ISSN 2277-2685
IJESR/July-Sep. 2024/ Vol-14/Issue-3/221-238
Reddy Sai Ravi Kumar *et. al.,* / International Journal of Engineering & Science Research

CARDIAC ABNORMALITIES DIAGNOSIS IN ELECTROCARDIOGRAMS USING CONVOLUTIONAL NEURAL NETWORK

[1] Reddy Sai Ravi Kumar, [2] Dr. K. Santhi Sree
[1] Student, Department of Information Technology, University College of Engineering, Science and Technology, JNTUH Hyderabad
[2] Professor, Department of Information Technology, University College of Engineering, Science and Technology, JNTUH Hyderabad

Abstract: Cardiovascular diseases, primarily heart conditions, are a prominent global cause of death, making early prediction vital. The cost-effective and noninvasive tool, Electrocardiogram (ECG), aids in detecting these diseases by monitoring heart activity. To enhance prediction accuracy, deep learning techniques are employed to identify four significant cardiac abnormalities: abnormal heartbeat, myocardial infarction, history of myocardial infarction, and normal cases. The project combines transfer learning from deep neural networks like SqueezeNet and AlexNet with a specialized CNN architecture. This approach is for extracting important features, improving predictions when integrated with traditional machine learning algorithms. The proposed model stands out by delivering exceptional performance, significantly advancing the prediction of medical conditions from images. It underscores the crucial role of artificial intelligence in revolutionizing healthcare practices. The integrated Xception model elevates feature extraction for cardiac abnormality detection from ECG images. Extracted features serve as inputs to machine learning models, enhancing their ability to discern intricate patterns and anomalies. This amalgamation of advanced feature extraction with robust machine learning algorithms contributes to the project's effectiveness in providing accurate diagnoses. The streamlined user interactions through Flask with SQLite underscore the system's practicality, offering secure signup, signin, and efficient testing for improved healthcare practices.

Index terms - *Cardiovascular, deep learning, electrocar diogram (ECG) images, feature extraction, machine learning, transfer learning.*

1. INTRODUCTION

According to the World Health Organization, cardio- vascular diseases (heart diseases) are the leading cause of death worldwide. They claim an estimated 17.9 million lives each year, accounting for 32% of all deaths worldwide. About 85% of all deaths from heart disease are due to heart attacks, also known as myocardial infarctions (MI) [1]. Many lives can be saved if an efficient diagnosis of cardiovascular disease is detected at an earlier stage [1]. Different techniques are used in the healthcare system to detect heart diseases, such as electrocardiogram (ECG), echocardiography (echo), cardiac magnetic resonance imaging, computed tomography, blood tests, etc. [2], [3]. The ECG is a common, inexpensive, and noninvasive tool for measuring the electrical activity of the heart [4]. It is used

to identify heart-related cardiovascular diseases [4], [5]. A highly skilled clinician can detect heart disease from the ECG waves. However, this manual process can lead to inaccurate results and is very time-consuming [5].
There is great potential to benefit from advances in artificial intelligence in healthcare to reduce medical errors. In particular, the use of machine learning and deep learning techniques for automatic prediction of heart diseases[3],[6]–[10]. Themachine learning methods require an expert entity for features extraction and selection to identify the appropriate features before applying the classification phase. Feature extraction is a process of reducing the number of features in a data set by transforming or projecting the data into a new lower-dimensional feature space preserving the relevant information of the input data [11], [12].
The concept of feature extraction is concerned with creating a new set of features (different from the input feature) that are a combination of original features into a lower-dimensional space that extract most, if not all, of the information in input data. The most well-known feature extraction method is a principal component analysis [13], [14]. However, feature selection is a process of removing irrelevant and redundant features (dimensions) from the data set in the training process of machine learning algorithms. Various methods can be used for feature selection, classified as unsupervised, which refers to the method that does not need the output label for feature selection, and supervised, which refers to the methods that use output label for feature selection. Under supervised feature selection, there are three methods: the filter method, the wrapper method, and the embedded method [11], [12].
Many machine learning methods have been used for predicting cardiovascular diseases. Soni et al. [15] compared several machine learning algorithms, such as decision tree (DT), Naïve Bayes (NB), K-nearest neighbors (K-NN), and neural network (NN) on UCI Cleveland heart disease dataset. They concluded that DT had the highest accuracy of 89%. Dissanayake and Md Johar [16] studied the effect of the feature selection process on machine learning classifiers for predicting heart diseases from the UCI Cleveland heart disease dataset.

2. LITERATURE SURVEY

Cardio-Vascular Diseases (CVD) are found to be rampant in the populace leading to fatal death. The statistics of a recent survey reports that the mortality rate is expanding due to obesity, cholesterol, high blood pressure and usage of tobacco among the people. The severity of the disease is piling up due to the above factors. Studying about the variations of these factors and their impact on CVD is the demand of the hour. This necessitates the usage of modern techniques to identify the disease at its outset and to aid a markdown in the mortality rate. [3] Artificial Intelligence and Data Mining domains have a research scope with their enormous techniques that would assist in the prediction of the CVD priory and identify their behavioural patterns in the large volume of data. The results of these predictions will help the clinicians in decision making and early diagnosis, which would reduce the risk of patients becoming fatal. [6, 8, 24, 28] This paper compares and reports the various Classification, Data Mining, Machine Learning, Deep Learning models that are used for prediction of the Cardio-Vascular diseases. The survey is organized as threefold: Classification and Data Mining Techniques for CVD, Machine Learning Models for CVD and Deep Learning Models for CVD prediction. The performance metrics used for reporting the accuracy, the dataset used for prediction and classification, and the tools used for each category of these techniques are also compiled and reported in this survey.

The Electrocardiogram (ECG) is the P-QRS-T wave depicting the cardiac activity of the heart. The subtle changes in the electric potential patterns of repolarization and depolarization are indicative of the disease afflicting the patient. These clinical time domain features of the ECG waveform can be used in cardiac health diagnosis. Due to the presence of noise and minute morphological parameter values, it is very difficult to identify the ECG classes accurately by the naked eye. [5] Various computer aided cardiac diagnosis (CACD) systems, analysis methods, challenges addressed and the future of cardiovascular disease screening are reviewed in this paper. Methods developed for time domain, frequency transform domain, and time-frequency domain analysis, such as the wavelet transform, cannot by themselves represent the inherent distinguishing features accurately. [6, 9, 10, 23] Hence, nonlinear methods which can capture the small variations in the ECG signal and provide improved accuracy in the presence of noise are discussed in greater detail in this review. A CACD system exploiting these nonlinear features can help clinicians to diagnose cardiovascular disease more accurately.

Heart disease (HD) is a fatal disease which takes the lives of maximum people compared to other diseases across the world. Early and accurate detection of the disease will help to save many valuable lives. The HD can be detected from medical tests, Electrocardiogram (ECG) signal [6, 9], heart sounds, Computed Tomography (CT) Images etc. Out of all types of detection of HD from ECG signals plays a vital role. In this paper, the ECG samples of the subjects have been considered as the required inputs to the HD detection model. In recent past, many useful articles have been reported for classification of HD using different machine learning (ML) and deep learning (DL) models. It is observed that with imbalanced HD data the detection accuracy is lower. With an objective to achieve better detection of HD, suitable DL and ML models have been identified in this paper and the required classification models have been developed and tested [6]. The Generative Adversarial Network (GAN) model is chosen with an objective to deal with imbalanced data by generating and using additional fake data for detection purpose. Further, an ensemble model using long short-term memory (LSTM) and GAN is developed in this paper which demonstrates higher performance compared to individual DL model used in this paper [9]. The simulation results using standard MIT-BIH show that the proposed GAN-LSTM model provides the highest accuracy, F1-score and area under curve (AUC) of 0.992, 0.987 and 0.984 respectively compared to other models. Similarly, for PTB-ECG dataset the GAN-LSTM model outperforms all other models with accuracy, F1-score and AUC of 0.994, 0.993 and 0.995 respectively. It is observed that out of the five models investigated, the GAN model performs the best whereas the detection potentiality of the NB model is the lowest. Further research work can be carried out by choosing all other different ensemble models and using other different datasets and the performance can be similarly obtained and compared. The proposed best detection methodology can also be applied to other diseases and healthcare problems.

Artificial intelligence (AI) has given the electrocardiogram (ECG) and clinicians reading them super-human diagnostic abilities [3]. Trained without hard-coded rules by finding often subclinical patterns in huge datasets, AI transforms the ECG, a ubiquitous, non-invasive cardiac test that is integrated into practice workflows, into a screening tool and predictor of cardiac and non-cardiac diseases, often in asymptomatic individuals. This review [7] describes the mathematical background behind supervised AI algorithms, and discusses selected AI ECG cardiac screening algorithms including those for the detection of left ventricular dysfunction, episodic atrial fibrillation from a tracing recorded during normal sinus rhythm, and other structural and valvular diseases. The ability to learn from big data sets, without the need to understand the biological mechanism, has created opportunities for detecting non-cardiac diseases as COVID-19 and introduced challenges with regards to data privacy. [6, 10]Like all medical tests, the AI ECG must be carefully vetted and validated in real-world clinical environments. Finally, with mobile form factors that allow acquisition of medical-grade ECGs from smartphones and wearables, the use of AI may enable massive scalability to democratize healthcare.

An electrocardiogram (ECG) is an important diagnostic tool for the assessment of cardiac arrhythmias in clinical routine. In this study [8], a deep learning framework previously trained on a general image data set is transferred to carry out automatic ECG arrhythmia diagnostics by classifying patient ECG's into corresponding cardiac conditions. Transferred deep convolutional neural network (namely AlexNet) is used as a feature extractor and the extracted features are fed into a simple back propagation neural network to carry out the final classification. Three different conditions of ECG waveform are selected from MIT-BIH arrhythmia database to evaluate the proposed framework [23] Main focus of this study is to implement a simple, reliable and easily applicable deep learning technique for the classification of the selected three different cardiac conditions. Obtained results demonstrated that the transferred deep learning feature extractor cascaded with a conventional back propagation neural network were able to obtain very high performance rates. Highest obtained correct recognition rate is 98.51% while obtaining testing accuracy around 92%. Based on these results, transferred deep learning proved to be an efficient automatic cardiac arrhythmia detection method while eliminating the burden of training a deep convolutional neural network from scratch providing an easily applicable technique [21, 25, 26].

3. METHODOLOGY

i) Proposed Work:

The proposed model comprises two key phases: image processing and model building. In the image processing phase, we employ ImageDataGenerator for tasks such as rescaling, shear transformation, zooming, flipping, and reshaping to prepare the data. The model building phase utilizes deep learning models like [20, 22] Squeeze Net, AlexNet, and CNN for feature extraction, followed by the evaluation of these features using traditional machine learning algorithms such as Random Forest, SVM, KNN, Decision Tree, and Naive Bayes. This comprehensive approach aims to achieve high accuracy and reliability in cardiovascular disease prediction, contributing significantly to healthcare applications. The integrated Xception model elevates feature extraction for cardiac abnormality detection from ECG images [23]. Extracted features serve as inputs to machine learning models, enhancing their ability to discern intricate patterns and anomalies. This amalgamation of advanced feature extraction with robust machine learning algorithms contributes to the project's effectiveness in providing accurate diagnoses. The streamlined user interactions through Flask with SQLite underscore the system's practicality, offering secure signup, signin, and efficient testing for improved healthcare practices.

ISSN 2277-2685
IJESR/July-Sep. 2024/ Vol-14/Issue-3/221-238
Reddy Sai Ravi Kumar *et. al.,* / International Journal of Engineering & Science Research

ii) System Architecture:

- The process begins with the input data, which consists of ECG images of patients [23]. These images are the raw material for the prediction system.
- In the feature extraction phase, deep learning models are applied to the input data. These models include SqueezeNet, AlexNet, CNN [30, 31, 33], and an extension xception model, which are capable of automatically identifying relevant patterns and features within the ECG images.
- Once features are extracted from the ECG images, they are used as input for machine learning classification algorithms. The machine learning models process the extracted features and make predictions or classifications based on the information learned during training.
- The final output provides the results of the machine learning classifications indicating the presence or absence of specific cardiac abnormalities.

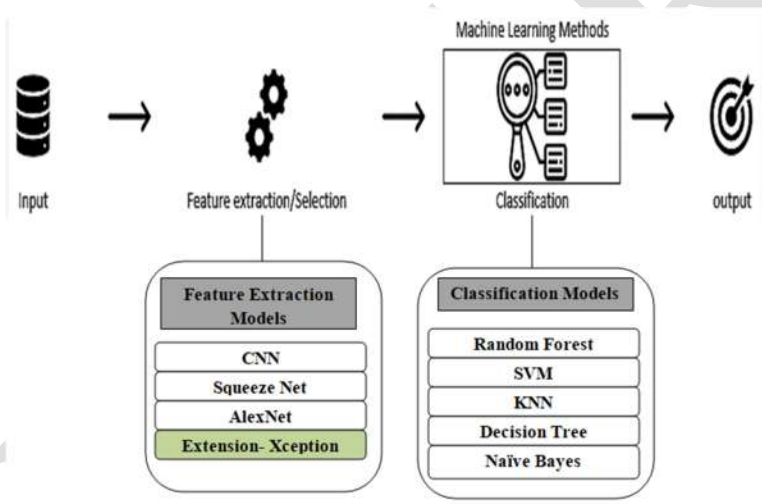

Fig 1 System Architecture

iii) Dataset collection:

ECG Image Data [23] shows the heart's electrical activity visually, helping diagnose heart issues and fueling automated diagnosis with machine learning. ECG Image Data consists of visual representations of heart electrical activity, aiding in diagnostics and research. These images are vital in detecting heart conditions and are used in machine learning to develop automated diagnosis. So, these are SAMPLE IMAGES.

Fig 2 ECG Image Data

iv) Image Processing:

Image processing plays a pivotal role in object detection within autonomous driving systems, encompassing several key steps. The initial phase involves converting the input image into a blob object, optimizing it for subsequent analysis and manipulation. Following this, the classes of objects to be detected are defined, delineating the specific categories that the algorithm aims to identify. Simultaneously, bounding boxes are declared, outlining the regions of interest within the image where objects are expected to be located. The processed data is then converted into a NumPy array, a critical step for efficient numerical computation and analysis.

The subsequent stage involves loading a pre-trained model, leveraging existing knowledge from extensive datasets. This includes reading the network layers of the pre-trained model, containing learned features and parameters vital for accurate object detection. Additionally, output layers are extracted, providing final predictions and enabling effective object discernment and classification.

Further, in the image processing pipeline, the image and annotation file are appended, ensuring comprehensive information for subsequent analysis. The color space is adjusted by converting from BGR to RGB, and a mask is created to highlight relevant features. Finally, the image is resized, optimizing it for further processing and analysis. This comprehensive image processing workflow establishes a solid foundation for robust and accurate object detection in the dynamic context of autonomous driving systems, contributing to enhanced safety and decision-making capabilities on the road.

v) Feature extraction:

Feature extraction [12] refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data.

Feature extraction can be accomplished manually or automatically [11]:

- Manual feature extraction requires identifying and describing the features that are relevant for a given problem and implementing a way to extract those features. In many situations, having a good understanding of the background or domain can help make informed decisions as to which features could be useful. Over decades of research, engineers and scientists have developed feature extraction methods for images, signals, and text. An example of a simple feature is the mean of a window in a signal.
- Automated feature extraction uses specialized algorithms or deep networks to extract features automatically from signals or images without the need for human intervention. This technique can be very useful when you want to move quickly from raw data to developing machine learning algorithms. Wavelet scattering is an example of automated feature extraction [11, 12].

With the ascent of deep learning, feature extraction has been largely replaced by the first layers of deep networks – but mostly for image data. For signal and time-series applications, feature extraction remains the first challenge that requires significant expertise before one can build effective predictive models.

ISSN 2277-2685
IJESR/July-Sep. 2024/ Vol-14/Issue-3/221-238
Reddy Sai Ravi Kumar *et. al.,* / International Journal of Engineering & Science Research

vi) Algorithms:

- **Cnn:** A Convolutional Neural Network (CNN) is a neural network tailored for visual data analysis. It excels in image-related tasks, automatically learning features to recognize objects and patterns. CNNs consist of interconnected layers, including convolution, pooling, and fully connected layers, effectively extracting image information and serving as valuable tools in computer vision and image analysis. We have used it for feature extraction and we have built the model for predictions. In our project, we employed CNN [30] for feature extraction and building predictive models.

```
np.random.seed(1000)  # Fix seed

models = Sequential()

models.add(Conv2D(filters=64, kernel_size=(5, 5), input_shape=(227, 227, 3), activation='relu'))
models.add(BatchNormalization(axis=3))
models.add(Conv2D(filters=64, kernel_size=(5, 5), activation='relu'))
models.add(MaxPooling2D((2, 2)))
models.add(BatchNormalization(axis=3))
models.add(Dropout(0.1))

models.add(Conv2D(filters=128, kernel_size=(5, 5), activation='relu'))
models.add(BatchNormalization(axis=3))
models.add(Conv2D(filters=128, kernel_size=(5, 5), activation='relu'))
models.add(MaxPooling2D((2, 2)))
models.add(BatchNormalization(axis=3))
models.add(Dropout(0.1))

models.add(Conv2D(filters=256, kernel_size=(5, 5), activation='relu'))
models.add(BatchNormalization(axis=3))
models.add(Conv2D(filters=256, kernel_size=(5, 5), activation='relu'))
models.add(MaxPooling2D((2, 2)))
models.add(BatchNormalization(axis=3))
models.add(Dropout(0.1))
```

Fig 3 CNN

- **Squeeze net:** SqueezeNet [20] is a compact and efficient CNN architecture for image classification. It's known for its small size and competitive accuracy, achieved through smart design choices like 1x1 convolutional filters and Squeeze-and-Excitation blocks. This compact size allows for faster real-time applications and works well on resource-constrained devices. In our project, we employed squeeze net for feature extraction and building predictive models.

```
def fire_module(x, squeeze, expand):
    y = Conv2D(filters=squeeze, kernel_size=1, activation="relu", padding="same")(x)
    y = BatchNormalization(momentum=bnmomentum)(y)
    y1 = Conv2D(filters=expand // 2, kernel_size=1, activation="relu", padding="same")(y)
    y1 = BatchNormalization(momentum=bnmomentum)(y1)
    y3 = Conv2D(filters=expand // 2, kernel_size=3, activation="relu", padding="same")(y)
    y3 = BatchNormalization(momentum=bnmomentum)(y3)
    return concatenate([y1, y3])

def SqueezeNet(input_shape, num_classes):
    input = Input(shape=input_shape)
    y = Conv2D(kernel_size=7, filters=96, strides=2, padding="same", activation="relu")(input)
    y = BatchNormalization(momentum=bnmomentum)(y)
    y = MaxPooling2D(pool_size=3, strides=2)(y)
    y = fire_module(y, 16, 64)
    y = fire_module(y, 16, 64)
    y = fire_module(y, 32, 128)
    y = MaxPooling2D(pool_size=3, strides=2)(y)
    y = fire_module(y, 32, 128)
    y = fire_module(y, 48, 192)
    y = fire_module(y, 48, 192)
    y = fire_module(y, 64, 256)
    y = MaxPooling2D(pool_size=2)(y)
    y = fire_module(y, 64, 256)
    y = Conv2D(kernel_size=1, filters=1000, strides=1, padding="same", activation="relu")(y)
    y = GlobalAveragePooling2D()(y)
    y = Dense(num_classes, activation="softmax")(y)
    model = Model(input, y)
    return model
```

Fig 4 Squeeze net

- **Alex net:** AlexNet is a pioneering CNN architecture known for winning the ImageNet challenge in 2012. It's deep with eight layers, including five convolutional layers and three fully connected ones. [21] AlexNet introduced ReLU activation and dropout, marking a breakthrough in deep learning for image classification. Its success paved the way for further advancements in AI. In our project, we employed alex net for feature extraction and building predictive models.

```
#Instantiation
AlexNet = Sequential()

#1st Convolutional Layer
AlexNet.add(Conv2D(filters=96, input_shape=(224,224,3), kernel_size=(11,11), strides=(4,4), padding='same'))
AlexNet.add(BatchNormalization())
AlexNet.add(Activation('relu'))
AlexNet.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='same'))

#2nd Convolutional Layer
AlexNet.add(Conv2D(filters=256, kernel_size=(5, 5), strides=(1,1), padding='same'))
AlexNet.add(BatchNormalization())
AlexNet.add(Activation('relu'))
AlexNet.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='same'))

#3rd Convolutional Layer
AlexNet.add(Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), padding='same'))
AlexNet.add(BatchNormalization())
AlexNet.add(Activation('relu'))

#4th Convolutional Layer
AlexNet.add(Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), padding='same'))
AlexNet.add(BatchNormalization())
AlexNet.add(Activation('relu'))
```

Fig 5 Alexnet

- **Random Forest:** Random Forest [44] is an ensemble learning algorithm that combines multiple decision trees to make predictions. Each tree in the forest independently predicts the class, and the final prediction is determined by voting or averaging. Random Forest is known for its robustness against overfitting and its ability to handle high-dimensional data, making it an ideal choice for classification tasks. In this project, it plays a crucial role in leveraging the features extracted by deep learning models for accurate cardiac abnormality classification.

```
from sklearn.ensemble import RandomForestClassifier
RF_model = RandomForestClassifier(n_estimators = 50, random_state = 42)
RF_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_RF = RF_model.predict(X_test_features)
#Inverse le transform to get original label back.
prediction_RF = le.inverse_transform(prediction_RF)

rf_acc_xec = accuracy_score(test_labels, prediction_RF)
rf_prec_xec = precision_score(test_labels, prediction_RF,average='weighted')
rf_rec_xec = recall_score(test_labels, prediction_RF,average='weighted')
rf_f1_xec = f1_score(test_labels, prediction_RF,average='weighted')
```

Fig 5 Random forest

- **Support Vector Machine (SVM):** Support Vector Machine is a powerful supervised learning algorithm used for classification and regression. SVM aims to find the hyperplane that best separates different classes in the feature space while maximizing the margin. SVM can handle both linear and non-linear data using kernel functions. Its strengths lie in its effectiveness for high-dimensional data and its ability to generalize well to new, unseen data. In this project, [17] SVM is utilized to further enhance the accuracy of cardiac abnormality detection.

```
from sklearn.svm import SVC
svm_model = SVC()
svm_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_svm = svm_model.predict(X_test_features)
#Inverse le transform to get original label back.
prediction_svm = le.inverse_transform(prediction_svm)

svm_acc_xec = accuracy_score(test_labels, prediction_svm)
svm_prec_xec = precision_score(test_labels, prediction_svm,average='weighted')
svm_rec_xec = recall_score(test_labels, prediction_svm,average='weighted')
svm_f1_xec = f1_score(test_labels, prediction_svm,average='weighted')
```

Fig 6 SVM

- **K-Nearest Neighbors (KNN):** K-Nearest Neighbors is an instance-based learning algorithm used for classification and regression. It makes predictions by finding the k-nearest data points in the feature space and determining the majority class among them. KNN is simple to understand and implement and is especially suitable for multi-class problems. It can also handle noisy data effectively. In this project, KNN provides an additional layer of classification to ensure accurate results [17].

```
from sklearn.neighbors import KNeighborsClassifier
knn_model = KNeighborsClassifier(n_neighbors=3)
knn_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_knn = knn_model.predict(X_test_features)
#Inverse le transform to get original label back.
prediction_knn = le.inverse_transform(prediction_knn)

knn_acc_xec = accuracy_score(test_labels, prediction_knn)
knn_prec_xec = precision_score(test_labels, prediction_knn,average='weighted')
knn_rec_xec = recall_score(test_labels, prediction_knn,average='weighted')
knn_f1_xec = f1_score(test_labels, prediction_knn,average='weighted')
```

Fig 7 KNN

- **Decision Tree:** Decision Tree is a tree-like structure where each node represents a feature, each branch represents a decision rule, and each leaf node represents a class label. It makes decisions by recursively splitting the dataset based on informative features. Decision Trees are interpretable and can handle both categorical and numerical data. They are particularly useful for capturing non-linear relationships in the data. In this project, Decision Trees play a role in refining the classification process.

```
from sklearn.tree import DecisionTreeClassifier
dt_model = DecisionTreeClassifier(max_depth=30)
dt_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_dt = dt_model.predict(X_test_features)
#Inverse le transform to get original label back.
prediction_dt = le.inverse_transform(prediction_dt)

dt_acc_xec = accuracy_score(test_labels, prediction_dt)
dt_prec_xec = precision_score(test_labels, prediction_dt,average='weighted')
dt_rec_xec = recall_score(test_labels, prediction_dt,average='weighted')
dt_f1_xec = f1_score(test_labels, prediction_dt,average='weighted')
```

Fig 8 Decision tree

- **Naive Bayes:** Naive Bayes is a probabilistic classifier based on Bayes' theorem with an assumption of feature independence. It calculates the probability of a data point belonging to a class based on the probabilities of its features occurring in each class. Naive Bayes is known for its simplicity, speed, and effectiveness in tasks like text classification and spam filtering. In this project, [45] Naive Bayes is employed to further complement the classification process and ensure robustness.

```
from sklearn.naive_bayes import GaussianNB
nb_model = GaussianNB()
nb_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_nb = nb_model.predict(X_test_features)
#Inverse le transform to get original label back.
prediction_nb = le.inverse_transform(prediction_nb)

nb_acc_xec = accuracy_score(test_labels, prediction_nb)
nb_prec_xec = precision_score(test_labels, prediction_nb,average='weighted')
nb_rec_xec = recall_score(test_labels, prediction_nb,average='weighted')
nb_f1_xec = f1_score(test_labels, prediction_nb,average='weighted')
```

Fig 9 Naïve bayes

- **xception:** In this project we have developed xception model, Xception, short for "Extreme Inception," is a Google-developed deep learning architecture specialized in image recognition. It's notable for its efficient depth-wise separable convolutions, reducing computation while boosting performance. Xception is highly accurate and versatile, making it a significant innovation in computer vision deep learning. In our project, we employed xception algo for feature extraction and building predictive models.

```
base_model = Xception(weights='imagenet', include_top=False )

# add a global spatial average pooling layer
x = base_model.output
x = GlobalAveragePooling2D()(x)
# let's add a fully-connected layer
x = Dense(512, activation='relu')(x)

x = Dropout(0.3)(x)
# and a logistic layer -- let's say we have 200 classes
predictions = Dense(4, activation='softmax')(x)

# this is the model we will train
model2 = Model(inputs=base_model.input, outputs=predictions)

model2.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics=["accuracy",f1_m,precision_m, recall_m])
model2.summary()
```

Fig 10 Xception model

4. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

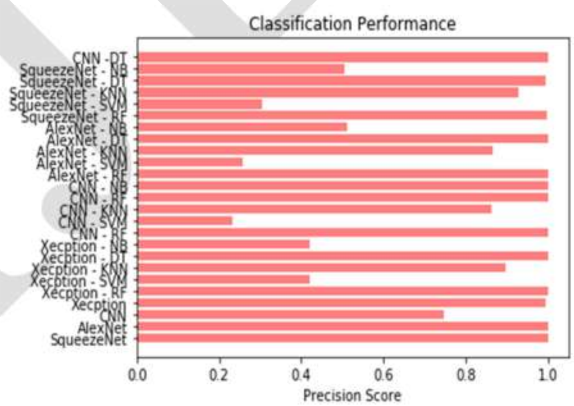

Fig 11 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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

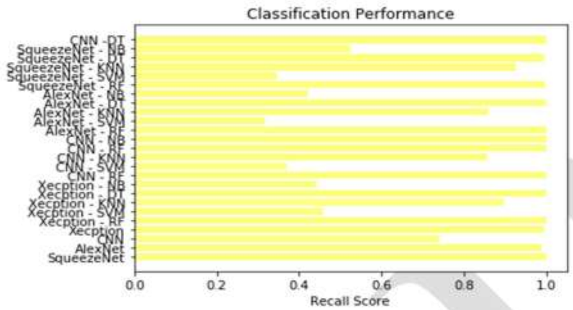

Fig 12 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

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

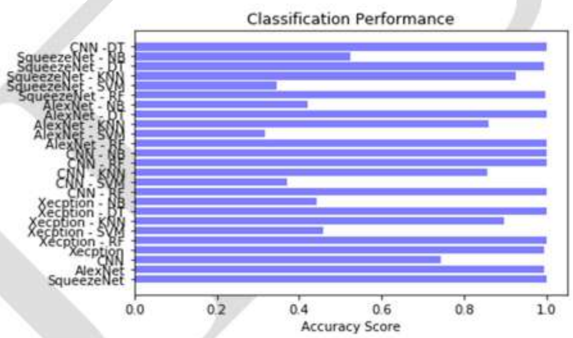

Fig 13 Accuracy graph

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

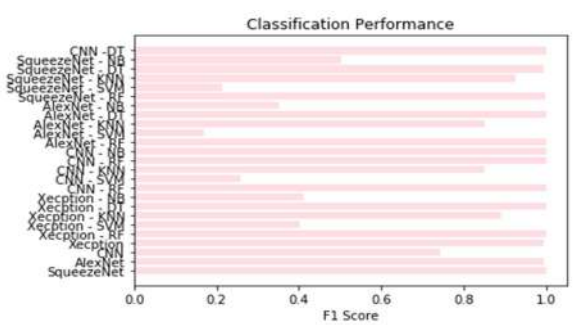

Fig 14 F1Score

	ML Model	Accuracy	Precision	Recall	F1_score
0	SqueezeNet	1.000	1.000	1.000	1.000
1	AlexNet	0.995	1.000	0.990	0.995
2	CNN	0.743	0.747	0.742	0.745
3	**Xception**	**0.995**	**0.995**	**0.995**	**0.995**
4	Xception - RF	1.000	1.000	1.000	1.000
5	Xecption - SVM	0.459	0.421	0.459	0.403
6	Xecption - KNN	0.896	0.899	0.896	0.892
7	Xecption - DT	1.000	1.000	1.000	1.000
8	Xecption - NB	0.444	0.421	0.444	0.411
9	CNN - RF	1.000	1.000	1.000	1.000
10	CNN - SVM	0.370	0.232	0.370	0.259
11	CNN - KNN	0.858	0.862	0.858	0.850
12	CNN - RF	1.000	1.000	1.000	1.000
13	CNN - NB	1.000	1.000	1.000	1.000
14	AlexNet - RF	1.000	1.000	1.000	1.000
15	AlexNet - SVM	0.317	0.257	0.317	0.169
16	AlexNet - KNN	0.861	0.865	0.861	0.850
17	AlexNet - DT	1.000	1.000	1.000	1.000
18	AlexNet - NB	0.420	0.513	0.420	0.352
19	SqueezeNet - RF	0.998	0.998	0.998	0.998
20	SqueezeNet - SVM	0.346	0.305	0.346	0.215
21	SqueezeNet - KNN	0.927	0.928	0.927	0.925
22	SqueezeNet - DT	0.994	0.994	0.994	0.994
23	SqueezeNet - NB	0.525	0.504	0.525	0.503
24	CNN -DT	1.000	1.000	1.000	1.000

Fig 15 Performance Evaluation table

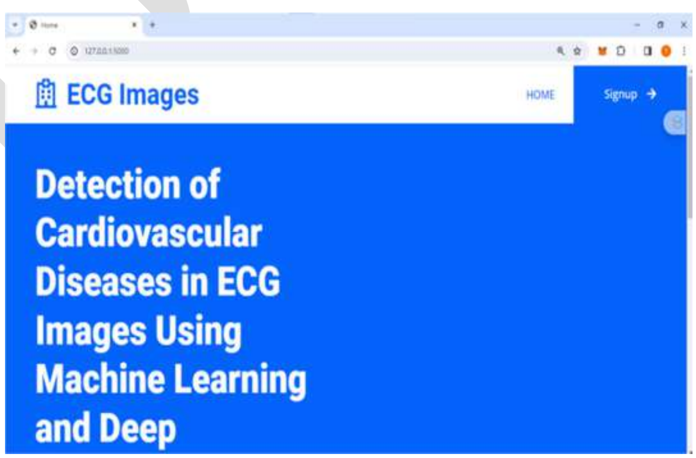

Fig 16 Home page

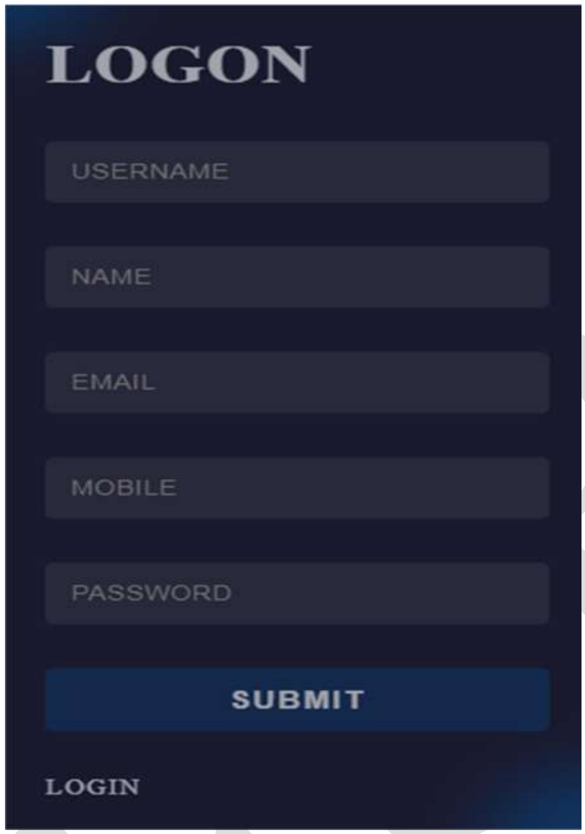

Fig 17 Registration page

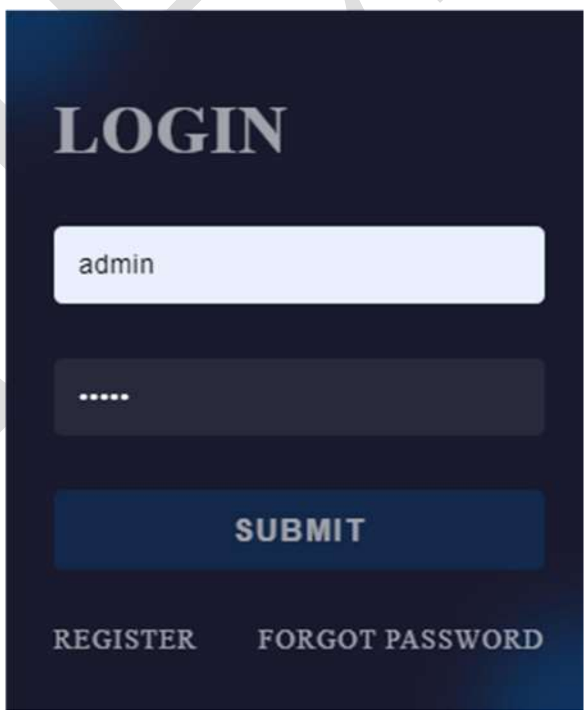

Fig 17 Login page

ISSN 2277-2685
IJESR/July-Sep. 2024/ Vol-14/Issue-3/221-238
Reddy Sai Ravi Kumar *et. al.,* / International Journal of Engineering & Science Research

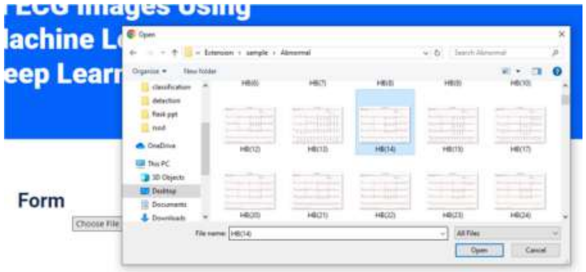

Fig 18 Input image folder

Fig 19 Upload input image

Fig 20 Predict result for given input

5. CONCLUSION

The project successfully aimed to predict major cardiac abnormalities from ECG images [32], employing deep learning techniques. Notably, models like SqueezeNet and AlexNet achieved near-perfect classification, showcasing the efficacy of deep learning in accurate predictions. The extension, Xception, demonstrated exceptional accuracy in predicting cardiac abnormalities. This highlights the effectiveness of advanced deep learning models in enhancing diagnostic capabilities beyond traditional methods. While deep learning models excelled, traditional machine learning had mixed results when coupled with deep learning-based feature extraction. This suggests the unique advantages of deep learning in extracting complex patterns from ECG images [32, 33]. The integration of Flask with SQLite ensured a seamless and secure front-end for user testing. This practical and secure interface enhances the user experience and facilitates efficient testing of the developed models. The project's success underscores the potential of AI, particularly deep learning, in the early detection of cardiovascular diseases. This has significant implications for improving healthcare outcomes by enabling timely interventions. The project concludes by emphasizing the importance of leveraging AI for early disease detection. It encourages further research in this vital healthcare domain, recognizing the transformative impact of advanced technologies on improving diagnostic capabilities and patient outcomes.

6. FUTURE SCOPE

To further improve the proposed [33] CNN model's performance, future research could focus on fine-tuning its hyperparameters. By systematically adjusting parameters like learning rates, batch sizes, and dropout rates, the model's accuracy and efficiency can be enhanced. The integration of the CNN model into the Industrial Internet of Things (IIoT) realm presents exciting possibilities. Beyond cardiovascular disease prediction, the model can be adapted for various classification tasks within IIoT applications, such as anomaly detection in industrial equipment or quality control in manufacturing processes. Exploring additional layers or different network architectures can lead to performance enhancements. Researchers may investigate the incorporation of more convolutional or recurrent layers, or even explore emerging network architectures to further boost the [31, 33, 36]CNN model's capabilities in detecting cardiovascular diseases. By accommodating larger and more diverse datasets, the system's effectiveness can be broadened. This expansion should include data from various sources and populations to ensure the model's generalizability, making it applicable across a wide range of cardiovascular diseases and diverse patient profiles.

REFERENCES

[1] World Health Organization (WHO), “Cardiovascular diseases,” Jun. 11, 2021. Accessed: Dec. 27, 2021. [Online]. Available: https://www.who. int/health-topics/cardiovascular-diseases

[2] Government of Westren Australia, Department of Health, “Common medical tests to diagnose heart conditions,” Accessed: Dec. 29, 2021. [Online]. Available: https://www.healthywa.wa.gov.au/Articles/A_ E/Common-medical-tests-to-diagnose-heart-conditions

[3] M. Swathy and K. Saruladha, “A comparative study of classification and prediction of cardio-vascular diseases (CVD) using machine learning and deep learning techniques,” ICT Exp., to be published, 2021. [Online]. Available: https://doi.org/10.1016/j.icte.2021.08.021

[4] R. R. Lopes et al., “Improving electrocardiogram-based detection of rare genetic heart disease using transfer learning: An application to phospholamban p.Arg14del mutation carriers,” Comput. Biol. Med., vol. 131, 2021, Art. no. 104262. [Online]. Available: https://doi.org/10.1016/j. compbiomed.2021.104262

[5] R. J. Martis, U. R. Acharya, and H. Adeli, “Current methods in electrocardiogram characterization,” Comput. Biol. Med., vol. 48, pp. 133–149, 2014. [Online]. Available: https://doi.org/10.1016/j.compbiomed.2014. 02.012

[6] A. Rath, D. Mishra, G. Panda, and S. C. Satapathy, “Heart disease detection using deep learning methods from imbalanced ECG samples,” Biomed. Signal Process. Control, vol. 68, 2021, Art. no. 102820. [Online]. Available: https://doi.org/10.1016/j.bspc.2021.102820

[7] A. Mincholé and B. Rodriguez, “Artificial intelligence for the electrocardiogram,” Nature Med., vol. 25, no. 1, pp. 22–23, 2019. [Online]. Available: https://doi.org/10.1038/s41591-018-0306-1

[8] A.Isin and S. Ozdalili, “Cardiac arrhythmia detection using deep learning,” Procedia Comput. Sci., vol. 120, pp. 268–275, 2017. [Online]. Available: https://doi.org/10.1016/j.procs.2017.11.238

[9] H. Bleijendaal et al., “Computer versus cardiologist: Is a machine learning algorithm able to outperform an expert in diagnosing phospholamban (PLN) p.Arg14del mutation on ECG?,” Heart Rhythm, vol. 18, no. 1, pp. 79–87, 2020. [Online]. Available: https://doi.org/10.1016/j. hrthm.2020.08.021

[10] U. R. Acharya, H. Fujita, O. S. Lih, M. Adam, J. H. Tan, and C. K. Chua, “Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network,” Knowl.-Based Syst., vol. 132, pp. 62–71, 2017. [Online]. Available: https://doi.org/10.1016/j. knosys.2017.06.003

[11] M. Kantardzic, Data Mining: Concepts, Models, Methods, and Algorithms, 3rd ed. Hoboken, NJ, USA: Wiley, 2020.

[12] S. García, J. Luengo, and F. Herrera, Data Preprocessing in Data Mining, 1st ed. Berlin, Germany: Springer, 2015.

[13] G. Dougherty, Pattern Recognition and Classification: An Introduction. Berlin, Germany: Springer, 2013.

[14] A. Subasi, Practical Machine Learning for Data Analysis Using Python. Cambridge, MA, USA: Academic, 2020.

[15] J. Soni, U. Ansari, D. Sharma, and S. Soni, “Predictive data mining for medical diagnosis: An overview of heart disease prediction,” Int. J. Comput. Appl., vol. 17, no. 8, pp. 43–48, 2011.

[16] K. Dissanayake and M. G. Md Johar, “Comparative study on heart disease prediction using feature selection techniques on classification algorithms,” Appl. Comput. Intell. Soft Comput., vol. 2021, 2021, Art. no. 5581806. [Online]. Available: https://doi.org/10.1155/2021/5581806

[17] A. H. Gonsalves, F. Thabtah, R. M. A. Mohammad, and G. Singh, “Prediction of coronary heart disease using machine learning: An experimental analysis,” in Proc. 3rd Int. Conf. Deep Learn. Technol., 2019, pp. 51–56. [Online]. Available: https://doi.org/10.1145/3342999.3343015

[18] H. Kim, M. I. M. Ishag, M. Piao, T. Kwon, and K. H. Ryu, “A data mining approach for cardiovascular disease diagnosis using heart rate variability and images of carotid arteries,” Symmetry, vol. 8, no. 6, 2016, Art. no. 47. [Online]. Available: https://doi.org/10.3390/sym8060047

[19] T. Ozcan, “A new composite approach for COVID-19 detection in X-ray images,” Appl. Soft Comput., vol. 111, 2021, Art. no. 107669. [Online]. Available: https://doi.org/10.1016/j.asoc.2021.107669

[20] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 MB model size,” 2016, arXiv:1602.07360.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Adv. Neural Inf. Process. Syst., vol. 25, pp. 1097–1105, 2012.

[22] A. H. Khan, M. Hussain, and M. K. Malik, “Cardiac disorder classification by electrocardiogram sensing using deep neural network,” Complexity, vol. 2021, 2021, Art. no. 5512243. [Online]. Available: https://doi.org/10. 1155/2021/5512243

[23] A. H. Khan and M. Hussain, “ECG images dataset of cardiac patients,” Mendeley Data, vol. V2, 2021. [Online]. Available: https://doi.org/10. 17632/gwbz3fsgp8.2

[24] C. Potes, P. Saman, A. Rahman, and B. Conroy, "Ensemble of featurebased and deep learning-based classifiers for detection of abnormal heart sounds," in Proc. Comput. Cardiol. Conf., 2016, pp. 621–624.

[25] A. Nannavecchia, F. Girardi, P. R. Fina, M. Scalera, and G. Dimauro, "Personal heart health monitoring based on 1D convolutional neural network," J. Imag., vol. 7, no. 2, 2021, Art. no. 26. [Online]. Available: https://doi.org/10.3390/jimaging7020026

[26] Q. Zhang, D. Zhou, and X. Zeng, "HeartID: A multiresolution convolutional neural network for ECG-based biometric human identification in smart health applications," IEEE Access, vol. 5, pp. 11805–11816, 2017. [Online]. Available: https://doi.org/10.1109/ACCESS.2017.2707460

[27] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, and R. S. Tan, "A deep convolutional neural network model to classify heartbeats," Comput. Biol. Med., vol. 89, pp. 389–396, 2017. [Online]. Available: https: //doi.org/10.1016/j.compbiomed.2017.08.022

[28] R. Bharti, A. Khamparia, M. Shabaz, G. Dhiman, S. Pande, and P. Singh, "Prediction of heart disease using a combination of machine learning and deep learning," Comput. Intell. Neurosci., vol. 2021, 2021, Art. no. 8387680. [Online]. Available: https://doi.org/10.1155/2021/ 8387680

[29] P. Bizopoulos and D. Koutsouris, "Deep learning in cardiology," IEEE Rev. Biomed. Eng., vol. 12, pp. 168–193, 2018. [Online]. Available: https: //doi.org/10.1109/RBME.2018.2885714

[30] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ECG classification by 1-D convolutional neural networks," IEEE Trans. Biomed. Eng., vol. 63, no. 3, pp. 664–675, Mar. 2016. [Online]. Available: https: //doi.org/10.1109/TBME.2015.2468589

[31] M. Zubair, J. Kim, and C. Yoon, "An automated ECG beat classification system using convolutional neural networks," in Proc. 6th Int. Conf. IT Convergence Secur., 2016, pp. 1–5. [Online]. Available: https://doi.org/ 10.1109/ICITCS.2016.7740310

[32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4510–4520.

[33] T. Rahman et al., "COV-ECGNET: COVID-19 detection using ECG trace images with deep convolutional neural network," 2021, arXiv:2106.00436.

[34] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2261–2269.

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2818–2826.

[36] A. Pal, R. Srivastva, and Y. N. Singh, "CardioNet: An efficient ECG arrhythmia classification system using transfer learning," Big Data Res., vol. 26, 2021, Art. no. 100271. [Online]. Available: https://doi.org/10.1016/j.bdr.2021.100271

[37] R. Avanzato and F. Beritelli, "Automatic ECG diagnosis using convolutional neural network," Electronics, vol. 9, no. 6, 2020, Art. no. 951. [Online]. Available: https://doi.org/10.3390/electronics9060951

[38] U. R. Acharya, H. Fujita, S. L. Oh, Y. Hagiwara, J. H. Tan, and M. Adam, “Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals,” Inf. Sci., vol. 415– 416, pp. 190–198, 2017. [Online]. Available: https://doi.org/10.1016/j.ins. 2017.06.027

[39] M. Naz, J. H. Shah, M. A. Khan, M. Sharif, M. Raza, and R. Damaševiˇ cius, “From ECG signals to images: A transformation based approach for deep learning,” PeerJ Comput. Sci., vol. 7, 2021, Art. no. e386, doi: 10.7717/peerj-cs.386.

[40] H. El-Amir and M. Hamdy, Deep Learning Pipeline: Building a Deep Learning Model With TensorFlow. New York, NY, USA: Apress Media, 2020.

[41] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, “Support vector machines,” IEEE Intell. Syst. Appl., vol. 13, no. 4, pp. 18–28, Apr. 1998. [Online]. Available: https://doi.org/10.1109/5254.708428

[42] M. Abubaker and W. M. Ashour, “Efficient data clustering algorithms: Improvements over Kmeans,” Int. J.Intell. Syst. Appl., vol. 5, no. 3, pp. 37–49, 2013. [Online]. Available: https://doi.org/10.5815/ijisa.2013.03.04

[43] B. Charbuty and A. Abdulazeez, “Classification based on decision tree algorithm for machine learning,” J. Appl. Sci. Technol. Trends, vol. 2, no. 1, pp. 20–28, 2021.

[44] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[45] E. Miranda, E. Irwansyah, A. Y. Amelga, M. M. Maribondang, and M. Salim, “Detection of cardiovascular disease risk’s level for adults using naive Bayes classifier,” Healthcare Inform. Res., vol. 22, no. 3, pp. 196–205, 2016. [Online]. Available: https://doi.org/10.4258/hir.2016. 22.3.196