discussed below.

In [6], N. Kumar and his colleagues created an Android smartphone application known as Disease Prediction Using Artificial Intelligence (DPAI) that can predict diseases like Diabetes, Coronavirus, and heart disease using a combination of three open-source datasets: Novel Corona Virus 2019 Dataset, Cardiovascular Disease Dataset, and Pima Indians Diabetes Database. The authors recommend that when a user enters data into the app, the data will be processed in an instantaneous database using a previously trained ML classifier generated on the same dataset, which is sent to Firebase. The prediction given by the classifier would be eventually shown in the app. The researchers have implemented ten different algorithms on the datasets, which KNN, ANN, RF, GB, SVM, ANFIS, GANFIS, LR, J48, and proposed Logistic Regression outperforms the other algorithms in different performance measures. To complete the project, they employed Python 3 and Java 10.0.2, along with Android Studio 3.1.0 as well as Jupyter Notebook 5.5.0. Their proposed model has an accuracy of 1.4765%. 1.2782 was the F-measure score, respectively, for

COVID. In addition, they were able to achieve a 1.8274 % accuracy. An F-measure of 1.7264 was found for the diabetes dataset, respectively. Finally, the researchers found an accuracy of 1.7362% and a 1.3821 F-measure for heart disease.

In [7] T. Wang et al. proposed a procedure that predict multiple disease detection by manipulating patients' diagnostic medical information to assess future disease risks systematically. To meet the needs of other contributors, medical diagnoses depending on the International Classification of Diseases (ICD) are separated into different classes for prediction. In their proposal, they validated using medical datasets from two independent hospitals. Their research showed that the Recurrent-Neural-Network interacts well with short- and long-term memory units at various levels of diagnostic aggregation.

In [8] A. Yaganteeswarudu presented a systematic view that could hold multiple disease predictions in a single platform using Flask API. In his approach, many more diseases can be included in the future. The exciting thing in his paper is that when analyzing diseases, all parameters that cause the disease are comprehended. As a result, the full extent of the disease's impact can be seen. And this will help to monitor the patient's present status and compare with his/her situation in advance, which will decrease the mortality ratio.

In [9] L. Men et al. proposed an approach that performs multi-disease prediction using deep learning. They utilize the technique by utilizing a Long-Short-Time-Memory network(LSTM) and broaden it to two independent appliances: time-aware and attention-based. They used time-aware to manage inconsistency in timing between clinical visits and attention-based technique aids to find the relevance of each visit of the prediction. They collected the dataset from a Southeast China hospital with 5M records. Then they tried to find the different time intervals between choices of time-aware and attention-based approaches.

In [10] A. Ampavathi et al. focused on a system that deploys the multi-disease prediction using a deep learning approach. They utilized a dataset that relates to "Lung Cancer, Hepatitis, Alzheimer's Disease, Liver Tumor, Diabetes, Parkinson's Illness, and Heart Illness." The whole approach is divided into three parts: normalization of data ,prediction and weighted normalized feature extraction. The Jaya method-based multiverse optimization algorithm is used to optimize the weighting function by combining two metaheuristic algorithms (JAMVO). RNN and DBN hybrid deep learning techniques are used to expose the best attributes. The weights of these two classifiers are modified using the identical hybrid optimization method to match the hybrid deep learning architecture.

In [11], V. Jackins et al. used three datasets, NIDDK, Framingham heart study, and Wisconsin breast cancer dataset, from online repositories to detect heart disease, cancer, and diabetes using the random forest and naive Bayes algorithms. The researchers first preprocessed the data, filled in missing values and cleared associated columns. The algorithms are implemented to the preprocessed dataset, followed by random forest and naïve Bayes models. Then the authors used the efficiency calculations to compute and examine the accuracy

of the models. The Naive Bayes algorithm's accuracy was 82.35% for coronary heart disease 74.46% for diabetes, and 63.74% for cancer. In contrast, the accuracy of the Random Forest model was 83.85%, 92.40%, and 74.03% for coronary heart disease, cancer data, and diabetes, respectively. For all three diseases, the random forest model surpassed the Naive Bayes method in terms of accuracy. Finally, they compared the proposed algorithms to the DBSCAN and K-means clustering algorithms to see how successful they were. The random forest algorithm exceeds the other two algorithms, according to the findings.

In [12], the researchers utilized the R statistical software, it employs a variety of ML techniques for the early identification of heart disease. The software uses intelligent algorithms like logistic regression and naïve Bayes to predict the accuracy of heart diseases. They use the dataset related to the UCI learning repository, which is classified into two datasets training and testing. From there, they find some records and attributes obtained about the medical. The authors use data visualization techniques based on attributes that improve prediction efficiency in this work. Next, they compare the accuracy among the algorithms, and they found that logistic regression accuracy (91.61%) is better than others. The authors claimed that these accuracies could be increased with the help of better techniques and diverse datasets, which would help predict heart disease at an early stage.

In [13], K. F. Haque and the team planned a CNN-based approach for predicting COVID-19 disease. They used the patient's chest X-ray. For training purposes, they took 330 images, which they split into two groups, 'COVID-19' and 'Normal.' An evenly split picture set of 82 chest X-rays and used to validate the model. The model has 95.34% and 97.56% accuracy and precision, respectively. They compared that model with other distinct CNN models with multiple convolutional layers. In terms of F1 score and overall performance, comparative evaluations show that their proposed model is superior to the other two. As a result, CNN stands a decent chance of locating COVID-19 despite its restricted time, resources, and budget.

In [14], K. Arumugam et al., Because there is little evidence to be able to predict heart disease in diabetic patients, the researchers focused on a strategy to predict heart disease in these type of patients. They used the naive Bayes, SVM, and decision tree models for prediction purpose. The decision tree model gives the best accuracy with 90.1% of the three models.

In [15], Karthikeyan Harimoorthy and Menakadevi Thangavelu proposed a technique to predict diabetes, CKD, and heart disease. They collected dataset from the UCI repository. The SVM-Radial bias kernel method, SVM-Polynomial, SVM-Linear, Decision tree, and RF models have been assessed with precision, accuracy, and misclassification rate. Among all the machine learning algorithms, the SVM-Radial bias kernel method gives the best accuracy in the CKD, Diabetes, and heart disease datasets, at 98.3%, 98.7%, and 89.9%, respectively.

In [16], Indukuri Mohit, K. Santhosh Kumar, Avula Uday

Kumar Reddy, and Badhagouni Suresh Kumar created a disease prediction web application that can predict the diseases like Diabetes, Heart diseases, and Breast cancer. The application uses machine learning algorithms to predict the conditions, logistic regression gave the best accuracy prediction for diabetes and breast cancer with 77.60%, 94.55%, and KNN showed good accuracy for the heart disease prediction with 83.84%. They intend to expand their web application in the future by adding disease-related deep learning models and new machine learning-trained diseases.

In [17], Anil Kumar Dubey proposed a technique to predict multiple diseases using deep learning, and the datasets were collected from Kaggle and UCI repositories. The methodology follows some steps like Optimal Feature selection, Data Acquisition, feature Extraction, and prediction. The neurons in the hidden layer count of DBN and NN is carefully controlled or tuned using the same technique in both prediction systems, and the (LA) and (BOA) are mostly employed to forecast (L-BOA). According to the findings, The L-BOA-NN+DBN had a 6 percent higher accuracy than the SVM and 15.5 % better than KNN; In terms of predicting multiple diseases, the newly created L-BOA-NN+DBN has achieved best outcomes than others models.

Based on the literature reviews mentioned above, we can observe that a lot of work has been done relevant to our research topic. In their study, the researchers implemented a variety of classifiers to predict these illnesses. But we can observe that they did not consider the imbalance issues of their used dataset. When a dataset favours one class over another, a ML model trained on the same dataset would also favour that class, which would question the overall sustainability of the research. Another issue that we observed was that the predictions provided by the classifiers were used as a black box. There was no real explanation of how the classifier reached the outcome and what were the significant attributes that affected the prediction. We detected five diseases and created a website and an Android application for our proposed study. But, the researchers have not been able to amass this many diseases in their work and build a website and an Android application simultaneously.

The focus of this work is on the use of ML techniques and deep learning with the user's textual and image data to predict whether or not the user is infected with a specific disease. The following is the essential thing we perform at work:

- Machine learning models are used to create an effective automated illness diagnostic model in real-time through a website and an Android smartphone application.
- End-users can predict five critical diseases, i.e., COVID-19, heart disease, Liver disease, Diabetes, and Kidney disease, and observe the main contributing factors responsible for causing the disease.
- For COVID-19 detection, users need to upload a picture of their chest X-ray, and for the other diseases, they need to input their numerical diagnostics values.
- RF Classifier, LR, SVM, KNN, Decision Tree Classifier, and Extra Trees Classifier models were used to predict Dia-

betes, Liver Disease, CKD and Heart Disease .

- The outcome were compared using different performance metrics, and Random Forest outperformed the other classifiers in making the prediction. It has been used to create the final model for Liver Disease, Diabetes, Heart Disease, and Kidney Disease prediction.

- An attention-based CNN model has been used to predict COVID-19 on the image dataset.

- SMOTE has been used to overcome unbalanced issues related to the target variable.

- Hyperparameter tuning has been used to determine the optimal SMOTE ratio. Hyperparameter tuning was utilized to find the best set of parameters for the used classifiers.

- Explainable AI library Lime and Grad-CAM were used to interpret our results and determine the main contributing factors that impacted the prediction-making process.

The structure of the paper is outlined below: We discuss the background of our study, evaluate several relevant works, and identify gaps in section I. We describe the whole methodology of the project in section II. Afterward, we elaborate on Results and Discussion in section III. In section IV, we complete the paper.

II. PROPOSED SYSTEM

The methodology aims to forecast an individual's risk of contracting Diabetes, COVID-19, heart disease, liver disease, and chronic kidney disease, depending on the values inserted from the pathology report.

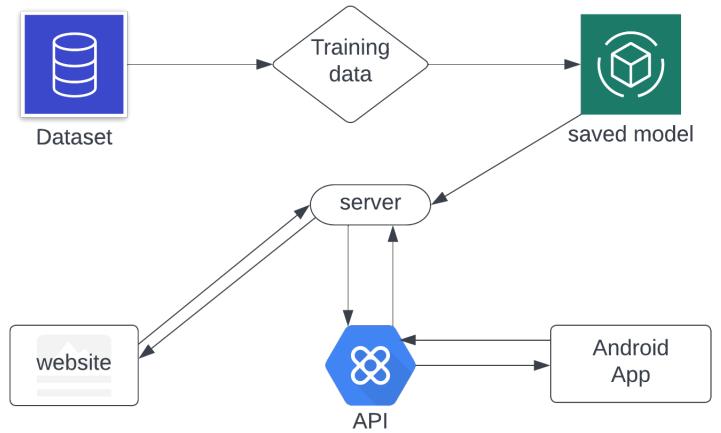


Fig. 1. The proposed project's flow chart diagram..

The flow chart for our proposed project is presented in Fig. 1. The various datasets were gathered from open-source repositories, and we have done several preprocessing steps to fill the null values. In addition, we transformed categorical features into numerical features. Pearson's Correlation Coefficient was utilized to identify the main attributes. Then, the data was split into two parts: Test and Train. Random Forest was used to create the Diabetes, Heart Disease, Liver Disease, and Kidney

Disease prediction model. The model was created using an attention-based CNN model for COVID-19 prediction. The models were then uploaded to the server and deployed. Then, to produce real-time predictions, we developed an Android application and a website. When a user inserts an image or the numerical diagnostic values through the website, it sends a request to the server, and then the server sends a response with the prediction. The same thing happens when a user uses the Android application to upload an image or numerical diagnostic values, but an API we designed acts as an intermediary between the server and the Android application.

A. Dataset and Preprocessing

In this work, we have used public datasets collected from various sources on the internet. The chronic kidney disease dataset [18] had 400 rows and 26 columns. In Table I, we can observe some of the dataset's essential contents. Fig. 2 shows the heatmap of the features. The dataset required preprocessing as it had missing values. For example, 152 entries from the red blood cells (RBC) column, 130 entries from the red blood cell count (RC) column, 105 from the white blood cell count (WC) were missing. We dropped the id column as it had nothing to do with the prediction. We filled the missing entries with the mean values and converted the categorical values to numerical values. 250 of 400 rows had harmful chronic kidney disease, but 150 had positive kidney disease. For diabetes detection, we used the Pima Indians Diabetes Dataset [19]. It had a row of 2768 rows and 9 columns. In Table II, we can see some of the dataset's contents. Fig. 3 shows the heatmap of the features. The dataset required preprocessing as it had missing values. For example, 1330 entries from the Insulin column, 1816 entries from the outcome column, 800 from the SkinThickness column were missing. We did not have to drop any columns as the correlation values were within range. We filled the missing entries with the mean values and converted the categorical values to numerical values. 500 out of 768 entries did not have diabetes, and 268 out of 500 had diabetes. The heart disease dataset [20] had 70000 rows and 13 columns. In Table III, we can see some of the dataset's important contents. 35021 out of 70000 entries were negative cases, and 34979 out of 70000 were positive cases. Fig. 4 shows the heatmap of the features. We used the Indian Liver Patients Records dataset [21], which had 583 rows and 11 columns. In Table IV, we can see some of the dataset's important contents. The feature heatmap is depicted in Fig. 5. Due to missing values, the dataset needed to be preprocessed.

For example, four entries from the Albumin and Globulin Ratio were missing. We did not have to drop any columns as there was no correlation between features. The dataset had 416 entries for the liver patient and 167 entries for the non-liver patient.

For COVID-19, we used COVID-19 Radiography Database [22]. It contains 6012 X-ray images of lung opacity, 1345 X-ray images of viral pneumonia, 3616 X-ray images of COVID-19, and 10192 X-ray images of regular patients. Since this dataset is imbalanced, we take 1345 X-ray images for

each class. We divide the entire dataset into training and validation, respectively, with 80% and 20%. The models are trained in batches of 32. Three subcategories are created from each training and validation dataset: 'COVID,' 'Normal,' and 'Viral Pneumonia,' containing various X-ray pictures. All photographs are transformed to 256x256 pixels to simultaneously retain unanimity and image quality. All photos were shuffled and converted to RGB color space. Fig. 6 shows sample images from the dataset. All the basic steps of supervised machine learning are summarized into a block diagram in Fig. 7. In Fig. 9, COVID-19 detection has been demonstrated using the proposed deep learning model.

TABLE I
FEATURES OF CHRONIC KIDNEY DATASET.

bp	rbc	wc	appet	pus cell	htn	hemo	dm	Anemia	Classification
80	5.2	44	good	normal	yes	15.4	yes	no	0
50		38	good	normal	no	11.3	no	no	0
80		31	poor	normal	no	9.6	no	yes	0
70	3.9	32	poor	abnormal	yes	11.2	yes	yes	0
80	4.6	35	good	normal	no	11.6	no	no	0

TABLE II
FEATURES OF PIMA INDIANS DIABETES DATASET.

age	pregnancies	bp	glucose	insulin	skin thickness	bmi	Diabetes pedigree function	outcome
39	8	72	133	0	0	32.9	0.27	1
46	5	62	44	0	0	25	0.587	0
24	2	58	141	128	34	25.4	0.699	0
42	7	66	114	0	0	32.8	0.258	1
32	5	74	99	0	27	29	0.203	0

TABLE III
FEATURES OF HEART DISEASE DATASET

age	gender	ap_hi	ap_low	Cholesterol	Gluc	smoke	alco	active	cardio
18393	2	110	80	1	1	0	0	1	0
20228	1	140	90	3	1	0	0	1	1
18857	1	130	70	3	1	0	0	0	1
17623	2	150	100	1	1	0	0	1	1
17474	1	100	60	1	1	0	0	0	0

TABLE IV
FEATURES OF INDIAN LIVER PATIENTS RECORDS DATASET.

age	gender	Total bilirubin	Direct bilirubin	Alkaline phosphatase	Alanine aminotransferase	Aspartate aminotransferase	Total proteins	Albumin and globulin ratio	dataset
62	male	10.9	5.5	699	64	100	7.5	0.74	1
62	male	7.3	4.1	490	60	68	7.0	0.89	1
38	male	1.0	0.4	182	14	20	6.8	1.00	1
64	female	0.6	0.2	197	17	19	7.8	0.90	1
72	male	3.9	2.0	195	27	59	7.3	0.40	1

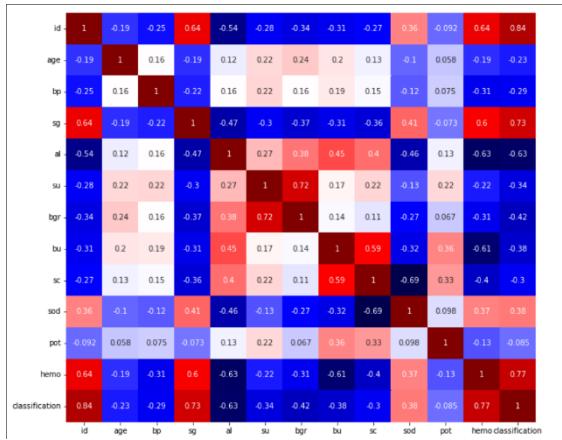


Fig. 2. Heatmap of chronic kidney dataset.

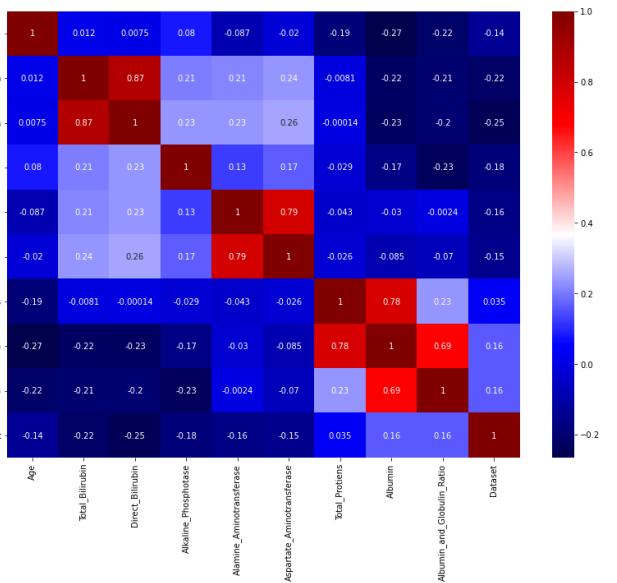


Fig. 2. Heatmap of chronic kidney dataset.

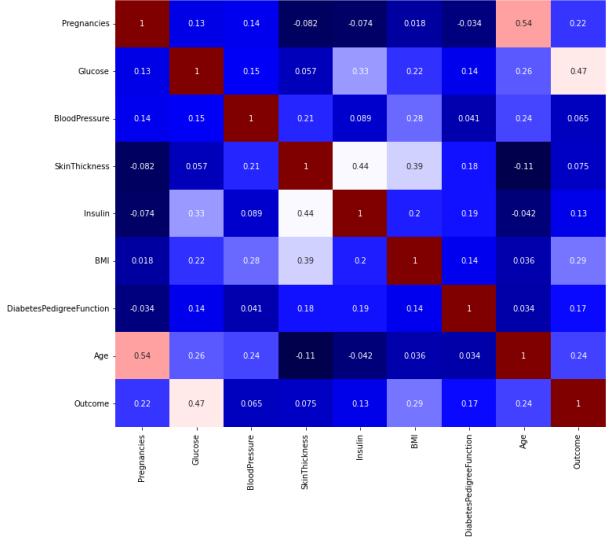


Fig. 3. Heatmap of Pima Indians Diabetes Dataset.

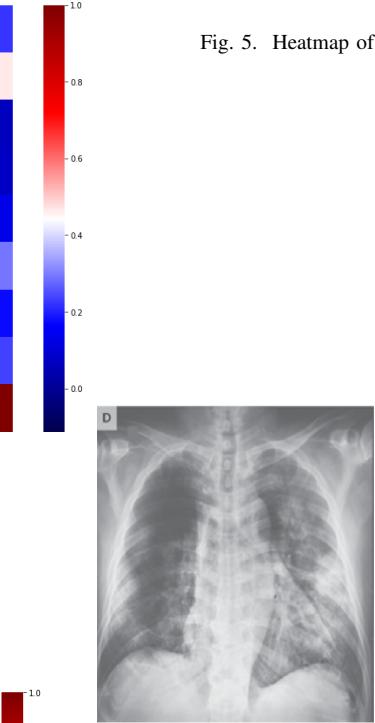


Fig. 5. Heatmap of Indian liver patients records dataset.

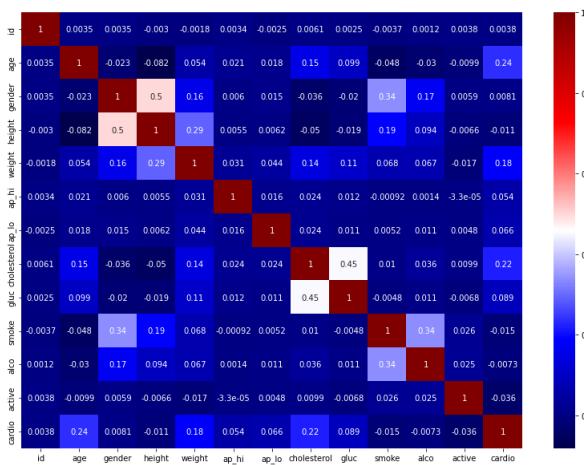


Fig. 4. Heatmap of heart disease Dataset.

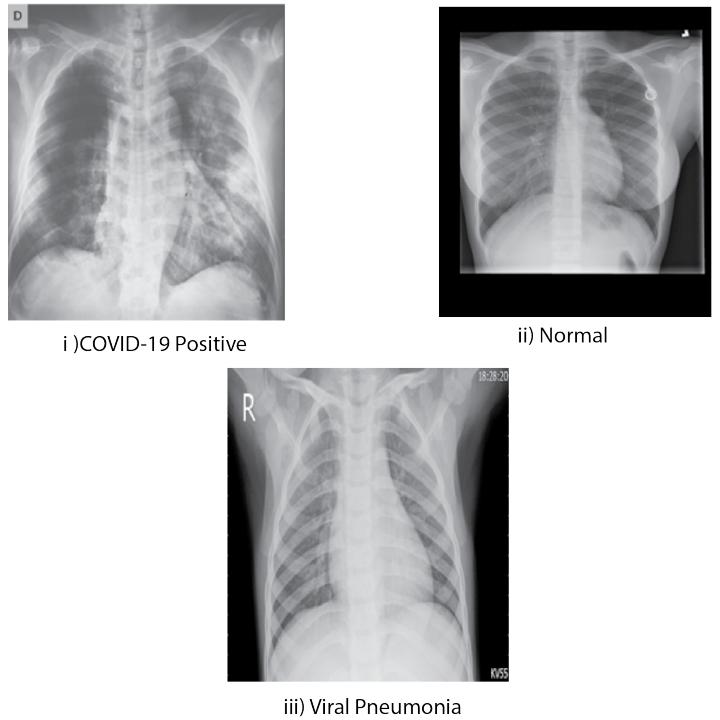


Fig. 6. Sample images from COVID-19 dataset : i) COVID-19 Positive img, ii) Normal img, and iii) Viral Pneumonia img.

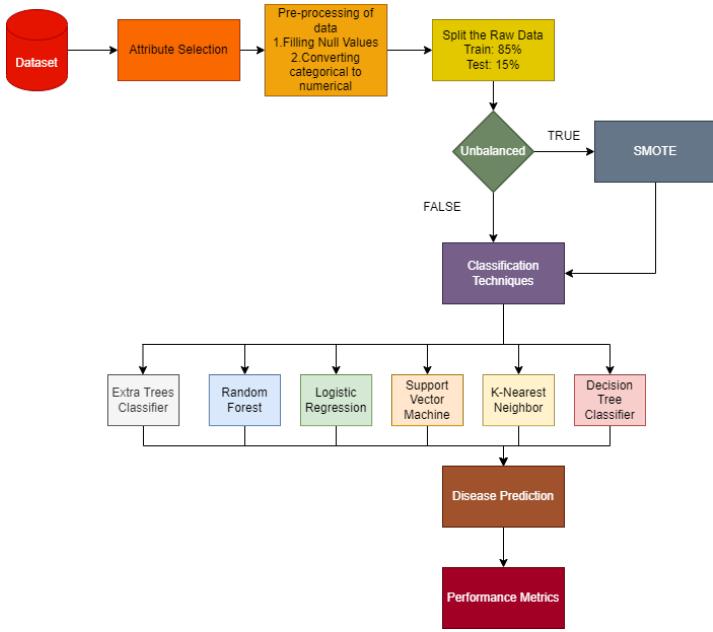


Fig. 7. Summary of the supervised machine learning part of this work.

B. Algorithms

- Logistic Regression: It is a classification type of algorithm and working on probability, and it can predict the dependent variable, which is categorical. [23] Logistic regression follows a technique that predicts 0 or 1, yes or no, and success or failure. It uses a link function that works with a logistic function, and it is like an S shape, and in this threshold, the technique is used if the value is below the threshold, then it will tend to 0, and above the point, it will tend 1.

- Random Forest: Random Forest is a ML technique that uses classification and regression to make decisions. In this algorithm, these two techniques are used to solve the difficulty in a dataset and improve the performance of the function. It is very familiar with the decision tree, and each attribute makes the decision tree. This algorithm uses the majority of the predictions from each tree. [23].

- SVM: It is a kind of algorithm that works on classification problems in machine learning. It works well on practical issues and can solve linear or nonlinear problems. This algorithm plots a hyperplane for each attribute, and quality is represented as a coordinate in the dataset. Hyperplane divides the class from one to another, and the main reason is that we will set the latest data to the best category in the future [23].

- KNN: It is a kind of algorithm that works with supervised learning techniques and new and old data. Put the new data with their similar available categories. We use this in classification rather than regression. It follows the Euclidean distance techniques between data points and to find the nearest neighbors. It can be beneficial if the training data is significant. [23]

- Decision tree: It is a vital classification method in machine learning approaches. It can be applied and utilized in various fields, including finding missing data in a class, improving

search engines, and medical applications. The classical decision tree algorithms ID3, C4.5, and C5.0 have the advantages of high classifying speed and strong learning ability. [23]

- Extra trees classifier: It is a feature selection strategy based on ensemble learning. It's similar to a random forest classifier, but the decision trees are built differently. It can contain randomization while still optimizing, and this technique utilizes averaging to increase predictive accuracy and avoid overfitting problems. [23] This algorithm has various advantages, including a short execution time and a low computational cost.

C. Attention Based CNN model

CNN has contributed to the classification of photos, particularly medical ones. This has opened up new doors and made disease detection a lot easier. It also has greater accuracy in detecting new Coronaviruses. The suggested COVID-19 detection model is a custom CNN model based on channel and spatial attention, with ten convolutional layers, four max-pooling layers, and one layer which is fully connected. Fig. 9 depicts the proposed model's general block diagram and architecture. This model was trained using 1345 normal, COVID-19, and viral pneumonia pictures. In this model, its input layer takes the dataset images with the size of 256x256. The first layer is a 2D convolutional layer. We use 3x3 kernels and an Exponential Linear Unit (ELU) activation function in this layer. Then, standardize the inputs using batch normalization. The second layer is identical to the previous layer. After the second layer, the spatial connection of visual information in the COVID-19 Xray pictures is captured using channel and spatial attention modules. The subsequent subsection will go through these attention modules. After every two convolutional layers, these attention modules are applied. The following eight layers are also the same as the first layer and second layer, but these layers have a 2x2 max-pooling layer to prevent overfitting. After the last channel and spatial attention, Instead of completely linked layers, we employ the global average pooling layer since it determines the average value for each feature map, reducing overfitting. Then a dropout layer. Dropout is set to 0.5. Finally, there is a dense layer. This layer receives the input data and identifies the photos based on their class. Since our dataset includes three classes, we apply Softmax to recognize them.

D. Attention module

In our model, The spatial connection of visual hints in COVID-19 X-ray pictures is captured using this module. We use the channel attention, and spatial attention concept suggested by Woo et al [24]. Fig. 8 shows a schematic of these attention modules. Firstly channel attention takes a normalized convolution layers tensors. The outputs of the channel attention module are then passed to the spatial attention module, which forwards them to a convolution layer. After every two convolution layers, this approach is used.

- 1) Channel attention module: First, we conduct max and average pooling on the input tensor from a normalised convolution layer in the channel attention module. Then send

max pooling and average pooling to a shared multi-layer perceptron network (MLP). The resultant feature vectors are then combined using element-wise summation. In conclusion, the following formula is used to determine channel attention:

$$M_c(F) = \sigma(W_1(W_0(F_{avg}^c)) + W_1(W_0(F_{max}^c))), \quad (1)$$

2) *Spatial attention module*: The input tensor from the channel attention module is additionally subjected to max pooling and average pooling in the spatial attention module. Then, using the Sigmoid function (σ), these two resulting tensors are concatenated to construct a convolution of filter size (f) of 7×7 . The output tensor concatenated $M_s(F)$ is specified as:

$$M_s(F) = \sigma(f^{7 \times 7}([F_{avg}^c; F_{max}^c])), \quad (2)$$

E. Hyperparameter Tuning

- **Hyperparameter**: A hyperparameter is a condition that is made before the start of the learning process. Before the learning algorithm starts training the model, the ML user selects and sets the value of the Hyperparameter. These elements are adjustable and directly impact how successfully model trains. These are used to define the model's learning capability and complexity. It has benefits like Searching the space of possible hyperparameters efficiently, and It is simple to manage a large number of experiments for hyperparameter tuning. [25] Examples include hyperparameter number of epochs, number of branches in a decision tree, and learning rate.

F. Performance Metrics

- **Accuracy**: The most straightforward intuitive performance metric is accuracy. Accuracy can tell us if a model is being trained appropriately and how well it will perform. However, it does not provide extensive information about how it applies to the problem as it becomes misleading while dealing with imbalanced datasets. It follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

In this paper [26], the researchers discovered that a model with higher accuracy dramatically enhances people's trust in the model.

- **Precision**: Precision evaluates the performance of a machine learning model — the accuracy of the model's optimistic forecast. Precision is a metric that measures how many correct optimistic forecasts have been made. As a result, precision evaluates the minority class's accuracy. [26]

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

- **Recall**: A machine learning model's recall is dependent on positive samples but independent of negative samples. It allows us to determine how many positive samples the ML model correctly classified. The recall is concerned with accurately

characterizing all positive samples. It makes no distinction between positive and negative samples. [26]

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

- **F1-score**: We cannot solely rely on accuracy as the performance metric to justify the classifier's performance. In most cases, the F1-score is better than the accuracy. Both false positives and false negatives are taken into consideration, giving a more precise explanation of the classifier's performance even if the dataset does not have an even class distribution. [26]

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

G. Over Sampling Technique

- **SMOTE**: Unbalanced data sets are common in practice, and most machine learning algorithms are vulnerable to them. Thus we need to improve their performance using approaches like SMOTE. [25] SMOTE has the advantage of not duplicating data points instead of creating synthetic data points that deviate slightly from the original data points. SMOTE is an excellent oversampling method. The SMOTE function has different parameters and depending on the values of these parameters, and the classifiers produced different accuracy. These parameters were tweaked to find the optimum accuracy for achieving the ideal SMOTE ratio and k neighbors value, the number of nearest neighbours utilised in the creation of synthetic samples.

H. Explainable AI

- **Lime**: Lime is a technique that can explain any ML model's prediction. Learning an interpretable model locally around the forecast can improve the model's interpretability and, more importantly, explain each prediction. Feature values in a single data sampling is altered before assessing the effect on the outcome. [25]

- **Grad-CAM**: Selvaraju et al. invented Gradient-weighted Class Activation Mapping, or "Grad-CAM," to display to programmers how their model looks in an image and assist them to discover the greatest accuracy. [27]. It demonstrates which element of the input picture the network is aiming for and how the network ends at its desired outcome. For each class designation, Grad-CAM creates a "heatmap visualization." We can use this "heatmap" to confirm where the CNN is gazing in the image visually.

III. RESULTS AND DISCUSSION

The following section analyzes the proposed system's outputs and accuracy, presenting the findings. Most of the time, the algorithm with the highest accuracy produces the most precise results.

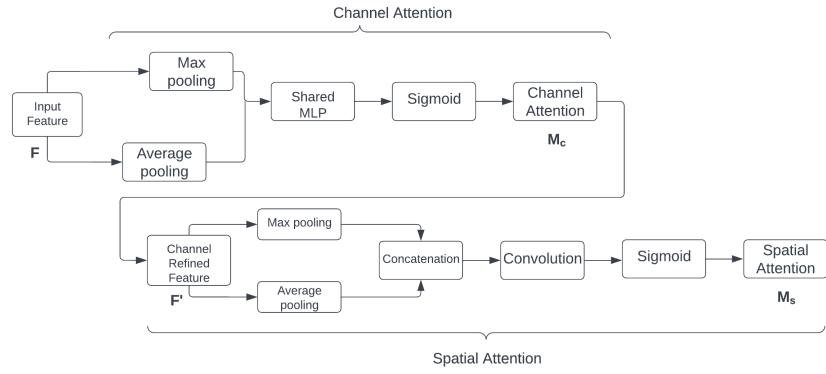


Fig. 8. Block diagram of the Attention module

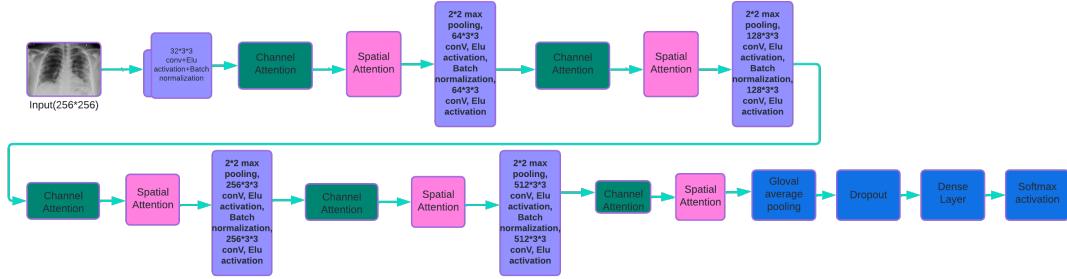


Fig. 9. The proposed deep learning model (Attention-based CNN) for COVID-19 Chest X-ray image classification is depicted in a block diagram.

TABLE V
COMPARISON OF ALGORITHM PERFORMANCE FOR DIABETES DISEASE

Classifier	Sampling technique	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	Without Smote	0.76	0.79	0.70	0.71	0.845
	With Smote	0.76	0.76	0.70	0.71	0.847
Random Forest	Without Smote	0.80	0.80	0.76	0.77	0.855
	With Smote	0.85	0.84	0.84	0.84	0.868
Extra Trees	Without Smote	0.77	0.77	0.73	0.74	0.853
	With Smote	0.84	0.84	0.83	0.84	0.864
SVM	Without Smote	0.74	0.76	0.68	0.69	0.817
	With Smote	0.80	0.79	0.81	0.79	0.850
KNN	Without Smote	0.74	0.77	0.67	0.68	0.838
	With Smote	0.74	0.77	0.67	0.68	0.825
Decision Tree	Without Smote	0.68	0.66	0.64	0.64	0.637
	With Smote	0.80	0.81	0.77	0.78	0.763

TABLE VI
COMPARISON OF ALGORITHM PERFORMANCE FOR LIVER DISEASE

Classifier	Sampling technique	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	Without Smote	0.71	0.62	0.55	0.53	0.747
	With Smote	0.71	0.44	0.48	0.43	0.748
Random Forest	Without Smote	0.68	0.59	0.56	0.57	0.690
	With Smote	0.75	0.71	0.67	0.69	0.763
Extra Trees	Without Smote	0.72	0.65	0.61	0.62	0.733
	With Smote	0.74	0.70	0.68	0.68	0.748
SVM	Without Smote	0.71	0.36	0.50	0.42	0.715
	With Smote	0.71	0.35	0.50	0.41	0.725
KNN	Without Smote	0.70	0.48	0.50	0.43	0.690
	With Smote	0.70	0.60	0.51	0.44	0.690
Decision Tree	Without Smote	0.65	0.59	0.59	0.59	0.592
	With Smote	0.70	0.66	0.66	0.66	0.649

TABLE VII
COMPARISON OF ALGORITHM PERFORMANCE FOR HEART DISEASE

Classifier	Sampling technique	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	Without Smote	0.64	0.64	0.64	0.64	0.696
Random Forest	Without Smote	0.72	0.72	0.72	0.72	0.778
Extra Trees	Without Smote	0.71	0.71	0.71	0.71	0.762
SVM	Without Smote	0.64	0.64	0.63	0.63	0.696
KNN	Without Smote	0.63	0.63	0.63	0.63	0.692
Decision Tree	Without Smote	0.69	0.69	0.69	0.69	0.692

In Table V,VI,VII,VIII For Diabetes, Liver, Heart, and Kidney Diseases, we have demonstrated the various performance measures compared side by side. For SMOTE and without SMOTE, recall, F1 score, accuracy, precision, and area under the ROC curve figures are displayed. For Diabetes prediction, Random Forest surpasses the other classifiers. Without SMOTE, we were able to reach an accuracy of 80%, and after

using SMOTE, accuracy increased to 85%. Although the Extra Trees classifier achieved an accuracy of 72% before using SMOTE after SMOTE, the Random Forest classifier increased to 75%, the highest among the classifiers. As the heart disease dataset was balanced, there was no need to use SMOTE in this scenario. The highest accuracy was given by the Random Forest Classifier, which was 72%. We also see the same pattern for the kidney dataset as the dataset was balanced. With a 97 % accuracy, the Random Forest Classifier once again surpassed the other classifiers.

TABLE VIII
COMPARISON OF ALGORITHM PERFORMANCE FOR KIDNEY DISEASE

Classifier	Sampling technique	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	Without Smote	0.91	0.91	0.93	0.91	0.981
Random Forest	Without Smote	0.97	0.97	0.98	0.97	0.993
Extra Trees	Without Smote	0.95	0.95	0.95	0.95	0.993
SVM	Without Smote	0.89	0.88	0.90	0.89	0.977
KNN	Without Smote	0.93	0.92	0.93	0.92	0.959
Decision Tree	Without Smote	0.90	0.90	0.89	0.90	0.892

The ROC Curve was used to summarize a classifier's ability to distinguish across classes. The ROCs of Logistic regression, Random forest, Extra trees, Decision tree, SVM, and KNN algorithm for diabetes, liver, heart, and kidney disease are presented in Fig. 10, Fig. 11, Fig. 12, and Fig. 13, respectively. From the ROC curve, we can clearly see why our proposed model is more competent in making predictions than other classifiers. In Table X, we analyze the performance of our suggested model with that of other models for diabetes, liver disease, heart disease, and kidney disease prediction. Our proposed model (Random forest) performance is better than other recent proposed models.

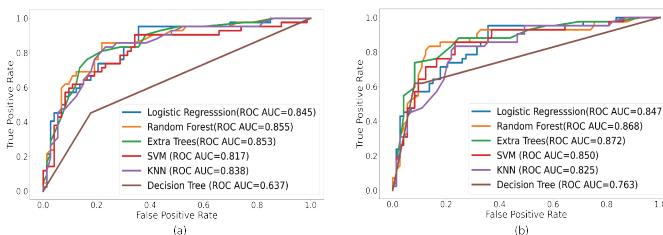


Fig. 10. ROC curves of six algorithms for diabetes disease. (a) Without Smote and (b) With Smote.

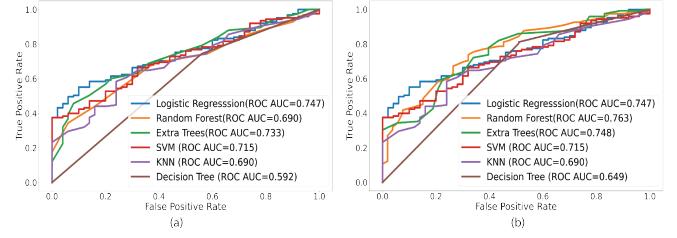


Fig. 11. ROC curves of six algorithms for liver disease. (a) Without Smote and (b) With Smote.

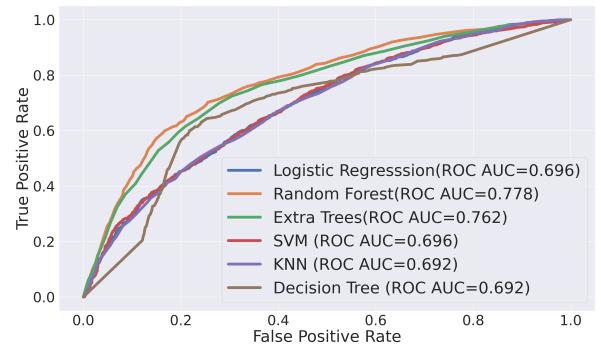


Fig. 12. ROC curves of six algorithms for heart disease without smote.

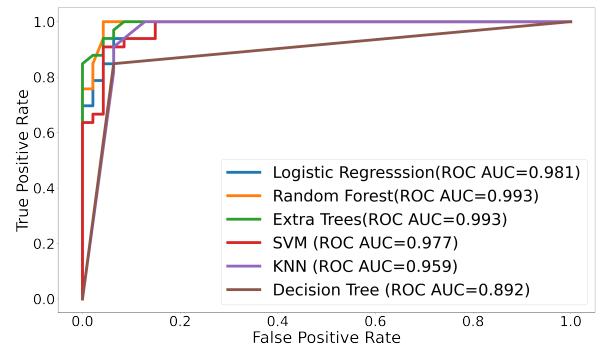


Fig. 13. ROC curves of six algorithms for kidney disease without smote.

For COVID-19 detection, the proposed model is an attention-based CNN model , which has convolutional layers. This model was trained over the course of 18 epochs. The models are trained in batches of 32. All of the training photos have been scaled to a resolution of 256x256 pixels. Our proposed model's training and validation accuracy and loss with corresponding epochs are displayed in Fig. 14.

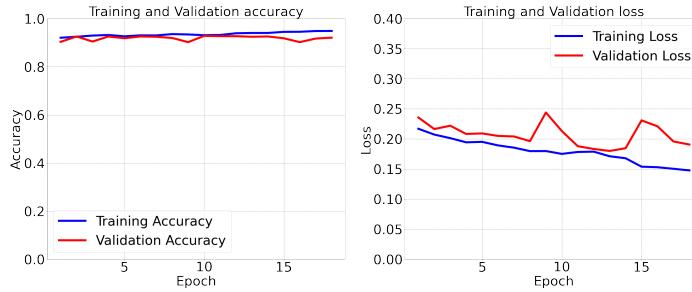


Fig. 14. The training and validation accuracy and corresponding epochs for the attention-based CNN model.

Fig. 14 shows, that validation accuracy for our models is 92 percent, with a loss of 0.1906. The training accuracy rate is 94.86 percent, with a loss rate of 0.1476 percent. In Table IX, we can notice the class-wise F1-score, recall, and precision of the COVID-19 dataset. In each class, the F1 score, recall, and precision are greater than or equal to 90%. For viral pneumonia class prediction, it presents the highest F1 score, recall, and precision. In Table XI, in comparison to other pre-trained models, our proposed model is smaller. However, It gives better accuracy, F1 score, recall, and precision than other proposed classifiers. COVID-19 detection performance reported in recent literature is equivalent to the performance obtained from this research.

TABLE IX

AVERAGE PRECISION, RECALL, AND F1-SCORE WERE USED TO DO A CLASSWISE ANALYSIS OF THE COVID-19 DATASET

Class	Precision	Recall	F1-score
COVID	0.91	0.91	0.91
Normal	0.90	0.90	0.90
Viral Pneumonia	0.97	0.97	0.97

To provide the explainability of the model prediction, we used explainable AI Lime. In Section II, we explain how this explainable AI model works. We integrated the Random Forest model into these explainable models as it was our best model.

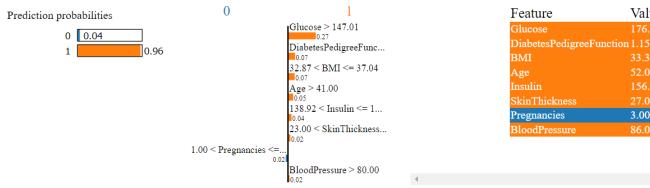


Fig. 15. Explanation of the prediction for diabetes disease using LIME.

For diabetes disease, we chose a random case from our test data to see the explanation of our model. Fig. 15 depicts an explanation for a patient affected by diabetes with summarizes the patient's medical history and symptoms and the result. The probability of being diagnosed with diabetes in this situation is 0.96. The orange bars represent the medical history and symptoms with much weight in favor of the prognosis, whereas

the blue bars represent data that contradicts it. According to the explanation, glucose, diabetes pedigree function, BMI, age, insulin, and skin thickness are the patient's significant symptoms and medical records that most contribute to the prediction.

Fig. 16 describes the explanation of a non-Liver disease-affected patient. It was based on their diagnosis values.



Fig. 16. Explanation of the prediction for liver disease using LIME.

Since we chose a random test case for the explanation, now we can observe that the test case was of a negative patient with 87 percent confidence. As the "blue" bar represents the vital feature, and the "orange" bar for less critical, we can see there are two features with orange marked and the rest of them are blue marked. So, this can be clear that the "Random Forest" model considers the eight features more important than other features. As a result, "Lime" discovered that direct Bilirubin, age, total bilirubin, direct bilirubin, alkaline_phosphatase, alanine_aminotransferase, aspartate_aminotransferase, and total_proteins all had a role in the progression of liver illness. And the other two were less critical.

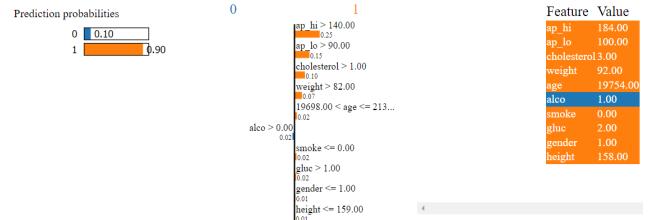


Fig. 17. Explanation of the prediction for heart disease using LIME.

Our Model predicts heart disease in this patient with 90 percent confidence and 10 percent confident that the patient has heart disease and justifies the prediction by stating that the ap_hi level is more significant than 140, ap_lo is greater than 90, cholesterol is more important than 1.0, and so on that, we can relate from the Fig. 17. On the right, we can see the worth of the genuine feature for the patient.

Fig. 18 describes the explanation of a Kidney disease-affected patient. We did consider the test case randomly from our dataset. After applying the "Lime" algorithm, Our Random Forest model described the test case as a positive patient with 97 percent confidence, which we did not know before. Justified

TABLE X
COMPARISON OF THE PROPOSED SYSTEM'S PERFORMANCE FOR DIABETES, LIVER DISEASE, HEART DISEASE, AND KIDNEY DISEASE PREDICTION WITH EXISTING WORKS

Disease	Reference	Classifier	Dataset	Accuracy	Other Metrics
Diabetes	[28]	Decision tree	Egyptian diabetes patients	84%	N/A
	[29]	Random forest	Pima Indian	77.21%	Sensitivity:74.58% Specificity:79.58%
	[30]	Random forest	Pima Indian	71.428%	Precision:64.102% Recall:45.454% F1-Score:53.191%
	[11]	Random forest	Pima Indian	74.03%	N/A
	This work	Random forest	Pima Indian	85%	Precision:84% Recall:84% F1-Score:84% ROC AUC:0.868
Liver	[31]	PSO	Egyptian NCCVH database	66.4%	N/A
	[32]	Naïve Bayes and FT Tree	WEKA dataset	Naïve Bayes:75.54% FT Tree:72.6624%	N/A
	[30]	SVM & Backpropagation	(UCI) Machine Learning Repository	SVM :71% Backpropagation:73.2%	N/A
	[33]	Modified Rotation Forest	UCI liver dataset and Indian dataset	74.78%	N/A
	This work	Random forest	(UCI) Machine Learning Repository	75%	Precision:71% Recall:67% F1-Score:69% ROC AUC:0.763
Heart	[34]	Random forest	(UCI) Machine Learning Repository	84.1604%	ROC AUC:0.9018
	[35]	Decision tree	Hungarian heart disease	67.7%	N/A
	[36]	Random forest	SCleveland Heart Disease dataset	85.81%	N/A
	[11]	Random forest	Framingham heart study	83.85%	N/A
	This work	Random forest	(UCI) Machine Learning Repository	85%	Precision:72% Recall:72% F1-Score:72% ROC AUC:0.778
Kidney	[37]	RBF	(UCI) Machine Learning Repository	95.84%	Precision:90% Recall:91.14% F-measure:0.9057
	[38]	Naive Bayes	(UCI) Machine Learning Repository	97%	Precision:97% Recall:97% F1-Score:97% ROC AUC:0.99
	[39]	Decision Tree	(UCI) Machine Learning Repository	93%	Precision:92.5% Recall:93% Fmeasure:92.7%
	[40]	Random Forest	(UCI) Machine Learning Repository	94.16%	Precision:95.12% Recall:96.29%
	This work	Random forest	(UCI) Machine Learning Repository	97%	Precision:97% Recall:98% F1-Score:97% ROC AUC:0.993

the prediction by stating that the hemoglobin is greater than 12.70 and less than or equal to 14.35, hypertension(htn) is less than or equal to 0, diabetes mellitus(dm) is less than or equal to 0, red blood cell count (RC) is greater than four and less than or equal to 5, white blood cell count (WC) is greater 6525 and less than or equal 79, blood pressure (bp) is greater than 70, and less than or equivalent 80, Anemia (ane) is less than or equal 0, and age is in between 52 to 61. And blood glucose random (bgr) and appetite class were considered less important.

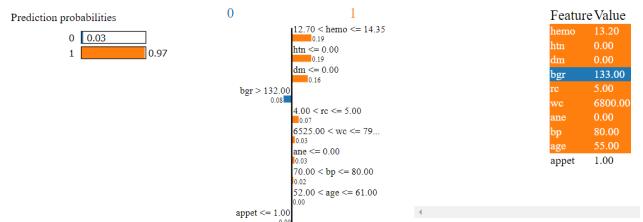


Fig. 18. Explanation of the prediction for kidney disease using LIME.

Fig. 19 is a visual representation of how our model takes some portion of an image as more important than other areas. We see, in terms of positive X-ray, Grad-CAM shows that the model concentrate on a specific area. In contrast, it does not offer the same.

TABLE XI
COMPARISON OF THE PROPOSED SYSTEM'S PERFORMANCE FOR COVID-19 PREDICTION WITH EXISTING WORKS

Reference	Techniques	Dataset	Validation Accuracy	Other Metrics
[41]	CNN (ResNet101)	5982 chest X-ray images (1765 COVID)	71.9%	Sensitivity: 77.3%, Specificity: 71.8%
[42]	ResNet50	15478 chest X-ray images (473 COVID)	93%	Sensitivity:90.1% Specificity:89.6%
[43]	DeTraC	1768 chest X-ray images (949 COVID)	93.1%	N/A
[44]	VGG-16 with both attention and convolution module	1125 chest X-ray images (At least 125 COVID)	79.58%	Precision:84% Recall:78% F1-Score:80%
This work	Attention-based CNN model	4035 chest X-ray images(1345 COVID)	92%	Precision:92% Recall:92% F1-Score:92%

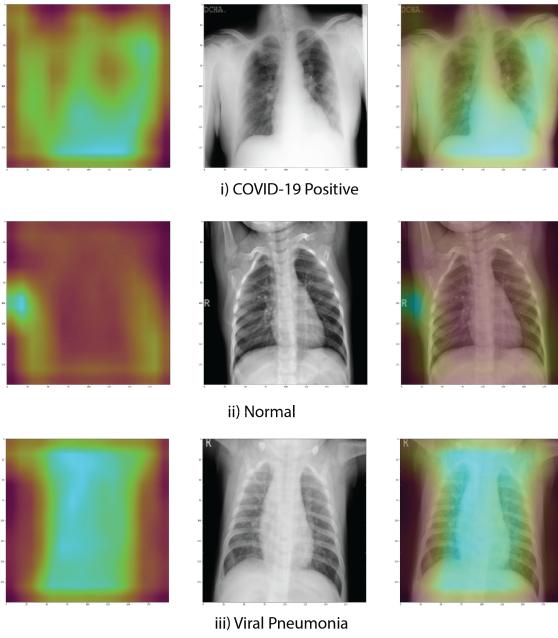


Fig. 19. Results of Grad-CAM visualization.

The user interface of our web application has been created in such a way that the user can utilize it efficiently. The user interface design for the multi-disease prediction system is shown below.

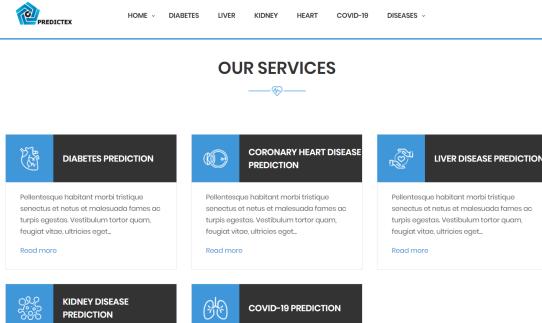


Fig. 20. The system's front-end view.

Fig. 20 shows the home page of our system, and users have an interface consisting of multiple disease choices. On Home-Screen, there are Five diseases users can choose from. They are Diabetes, Kidney Disease, Liver Disease, Coronary Heart Disease, and COVID-19 prediction.

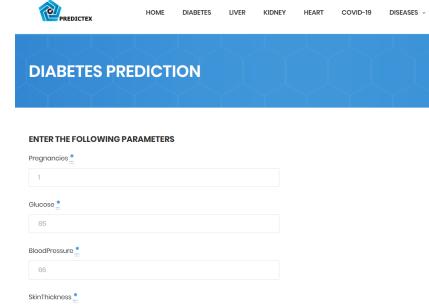


Fig. 21. Input windows for textual data.

Fig. 21 shows that this interface will appear on the home page after selecting a particular disease. For example, you choose Diabetes Prediction, and this page expects various parameters based on the diseases. For diabetes, it expects "Pregnancies," "Glucose," "Blood-Pressure, "Skin-Thickness" values, etc.

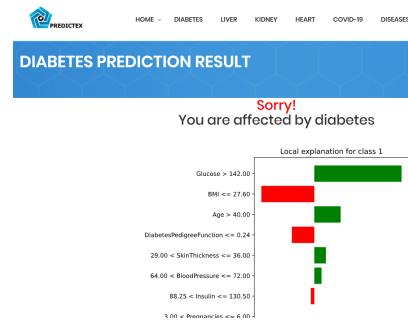


Fig. 22. Output screen of the diseases.

Fig. 22 shows, after submitting the expected values, our algorithm processes the values and gives an output whether

the user is infected with the disease or not as. If the user is infected, the system provides a result with a message: "Sorry! You are affected by Diabetes". It will also show an explanation of prediction.

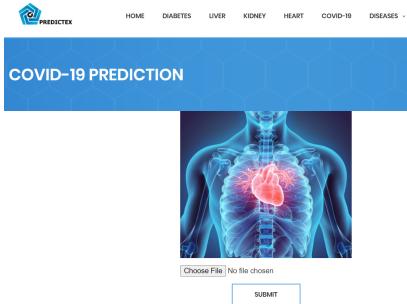


Fig. 23. User upload page for COVID-19 case.

Users upload and submit a chest X-ray image for COVID-19, as shown in Fig. 23.

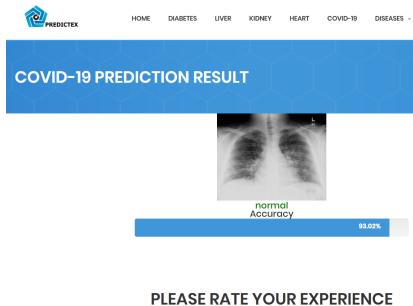


Fig. 24. Shows prediction result of COVID-19 detection.

Fig. 24 shows that the imaging system gives the prediction result with an accuracy output after processing. Fig. 26 shows the designed Android application user interface. The proposed Android application work through API. When a user uses our Android application to request a disease prediction, the request is sent to our central server, which computes the result and returns it as a response. The Android application receives the response and then displays the user's result. **In the form of a survey, we asked participants to rate our Android application.** A total of 24 participants took part in the survey, of which four persons were female, and twenty persons were male. Four of the participants were between the ages of 18 and 20. Sixteen of the participants were between the ages of 21 and 24. Finally, ages of the last four individuals were between 25 and 26. We requested the focus group rate the Android application based on five criteria on a scale from 0 to 5. Criteria included productivity of the application, consistency of the results, the engagement level of the application, design of the application, and simplicity of the interface. The findings of the survey are shown in Fig. 25.

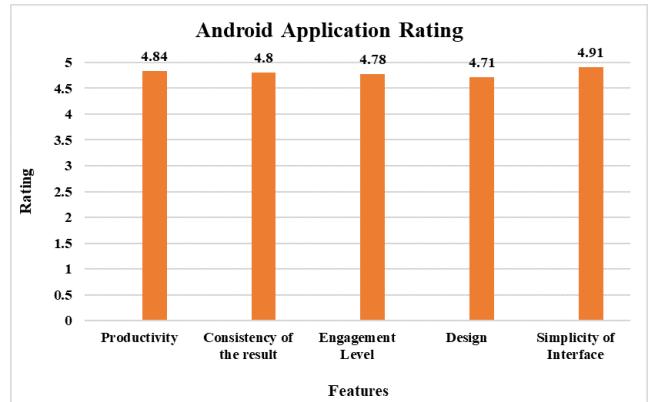


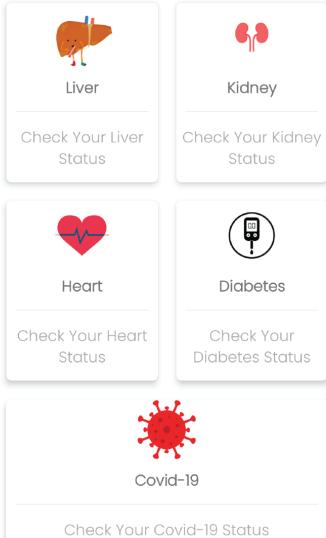
Fig. 25. Android application features review survey results.

IV. CONCLUSION

Different supervised ML algorithms and deep learning algorithms have been carefully analyzed on the dataset of several diseases and proposed that Random Forest Classifier and the attention-based CNN model offer the most defined prediction of these diseases. We have created a website and an Android application based on these two algorithms, which will predict whether a person is affected by these diseases or not in real-time. We have used different performance metrics such as the value of the ROC AUC, accuracy, precision, f1-score, and recall to show that our proposed models produce the most accurate predictions. Explainable AI models have provided interpretability and clarity to the predictions. The main symptoms can now be found and treated with its help. This website and Android application can help people predict unsophisticated diseases and help them detect these diseases and get screening from anywhere. We plan to improve the existing diseases dataset and add additional diseases to the lineup in the future, utilizing a more relevant dataset tailored to the Bangladeshi demographic.

REFERENCES

- [1] "Home — who.int." <https://www.who.int/>. [Accessed 08-May-2022].
- [2] T. Biswas, M. S. Islam, N. Linton, and L. B. Rawal, "Socio-economic inequality of chronic non-communicable diseases in bangladesh," *PloS one*, vol. 11, no. 11, p. e0167140, 2016.
- [3] "Over 35,000 develop kidney failure in Bangladesh every year — thedailystar.net." <https://www.thedailystar.net/city/news/18m-kidney-patients-bangladesh-every-year-1703665>. [Accessed 08-May-2022].
- [4] S. Alam, M. M. Hossain, G. Azam, G. Mustafa, M. Alam, A. Sayeed, M. Khan, and N. Ahmad, "Nonalcoholic fatty liver disease: A new frontier for hepatology in bangladesh and a call for action to combat," *Journal of Bangladesh College of Physicians and Surgeons*, vol. 35, no. 4, pp. 184–191, 2017.
- [5] "Bangladesh: WHO Coronavirus Disease (COVID-19) Dashboard With Vaccination Data — covid19.who.int." <https://covid19.who.int/region/searo/country/bd>. [Accessed 08-May-2022].
- [6] N. Kumar, N. Narayan Das, D. Gupta, K. Gupta, and J. Bindra, "Efficient automated disease diagnosis using machine learning models," *Journal of Healthcare Engineering*, vol. 2021, 2021.
- [7] T. Wang, Y. Tian, and R. G. Qiu, "Long short-term memory recurrent neural networks for multiple diseases risk prediction by leveraging longitudinal medical records," *IEEE journal of biomedical and health informatics*, vol. 24, no. 8, pp. 2337–2346, 2019.



KIDNEY DISEASE PREDICTION

ENTER THE FOLLOWING PARAMETERS

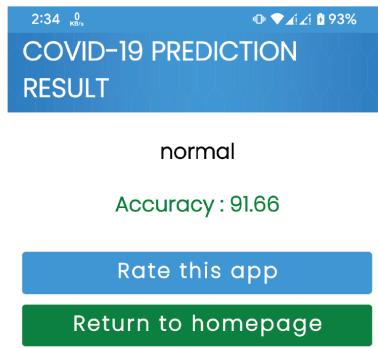
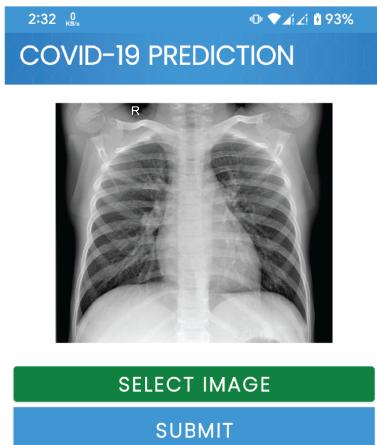
RBC count:	<input type="text" value="Enter RBC count"/>
WBC count:	<input type="text" value="Enter WBC count"/>
Appetite:	<input checked="" type="radio"/> Yes <input type="radio"/> No
Pus Cell:	<input checked="" type="radio"/> Normal <input type="radio"/> Not Normal
Hypertension:	<input checked="" type="radio"/> Yes <input type="radio"/> No
Hemoglobin	<input type="text" value="Enter value"/>
Blood Glucose	<input type="text" value="Enter value"/>
Diabetes Mellitus:	<input checked="" type="radio"/> Yes <input type="radio"/> No
Anemia:	<input checked="" type="radio"/> Yes <input type="radio"/> No

SUBMIT

i) Home page

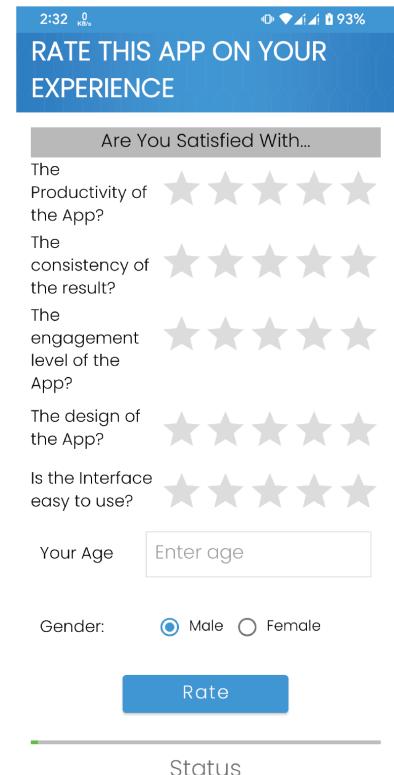
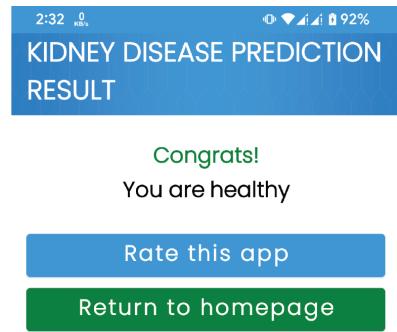
ii) Input diagnosed values

iii) Result



iv) Input Chest X-ray image

v) COVID-19 prediction result



vi) Rating page

Fig. 26. Android application user interface.

- [8] A. Yaganteeswarudu, "Multi disease prediction model by using machine learning and flask api," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 1242–1246, IEEE, 2020.
- [9] L. Men, N. Ilk, X. Tang, and Y. Liu, "Multi-disease prediction using lstm recurrent neural networks," *Expert Systems with Applications*, vol. 177, p. 114905, 2021.
- [10] A. Ampavathi and T. V. Saradhi, "Multi disease-prediction framework using hybrid deep learning: an optimal prediction model," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 24, no. 10, pp. 1146–1168, 2021.
- [11] V. Jackins, S. Vimal, M. Kaliappan, and M. Y. Lee, "AI-based smart prediction of clinical disease using random forest classifier and naive bayes," *The Journal of Supercomputing*, vol. 77, pp. 5198–5219, 2021.
- [12] K. G. Dinesh, K. Arumugaraj, K. D. Santhosh, and V. Mareeswari, "Prediction of cardiovascular disease using machine learning algorithms," in *International Conference on Current Trends towards Converging Technologies (ICCTCT)*, pp. 1–7, 2018.
- [13] K. F. Haque, F. F. Haque, L. Gandy, and A. Abdalgawad, "Automatic detection of COVID-19 from chest X-ray images with convolutional neural networks," in *International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, pp. 125–130, 2020.
- [14] K. Arumugam, M. Naved, P. P. Shinde, O. Leiva-Chauca, A. Huaman-Osorio, and T. Gonzales-Yanac, "Multiple disease prediction using machine learning algorithms," *Materials Today: Proceedings*, 2021.
- [15] K. Harimoorthy and M. Thangavelu, "Multi-disease prediction model using improved svm-radial bias technique in healthcare monitoring system," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3, pp. 3715–3723, 2021.
- [16] I. Mohit, K. S. Kumar, U. A. K. Reddy, and B. S. Kumar, "An approach to detect multiple diseases using machine learning algorithm," in *Journal of Physics: Conference Series*, vol. 2089, p. 012009, IOP Publishing, 2021.
- [17] A. K. Dubey, "Optimized hybrid learning for multi disease prediction enabled by lion with butterfly optimization algorithm," *Sādhanā*, vol. 46, no. 2, pp. 1–27, 2021.
- [18] G. M. Ifraz, M. H. Rashid, T. Tazin, S. Bourouis, and M. M. Khan, "Comparative analysis for prediction of kidney disease using intelligent machine learning methods," *Computational and Mathematical Methods in Medicine*, vol. 2021, 2021.
- [19] J. W. Smith, J. E. Everhart, W. Dickson, W. C. Knowler, and R. S. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus," in *Proceedings of the annual symposium on computer application in medical care*, p. 261, American Medical Informatics Association, 1988.
- [20] M. Padmanabhan, P. Yuan, G. Chada, and H. V. Nguyen, "Physician-friendly machine learning: A case study with cardiovascular disease risk prediction," *Journal of clinical medicine*, vol. 8, no. 7, p. 1050, 2019.
- [21] B. V. Ramana, M. S. P. Babu, and N. Venkateswarlu, "A critical comparative study of liver patients from usa and india: an exploratory analysis," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 3, p. 506, 2012.
- [22] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. A. Kashem, M. T. Islam, S. Al Maadeed, S. M. Zughraier, M. S. Khan, et al., "Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images," *Computers in biology and medicine*, vol. 132, p. 104319, 2021.
- [23] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR). [Internet]*, vol. 9, pp. 381–386, 2020.
- [24] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.
- [25] A. Patil, A. Framewala, and F. Kazi, "Explainability of smote based oversampling for imbalanced dataset problems," in *2020 3rd International Conference on Information and Computer Technologies (ICICT)*, pp. 41–45, IEEE, 2020.
- [26] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *European conference on information retrieval*, pp. 345–359, Springer, 2005.
- [27] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *International Conference on Computer Vision*, pp. 618–626, 2017.
- [28] S. Sahoo, T. Mitra, A. K. Mohanty, B. J. R. Sahoo, and S. Rath, "Diabetes prediction: A study of various classification based data mining techniques," *International Journal of Computer Science and Informatics*, 2022.
- [29] Q. Zou, K. Qu, Y. Luo, D. Yin, Y. Ju, and H. Tang, "Predicting diabetes mellitus with machine learning techniques," *Frontiers in Genetics*, vol. 9, 2018.
- [30] K. Orabi, Y. Kamal, and T. Rabah, "Early predictive system for diabetes mellitus disease," vol. 9728, pp. 420–427, 2016.
- [31] S. Hashem, G. Esmat, W. Elkale, S. Habashy, S. A. Raouf, M. Elhefnawi, M. I. Eladawy, and M. Elhefnawi, "Comparison of machine learning approaches for prediction of advanced liver fibrosis in chronic hepatitis c patients," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, pp. 861–868, 2017.
- [32] S. Dhamodharan, "Liver disease prediction using bayesian classification," 2016.
- [33] B. V. Ramana, M. P. Babu, and N. Venkateswarlu, "Liver classification using modified rotation forest," *International Journal of Engineering Research and Development*, vol. 6, pp. 17–24, 2012.
- [34] K. Pahwa and R. Kumar, "Prediction of heart disease using hybrid technique for selecting features," in *International Conference on Electrical, Computer and Electronics (UPCON)*, pp. 500–504, 2017.
- [35] S. Ekiz and P. Erdogmus, "Comparative study of heart disease classification," in *Electric Electronics, Computer Science, Biomedical Engineering's Meeting (EBBT)*, pp. 1–4, 2017.
- [36] Y. K. Singh, N. Sinha, and S. K. Singh, "Heart disease prediction system using random forest," in *International Conference on Advances in Computing and Data Sciences*, pp. 613–623, Springer, 2016.
- [37] E.-H. A. Rady and A. S. Anwar, "Prediction of kidney disease stages using data mining algorithms," *Informatics in Medicine Unlocked*, vol. 15, pp. 1–7, 2019.
- [38] D. S. Sisodia and A. Verma, "Prediction performance of individual and ensemble learners for chronic kidney disease," in *International Conference on Inventive Computing and Informatics (ICICI)*, pp. 1027–1031, 2017.
- [39] R. Ani, G. Sasi, U. R. Sankar, and O. S. Deepa, "Decision support system for diagnosis and prediction of chronic renal failure using random subspace classification," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1287–1292, 2016.
- [40] R. Gupta, N. Koli, N. Mahor, and N. Tejashri, "Performance analysis of machine learning classifier for predicting chronic kidney disease," in *International Conference for Emerging Technology (IN CET)*, pp. 1–4, 2020.
- [41] M. Z. Che Azemin, R. Hassan, M. I. Mohd Tamrin, and M. A. Md Ali, "COVID-19 deep learning prediction model using publicly available radiologist-adjudicated chest X-ray images as training data: preliminary findings," *International Journal of Biomedical Imaging*, vol. 2020, 2020.
- [42] T.-C. Lin and H.-C. Lee, "Covid-19 chest radiography images analysis based on integration of image preprocess, guided grad-CAM, machine learning and risk management," in *International Conference on Medical and Health Informatics*, pp. 281–288, 2020.
- [43] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network," *Applied Intelligence*, vol. 51, pp. 854–864, 2021.
- [44] C. Sitaula and M. B. Hossain, "Attention-based VGG-16 model for COVID-19 chest X-ray image classification," *Applied Intelligence*, vol. 51, pp. 2850–2863, 2021.