Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Novel Deep Learning Approach for Myocardial Infarction Detection and Multi-Label Classification

SIDRA ABBAS[1], STEPHEN OJO[2], MOEZ KRICHEN[3,4], MEZNAH A. ALAMRO[5], ALAEDDINE MIHOUB [6], LUCIA VILCEKOVA[7]

[1]Department of Computer Science, COMSATS University Islamabad, Islamabad, Pakistan
[2]Department of Electrical and Computer Engineering, College of Engineering Anderson, Anderson University, 29621 South Carolina, USA
[3]FCSIT, Al-Baha University, Al-Baha 65528, Saudi Arabia
[4]ReDCAD Laboratory, University of Sfax, Sfax 3038, Tunisia
[5]Department of Information Technology, College of Computer & Information Science, Princess Nourah Bint Abdul Rahman University, Riyadh 11564, Saudi Arabia
[6]Department of Management Information Systems, College of Business and Economics, Qassim University, P.O. Box 6640, Buraidah 51452, Saudi Arabia
[7]Faculty of Management, Comenius University Bratislava, Odbojárov 10, 82005 Bratislava 25, Slovakia

Corresponding author: lucia.vilcekova@fm.uniba.sk, sidraabbas@ieee.org

ABSTRACT One of the main causes of death from cardiovascular diseases is Myocardial Infarction (MI), which is brought on by coronary artery problems. Myocardial infarction is a pathological condition resulting from an anatomical issue with the Left Ventricle (LV). MI is a potentially fatal cardiac disease for which prompt medical attention can lower the fatality risk. This paper proposes a Deep Learning (DL) based approach to robustly detect binary and multiclass myocardial infarction (MI) in two environments, i.e., with and without data balancing. We employed a Myocardial infarction dataset that contains 1700 MI patients' medical records. The data preprocessing step is performed, during which data balancing and normalization are carried out. In many real-world medical datasets, class imbalance is a serious problem since it often causes the proposed algorithms to predict the majority class. We apply a class imbalance handling technique to solve the imbalance issue and create a balanced and trustworthy prediction approach. To ensure the generalizability and performance comparison, we employ various deep learning algorithms such as Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Deep Neural Network (DNN), Long Short-Term Memory (LSTM) without data balancing first, and then after that, with data balancing to enhance model performance. Experiments reveal that the DNN model outperformed others by applying the class imbalance handling technique compared to the method without balancing data. The DNN model attained a maximum test accuracy of 99.39% and a test loss of 0.0252 for the binary class and achieved a maximum test accuracy of 99.74% and a test loss of 0.0115 for multiclass categorization.

INDEX TERMS Myocardial Infraction, Cardiovascular Disease, Deep Learning, Convolutional Neural Network, Recurrent Neural Network, Deep Neural Network, Long Short Term Memory, Healthcare

I. INTRODUCTION

The heart is the body's primary organ, providing blood to all other organs and bones. If it stops working, vital organs such as the brain will stop working and will quickly die. Cardiovascular disease has attracted much attention in medical research due to its possible fatality. Cardiovascular disease diagnosis and treatment provide complex problems requiring computerized prediction to improve future medical treatments [1]. Two coronary arteries carry oxygenated blood that the heart muscle cells require to remain alive and functional. Heart cells perish if the blood supply to the heart is interrupted by obstructed arteries or sub-branches. This problem is known as ischemia; a myocardial infarction (MI) occurs when heart cells die from prolonged lack of blood flow [2].

Several factors contribute to the occurrence of Cardiovascular disease, which can be broadly categorized into genetic predispositions that include an extended family history of the disease, environmental factors such as smoking tobacco, abusing drugs, leading a sedentary lifestyle, and comorbidities such as uncontrolled diabetes, hypertension, dyslipidemia, associated lung diseases, mental illnesses, and other conditions that make a person more susceptible to MI [3]. It is an expensive endeavor. About 735,000 people have a heart attack in the United States alone each year, and 71.5% of those patients are first responders [4]. The prediction states that between 2015 and 2030, the incidence of coronary heart disease, the primary cause of MI, will increase by 18% [5]. The predicted cost of cardiovascular disease management by 2035 is expected to reach 1.1 trillion USD, up from 555 billion USD in 2015 [6]

Early detection of MI is essential to prevent cardiac failure, arrhythmia, or unexpected death. A variety of evaluation modalities, including electrocardiograms (ECGs) [5], magnetic resonance imaging (MRI) [7] and echocardiography [8], can be used to identify MI. The most often used technique for supporting cardiac functions and assessing the health of the myocardial and left ventricle is magnetic resonance imaging (MRI) [7]. Myocardial infarction is a pathological condition resulting from an anatomical issue with the left ventricle (LV).

Artificial intelligence (AI) uses algorithms that constantly learn patterns to simulate human intellect [9]. In machine learning (ML), computers are taught to recognize patterns using algorithms [10]. ML is an instance of AI. Since the heart functions as a metronome and the healthcare industry is extremely nonlinear, the literature has developed several methods for the automatic detection and localization of MI. The wavelet-transform-based approach [11], time-based domain methods [2], [12], a polynomial fitting-based approach [13], and supervised ML classifier [14], [15] are some of these tactics. Even though these techniques are effective, they still have certain drawbacks when using them in clinical settings. The process of segmenting and establishing cardiac strains is difficult and time-consuming. Furthermore, the heart's spatiotemporal motion is excessively complicated and could make implementation challenging [16].

Recent advancements in deep learning models have led to a noteworthy trend in several research areas, including cardiac segmentation [17], [12], [18]. However, only study studies have deep learning systems to predict myocardial infarctions. Numerous research studies have been done on MI detection using deep learning approaches, but they still need improvement. For example, prior studies ignored the need for sufficient data preprocessing and did not employ high-quality data. This research proposed a technique based on DL classifiers by applying Synthetic Minority Oversampling (SMOTE) for class imbalance problems.

The main findings of this study are outlined here.

- This study proposed a Deep Learning (DL) based approach to robustly detect binary and multiclass myocardial infarction (MI) in two environments, i.e., with data balancing and without data balancing.
- In many real-world medical datasets, class imbalance is a serious problem since it often causes the proposed algorithms to predict the negative class. To solve the imbalance issue and create a trustworthy prediction system, we applied a class imbalance technique, and next, we applied z-score data normalization to scale the dataset.
- To ensure the generalizability and performance comparison, we employ various deep learning algorithms such as Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Deep Neural Network (DNN), Long Short-Term Memory (LSTM) without data balancing first, and then after that, with data balancing to enhance model performance. Experiments reveal that the DNN model outperformed by applying the class imbalance handling technique compared to the method without balancing data. The DNN model attained a maximum test accuracy of 99.39% and a test loss of 0.0252 for the binary class and achieved a maximum test accuracy of 99.74% and a test loss of 0.0115 for multiclass categorization.

The next sections of this article are meticulously organized to help readers understand the study's framework. Section 2 presents a summary of current study findings and provides background information. The technical details of the proposed framework, the dataset's details, data preprocessing methods, and DL models are all covered in detail in Section 3. The experimental results are presented in Section 4, and a discussion of most of the study's findings is also presented. The study's conclusion, included in Section 6, provides a thorough analysis and conclusions from the proposed work. Future directions of the current work are also presented in this section.

II. LITERATURE REVIEW

Authors in [19] generated a model to detect chronic myocardial ischemia (MIS) by validating a CT-based radiomics machine learning. Recurrent analysis of 154 patients with coronary artery disorders (CAD) using cardiac computed tomography angiography (CCTA) and SPECT-myocardial perfusion imaging (MPI) revealed 94 patients to have suffered from MI. Features were gathered from every CCTA cross-sectional image to determine myocardial segments. A radiomics signature was established using multivariate logistic regression. Lastly, the radiomics nomogram was developed using an accurate model of MIS that was created by combining clinically relevant variables with machine learning. After that, we added 18 MIS patients from a different medical facility and used data from 49 CAD patients to test the model. In the training, test, and validation sets, the nomogram's accuracy for MIS prediction was 0.839, 0.832, and 0.816. The nomogram, signature, and vascular stenosis had correct diagnosis values of 0.824, 0.736, and 0.708.

Authors in [20] assessed how well machine learning (ML) performed in predicting the long-term risk of myocardial infarction and cardiac death in asymptomatic participants by integrating clinical parameters with coronary artery calcium (CAC) and automating the quantification of epicardial adipose tissue (EAT). A total of 1912 asymptomatic participants from the random EISNER experiment with long-term follow-up following CAC scoring were included in the analysis. A fully automated deep-learning technique was used to quantify the volume and density of EAT. ML's AUC was considerably higher than both the CAC score and the risk of atherosclerotic cardiovascular disease (ASCardiovascular disease). Higher ML scores were associated with a higher chance of suffering events in subjects. These correlations continued in the multivariable analysis, including CAC and ASCardiovascular disease risk measures. Authors in [21] used ML-based algorithms to automatically assess the degree of MI based on physiological, clinical, and paraclinical characteristics. To assess MI, two different kinds of machine learning models are examined: regression models calculate the percentage of infracted Mycordium patients assumed to have an acute MI at the time of admission to the emergency room, and classification algorithms categorize the presence of infarction and persistent microvascular obstruction (PMO). The associated Delayed Enhancement MRI (DE-MRI) scans and hand annotations of the heart and scar tissues provide the base truth labels for these supervised methodologies. Cross-validation was used to assess the 150 instances involved in the experiments. According to the results, the best models produced mean errors of 0.056 and 0.012 for quantifying the MI (PMO inclusive) and 88.67 and 77.33% for the performances of the myocardium (infarct exclusive).

An ML method for forecasting MI incidents based on several environmental and demographic factors is presented in [22]. The KORA MI registry dataset for Augsburg, Germany, between 1998 and 2015 provided the MI events they used. The decision tree, random forest, multi-layer perceptron, gradient boosting, and ridge regression machine learning methods were tested. The models can accurately forecast the overall annual number of MIs. Authors in [23] used harmonized EHR data to forecast incident MI by comparing DL and ML models to a baseline logistic regression utilizing just "known" risk variables. A large-scale case-control study was conducted to determine the result of a 6-month incident MI. They addressed the imbalance in the dataset by comparing several over- and under-sampling strategies. They investigated boosted gradient machines, random forests, shallow and deep neural networks, and regularized logistics regression. A logistic regression model with a small list of "known" risk variables for MI served as the baseline model for comparison. Ten-fold cross-validation was used to find the hyper-parameters. There were 2.25 million patients without an MI diagnosis, compared to 25,911 with one. A deep neural network with random undersampling produced better classification results than previous techniques. With an F1 Score of 0.092 and an AUC of 0.835, the DNN demonstrated a moderate advantage over a LR model that included just "known" risk variables.

Authors in [24] aimed to evaluate the predictive potential of ML in the context of major adverse cardiovascular events (MACEs) prediction. MACEs and 24 chosen clinical factors were evaluated for significance using logistic regression (LR) analysis. The training dataset was used to construct six machine learning models using five-fold cross-validation, and the testing dataset was used to evaluate the models' prediction accuracy with LR. 30.6% was the MACE rate determined after a mean monitoring of 1.42 years. The following factors were independent predictors of MACEs: age, creatinine, cholesterol levels, and Killip classification. The random forest (RF) algorithm had the greatest performance in the training dataset, with an accuracy of (0.734, 0.647–0.820) and an AUC of (0.749, 0.644–0.853). The RF demonstrated the greatest significant survival difference when differentiating patients with and without MACEs in the testing dataset. Authors in [25] offers a novel method for examining patient histories connected to cardiac conditions. A novel feature selection and ranking method is put forward to purify the high-preference characteristics that aid in the earlier diagnosis of MI. After covariance analysis of the projected feature vectors, probabilistic principal component analysis (PPCA) determines which vectors have the largest covariance. As a result, the dimensionality problem is banished from the dataset. Multi-linear regression (MLR) is used to identify the chosen salient characteristics to identify those combinations that are closely connected. They are then classified using support vector machines (SVM) based on the radial basis function (RBF). Patients with and without MI are included in the two classifications produced by SVM. Patients' clinical test results are used as a dataset for analysis, and the system's effectiveness is gauged. The system's performance is measured by correlating the expected patients and the death rate. The cardiac predictions are identified by combining these machine-learning techniques with the selected symptoms. The findings represent that the proposed framework is suitable for MI prediction.

Authors in [26] combined clinical characteristics with cardiac troponin concentrations at appearance or on serial testing to calculate the Collaboration for the Diagnosis and Evaluation of Acute Coronary Syndrome (CoDE-ACS) score (0–100), which represents a person's risk of MI. The models' efficiency was independently evaluated using data from 10,286 patients across seven cohorts after they had been trained on 10,038 patients. Regarding MI, CoDE-ACS performed exceptionally well in discriminating between subgroups. Compared to fixed cardiac troponin thresholds, it determined more patients' appearance as having a low probability of having an MI with an identical negative predictive value and less as having a high probability with a higher positive predictive value. Authors in [27] provided valuable insights into efficient preprocessing methods that yield clean, organized data for feature extraction and selection. In this work, distinct characteristics are extracted from photographs of ECG graphs. The various machine-learning techniques improve and streamline the diagnosing process. In particular, a heart attack will be indicated by fluctuations in the output of electrodes two and three; the other electrodes will also exhibit abnormalities. After applying 14 attributes to the current classifiers, the authors got more effective results;

GBC, for example, demonstrated a test accuracy of 98.79%. Ultimately, the gradient-boosting classifier demonstrated efficacy in distinguishing between various heart attacks. The author of the work [13] examined six ML models to predict heart disease with good outcomes. The six models employed were XGBoost, Decision Tree, Bagging, Support Vector Machine, Logistic Regression, and LightGBM. The corresponding accuracies of each model were 82.01%, 72.90%, 83.85%, 84.60%, and 72.80%. The LightGBM model was found to evaluate performance better than the others.

Electrocardiogram (ECG) signals can be used to diagnose myocardial infarction (MI) with great potential thanks to deep learning techniques. a CNN-based system that outperforms conventional machine learning techniques in terms of accuracy for automatic detection of MI using 12-lead ECG readings. Comparably, [28] created an RNN architecture that can identify MI from long-term ambulatory ECG data, emphasizing the possibility for in-the-moment monitoring and early cardiac event identification. Taken as a whole, these findings highlight how deep learning models, such as CNNs and RNNs, can enhance the accuracy of MI detection and open the door to better cardiac event monitoring and diagnosis. A sizable ECG picture dataset, including 21,801 records with 12 lead ECG time-series data from 18,869 patients, was offered from the PhysioNet PTB-XL dataset. Using the synthetic dataset as training material, a deep ECG image digitization model was created to turn the synthetic images into time-series data for assessment. To evaluate the quality of the picture digitization in comparison to the ground truth ECG time series, the signal-to-noise ratio (SNR) was computed. With an average signal recovery SNR of 27±2.8 dB, the results highlight the value of the suggested synthetic ECG image dataset for deep learning model training.

Author in [29] worked on multi-label classification problems that are becoming popular in the field of cardiac abnormality detection to forecast multiple heart diseases at once. The problem of various cardiovascular diseases (CVDs) co-occurring has been addressed by certain models that use attention-based CNNs and semi-supervised learning, respectively. These models improve the precision and resilience of ECG-based diagnostic systems by utilizing methods such as data augmentation, pseudo-label generation, and attention mechanisms. These models show the possibility for comprehensive cardiac risk assessment by simultaneously recognizing several cardiac problems or comorbidities from ECG data by using deep learning approaches, such as CNNs and attention processes. Improved multi-label classification and myocardial infarction (MI) detection from electrocardiogram (ECG) signals have been demonstrated in [30] through deep learning models. In order to effectively classify ECG readings into different heart diseases, research efforts have concentrated on creating residual networks and deep learning models. Furthermore, the incorporation of deep learning methods such as GoogleNet, AlexNet, and ResNet has proven to be highly accurate in the prediction and classification of cardiac disorders, such as congestive heart failure and arrhythmia. Moreover, the use of trained deep learning networks such as AlexNet and ResNet-18 in conjunction with spectrograms as input images has shown remarkable results in terms of classification accuracy, sensitivity, and precision, surpassing 99% when it comes to the identification of cardiac arrhythmias. In order to overcome obstacles and confirm the therapeutic usefulness of deep learning-based methods for improving cardiac care and patient outcomes, more study is necessary. A summary of the considered studies on machine learning for cardiac disease detection is shown in Table 1.

III. PROPOSED METHODOLOGY

This section explains in detail how the suggested approach will be implemented using methodologies, metrics for performance evaluation, SMOTE analysis, and dataset analysis. Figure 1 depicts the whole process used in this investigation to estimate myocardial infarction. To forecast MI and examine the data, we used the MI dataset. We use a data-preprocessing technique that balances the data SMOTE-wise and normalizes the data using z-score normalization. Moreover, we used DL classifiers without SMOTE at the beginning. The classifier was then run again on balanced data once the data had been balanced.

A. EXPERIMENTAL DATASET

MI is one of the most difficult issues in modern medicine. In the year following an acute myocardial infarction, there is a substantial death rate. All nations continue to have significant MI incidence rates. This is particularly true for the metropolitan population in highly developed nations, where irregular and sometimes unbalanced diets and chronic stressors are commonplace. For instance, more than a million Americans have MI annually, and between 200 and 300,000 of them pass away from acute MI before reaching a hospital. A database with 1700 records was established at the Krasnoyarsk Interdistrict Clinical Hospital in Russia to assess and forecast the consequences of MI in patients [31]. This work used data made available in August 2020 [32] in the UCI machine learning repository. There are 124 features in the dataset. 111 of those features include data about the demographics, medical histories, problems at the time of hospital admission, ECG findings, and the clinical interventions performed by the staff after the patient was admitted. The following 12 columns show various problems at four distinct points in time: (i) upon hospital arrival, (ii) 24 hours after hospitalization, (iii) 48 hours after hospitalization, and (iv) 72 hours after hospitalization. The data also includes records about the cause of death, broken down into seven groups. Predicting patients' vulnerability at the time of being admitted, which is crucial to their survival, was the main focus of this research. It is vital to take the essential preventative measures as soon as possible within the first few hours of hospitalization since failing to do so may have dire implications.

TABLE 1: Summary of Studies on Machine Learning for Cardiac Disease Detection

Year	Ref	Summary
2022	[19]	Developed a model using CT-based radiomics machine learning to detect chronic myocardial ischemia (MIS). Tested the model on CAD patients and achieved high accuracy.
2020	[20]	Evaluated the effectiveness of machine learning in predicting long-term myocardial infarction (MI) and cardiac death in asymptomatic participants. Used clinical parameters, coronary artery calcium (CAC), and epicardial adipose tissue quantification. ML outperformed traditional risk assessments.
2022	[21]	Utilized machine learning algorithms to automatically evaluate MI severity based on physiological, clinical, and paraclinical factors. Achieved accurate measurements of MI and myocardium function using MRI scans and hand annotations.
2022	[22]	Predicted MI incidents using environmental and demographic factors. Tested various machine learning models and accurately predicted annual MIs.
2020	[23]	Compared deep learning (DL) and machine learning (ML) models for predicting incident MI. DL models with random undersampling outperformed earlier methods.
2022	[24]	Evaluated the predictive power of machine learning for major adverse cardiovascular events (MACEs). Random forest algorithm performed best in predicting MACEs.
2022	[25]	Proposed a novel approach for analyzing patient histories related to cardiac problems using machine learning. Developed a framework for predicting MI based on clinical test results.
2023	[26]	Calculated the CoDE-ACS score using clinical characteristics and cardiac troponin concentrations to indicate MI risk. CoDE-ACS showed good performance in distinguishing MI subgroups.
2023	[27]	Explored preprocessing approaches for feature extraction and selection in ECG-based MI detection. Achieved high accuracy using gradient-boosting classifiers.
2023	[13]	Compared six machine learning models for predicting heart disease, with LightGBM performing the best.
2023	[28]	Developed deep learning models using CNNs and RNNs for MI detection and cardiac event monitoring. Demonstrated the relevance of synthetic ECG picture dataset for model training.
2023	[29]	Addressed multi-label classification issues in cardiac abnormality detection. Used attention-based CNNs and semi-supervised learning to predict multiple heart illnesses simultaneously.
2023	[30]	Applied deep learning models like ResNet and GoogleNet to classify ECG readings into heart disorders. Achieved high accuracy for cardiac arrhythmia identification.

1) Binary Classification

Binary classes are classification problems in which the target attribute has just two potential outcomes or classes. Another name for this is binary classification. Typically, binary classification aims to group occurrences into one of two groups according to input feature sets. The target feature in binary classification is frequently encoded as 0 or 1, where 0 denotes one class (such as negative), and 1 denotes the opposite class (such as positive). This research used a binary classification dataset to predict myocardial infarction.

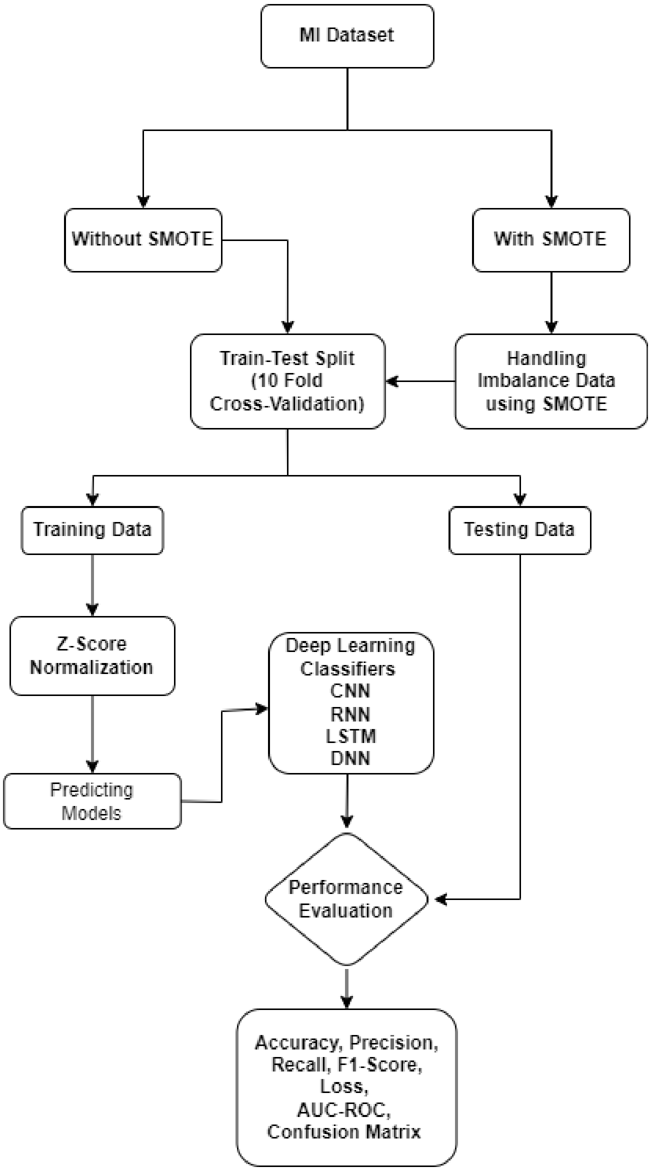

MI Dataset
Without SMOTE
With SMOTE
Train-Test Split (10 Fold Cross-Validation)
Handling Imbalance Data using SMOTE
Training Data
Testing Data
Z-Score Normalization
Predicting Models
Deep Learning Classifiers CNN RNN LSTM DNN
Performance Evaluation
Accuracy, Precision, Recall, F1-Score, Loss, AUC-ROC, Confusion Matrix

FIGURE 1: Proposed Model for Myocardial Infraction Prediction

2) Multi-classification

Multiclass classification is a problem in which the target variable contains more than two potential outcomes or classes. Multiclass classification entails selecting the proper class from a set of three or more alternative classes instead of binary classification, which aims to categorize cases into one of two groups. In multiclass classification, every class appears as a binary vector, and the target attribute is usually encoded using one-hot encoding. This research used the LET_IS column of the dataset as a categorical multiclass column and contains the following classes: Lethal outcome (cause) 0: unknown (alive) 1: cardiogenic shock 2: pulmonary edema

3: myocardial rupture 4: progress of congestive heart failure 5: thromboembolism 6: asystole 7: ventricular fibrillation.

B. DATA PREPROCESSING

Data preprocessing is essential in data assessment and machine learning systems. It must be cleaned and transformed to prepare raw data for further examination or machine learning model training. Appropriate preprocessing improves the caliber and efficiency of your models by resolving problems such as handling missing data, imbalanced data, etc. Class imbalance problems are addressed in classification problems by employing strategies like the Synthetic Minority Oversampling Technique (SMOTE), undersampling the majority class, or oversampling the minority class. This research utilized an SMOTE technique for balancing the data, which helped improve the performance of the proposed methodologies.

1) Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE is a prominent oversampling approach for unbalanced learning. Instead of just copying the samples, it creates new instances of minority classes [33]. Synthetic instances are constructed along the line segments, merging nearby instances from the minority class. The difference between Y_i and Y_j is initially computed if Y_i is an instance and Y_j is its closest neighbor. The variance is then multiplied by a lambda, a random value between zero and one. After that, a synthetic instance $Y_{\prime}^{i}$ between the initial point Y_i and the closest neighbor Y_j is created by adding Y_i the difference multiplied by lambda. Until the required number of minority instances is generated, this process is repeated. The following is the mathematical expression 1 utilized for SMOTE:

$$Y_{\prime}^{i} = Y_i + \lambda \left(Y_j - Y_i \right) \tag{1}$$

In this case, lambda is a fictitious value between 0 and 1. The number of instances to be produced can be manipulated by the α variable. The mathematical expression 2 displays the α variable working.

$$\alpha = \frac{\text{no. of instances in minority class after resample}}{\text{no. of instances in the majority class}} \tag{2}$$

Divide the dataset into test, validation, and training sets to assess how well machine learning models work. This prevents overfitting and aids in evaluating the models' capacity for generalization. This research divides the MI dataset first into 80% training and 20% testing sets. The following is the mathematical expression utilized for training and testing split:

Training Set Size

$$N_{train} = round(N \times train\,ratio) \tag{3}$$

Test Set Size

$$N_{test} = N - N_{train} \tag{4}$$

Equations 3 and 4 display the training and testing size. N is the total number of instances in the dataset. Train_ratio is the ratio of instances allocated to the training set. Test_ratio is the ratio of instances allocated to the test set. These equations give you the sizes of the training and test sets based on the specified ratios. Adjusting the train_ratio will directly affect the size of the training set, while the remaining samples implicitly determine the size of the test set after allocating to the training set.

2) Feature Normalization

Feature normalization is a data preprocessing approach utilized to scale the values of features to a similar range. This is done because many DL classifiers perform better when features are on a similar scale. This work employs a normalization approach to rescale the dataset features between 0 and 1, which facilitates their normalization and assembly in a comparable manner [34]. In this research, we apply Z-score scaling, also called standardization, and convert the features to have a mean of 0 and a standard deviation of 1. The mathematical expression 5 utilized for z-score scaling is:

$$Y_{standardized} = \frac{Y - \mu}{\sigma} \tag{5}$$

Where Y is the actual attribute value, μ is the mean of the attribute in the dataset, and σ is the standard deviation of the attribute.

Algorithm 1 presented a working proposed model for predicting myocardial infarction (MI). The required indicates the input, a dataset D_s containing information about myocardial infarction. The gain is the output to predict the probability of myocardial infarction, denoted as P_{MI}. Data Analysis function involves analyzing the dataset to understand its structure and characteristics. First, a model on unbalanced data is run, indicating that the initial dataset is imbalanced in the distribution of classes related to myocardial infarction. The data Preprocessing step involves preparing the data for modeling. It includes feature normalization using Z-score scaling. TrainDLClassifiers function is used for DL classifiers, and Deep learning classifiers are trained using preprocessed data. The dataset is split into training and test sets, and various deep learning models, such as Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and Deep Neural Networks (DNN), are utilized. Different loss functions, such as binary and categorical cross-entropy, are used based on the data type. The predictingModel function is utilized for the trained classifiers, and they predict outcomes on the test set. The algorithm then runs all models again on balanced data using the SMOTE (Synthetic Minority Over-sampling Technique) to handle data imbalance. This step ensures that both classes (myocardial infarction and non-myocardial infarction) are represented more evenly in the dataset. The algorithm's main work summarizes its main steps, including preprocessing the data, training the deep learning classifiers,

Algorithm 1 Proposed Model for Myocardial Prediction

```
**Require:** $D_s$ (MI Dataset)
**Gain:** $P_{MI}$ (Prediction of Myocardial Infraction)
**function** Data Analysis $D_a$
**function** Unbalanced Data $D$ {first model run on unbalanced data}
**function** Data Preprocessing $D_p$
    Feature Normalization applying Z-score scale $F_{norm}$ = $Y_{standardized}$ = $\frac{Y-\mu}{\sigma}$
**return** $F_{norm}$
**function** TrainDLClassifiers ($F_{norm}$, Y) {Utilized Deep Learning (DL) Classifiers }
$D_{split}$ $\leftarrow$ Dataset split into training and test set
RNN $\leftarrow$ Recurrent Neural Network
LSTM $\leftarrow$ Long Short Term Memory
CNN $\leftarrow$ Convolutional Neural Network
DNN $\leftarrow$ Deep Neural Network
    Use Binary Cross-entropy for binary data and Categorical Cross-entropy for Multiclass data
**return** TrainDLClassifiers
**function** PredictingModel (TrainDLClassifiers, test_set ($T_s$)
    $Y_{Predict}$ = DNNmodelTrain.predict($T_d$)
**return** $Y_{Predict}$
**function** Balanced Data $D$ {Second models run on balanced data}
    Applying SMOTE handle Imbalance data $S_b$ $Y_{\prime}^{i} = Y_i + \lambda\,(Y_j - Y_i)$
**Working** Main
    $F_{norm}$, Y $\leftarrow$ preprocessed data
    $DLmodel$ $\leftarrow$ TrainDLClassifiers ($F_{norm}$, Y)
    $Y_{Predict}$ $\leftarrow$ PredictingModel ($DLmodel$ , $T_d$ )
    Performance evaluation metrics
**return** Performance, $Y_{Predict}$
```

predicting outcomes, evaluating performance metrics, and returning the results.

C. DEEP LEARNING CLASSIFIERS

This research employs different DL classifiers, Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), Deep Neural Networks (DNN), and Long-Short-Term Memory (LSTM), to predict myocardial infarction. The details of every classifier are explained below.

1) Recurrent Neural Networks

A recurrent neural network (RNN) is an artificial neural network that processes data sequentially. Unlike standard feedforward neural networks, RNNs can display temporal dynamic behavior because their connections form directed cycles. Recurrent units or cells that preserve a hidden state makeup RNNs [35]. The network can retain sequence information by using the current input at every time step combined with the hidden state from the prior one. RNNs are incredibly deep because they retain a vector of activations for every timestep. Their depth makes it challenging for them to rain on because of the exploding and vanishing gradient issues they cause. Numerous attempts have been made to address the challenge of RNN training. The author of [36] successfully tackled the issue of vanishing gradients by creating the Long Short-Term Memory (LSTM) architecture, which is impervious to the vanishing gradient problem. Because of its ease of use, the LSTM has come to be accepted as a valid approach for solving the vanishing gradient problem.

2) Deep Neural Networks

A deep Neural Network (DNN) is an artificial neural network where several layers of deep architecture separate the input and output layers. A DNN is made up of layers that are made up of linked nodes or neurons. These networks can recognize and express intricate hierarchical patterns and characteristics in data. Deep learning, of which DNNs are essential, has attracted much interest and demonstrated considerable promise in several domains, including speech recognition, computer vision, natural language processing, and many more. Deep neural networks (DNNs) are highly effective in autonomously deriving hierarchical representations from data, which allows them to capture complex characteristics and relationships [37].

3) Convolutional Neural Networks

CNNs and conventional ANNs are similar in that both are made up of neurons that learn to optimize themselves [38]. ANNs are based on the fact that each neuron will continue to receive an input and carry out an operation. A single-wise scoring function (the weight) will be expressed by the network as a whole, from the raw picture vectors that are the input to the class score, which is the final output. The final layer will include loss functions linked to the classes. One of the main distinctions is that the layers of the CNN are made up of neurons arranged in three dimensions: the input's spatial dimensions and depth [39]. Convolutional Neural networks are widely used in computer vision systems. CNNs may comprise several convolutional layers, possibly separated by pooling and fully connected perceptron layers [40]. Conventional CNNs use shared weights in every layer to acquire features through convolutional layers. By decreasing the susceptibility of the output to distortions and the resolution of the dimensionality of intermediate map features, the feature pooling layer, also known as subsampling, broadens the scope of the network. The collected features are fed into a fully connected perceptron model at the last convolutional layer for feature classification and dimensionality reduction. A CNN architecture is created when these layers are stacked.

4) Long-Short Term Memory

Recurrent connections can increase neural network efficiency by employing their capability to recognize sequential reliances. However, the strategies used to train RNNs can significantly limit the memory generated by the recurrent connections. During the training phase, all models released thus far have had vanishing or exploding gradients, which has prevented the network from learning long-term sequential reliances in the data. The most often used model to address this issue is the long-short-term memory (LSTM) RNN. LSTM is among the most well-liked and effective approaches for lessening the outcomes of disappearing and bursting gradients [41]. This approach transforms hidden units from "sigmoid" or "tanh" networks to memory cells, whose inputs and outputs are controlled by gates. These gates maintain properties retrieved from earlier timesteps and regulate the information flow to hidden neurons. The hidden layer of LSTM has a significant degree of complexity. An ordinary LSTM contains roughly four times more parameters than a basic RNN for equal hidden layer sizes. Rather than trying to determine the minimal or perfect scheme, the goal when the LSTM technique was first proposed was to present a scheme that could improve learning long-range dependencies [42].

IV. EXPERIMENTAL RESULT AND DISCUSSION

This section discusses the proposed model, which combines DL classifiers to forecast myocardial infarction. Twenty percent of the dataset is used to test the model, while 80 percent is used to train it. This model learns from the given dataset by utilizing the power of DL classifiers. The performance of this model is also evaluated utilizing evaluation metrics explained in this section. This section provides a detailed and analytical evaluation of the results. To conduct this research, a pre-selected set of instruments and technologies were used in experiments. Python 3.8.8, a programming language with great influence on machine learning, greatly speeds up the process by providing many modules and tools to facilitate sophisticated data processing, analysis, and visualization. The basic conductor is Jupyter Notebook, a development platform well regarded for its exceptional ability to provide the perfect programming environment.

A. EVALUATION METRICS

This research estimates the efficiency of the proposed methodology using many evaluation metrics, such as accuracy and loss. These essential evaluation metrics provide detailed information about the interpretation of the proposed methodology. Accuracy is the first metric often regarded as the cornerstone of performance assessment. It determines the proportion of correctly identified instances using the total number of instances. The equation 6 explains this well. Even though the measure's calculation is simple, it has a significant impact.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

Loss, also known as the cost function or error function, quantifies how well a model's predictions match the actual target values. The goal of training a machine learning model is typically to minimize this loss. The choice of loss function depends on the specific problem and the type of data. Common loss functions include Mean Squared Error (MSE), Cross-Entropy Loss, and Hinge Loss, among others. The formula for loss varies depending on the specific loss function being used. The formula for computing sparse categorical cross-entropy loss is given below. Let N be the number of samples, C be the number of classes, and y_i be the true class label for the i_{th} sample. For a single sample i, the loss is computed as:

$$L_i = -log\left(\frac{e^{p_{y_i}}}{\sum_{j=1}^{c} e^{p_i}}\right) \tag{7}$$

Where: P_j is the predicted probability of class, j for the i_{th} sample. y_i is the true class label for the i_th sample. The overall loss for all N samples are the average of the individual losses:

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i) \tag{8}$$

B. FINDING AND DISCUSSION

The proposed model's implementation result and findings for binary and multiclass are provided in detail. Next, we assess the performance and present an analysis of all the DL methods and a graphical visualization of these models.

1) Result of Myocardial Diseases on Binary Label

Table 2 provides the results of different DL classifiers (RNN, DNN, CNN, and LSTM) on classifying myocardial diseases using binary labels. The evaluation metrics included in this table are accuracy and loss for training and testing datasets. RNN attained a value training accuracy of 96.76%, with a training loss of 0.1095. DNN attained a value training Accuracy of 96.76%, with a training loss of 0.0941. CNN attained a value training Accuracy of 96.76%, with a training loss of 0.1269. LSTM attained a value training accuracy of 96.76%, with a training loss of 0.1024. RNN attained a test accuracy value of 96.76%, with a test loss of 0.1095. DNN attained a test accuracy value of 96.76%, with a test loss of 0.0941. CNN attained a test accuracy value of 96.76%, with a test loss of 0.1269. LSTM attained a test accuracy value of 96.76%, with a test loss of 0.1024. All four classifiers (RNN, DNN, CNN, LSTM) achieved the same accuracy of 96.76% on both the training and testing datasets. The loss values for each classifier during training and testing are also consistent. These results indicate good generalization of the models to unseen data, as the test accuracy is similar to the training accuracy.

TABLE 2: Result of Myocardial Detection on Binary Label without SMOTE

Classifiers	Accuracy (%)	Loss (%)	Test Accuracy (%)	Test Loss (%)
RNN	0.9676	0.1095	0.9676	0.1095
DNN	0.9676	0.0941	0.9676	0.0941
CNN	0.9676	0.1269	0.9676	0.1269
LSTM	0.9676	0.1024	0.9676	0.1024

Table 3 provides the results of different classifiers (RNN, DNN, CNN, and LSTM) on the task of classifying myocardial diseases using binary labels, with the addition of Synthetic Minority Over-sampling Technique (SMOTE). SMOTE is a technique used to address class imbalance in the dataset by generating synthetic samples for the minority class. RNN training accuracy value increased to 98.78%, with a training loss of 0.0424. DNN training accuracy value increased to 99.39%, with a training loss of 0.0252. CNN training accuracy value increased to 99.24%, with a training loss of 0.0362. LSTM training accuracy value increased to 99.24%, with a training loss of 0.0255. RNN testing accuracy value increased to 98.78%, with a test loss of 0.0424. DNN testing accuracy value increased to 99.39%, with a test loss of 0.0252. CNN testing accuracy value increased to 99.24%, with a test loss of 0.0362. LSTM testing accuracy value increased to 99.24%, with a test loss of 0.0255. The results indicate a notable improvement in training and testing accuracies after applying SMOTE. The classifiers now achieve higher accuracy, suggesting that addressing class imbalance through oversampling has positively impacted the models' ability to distinguish between classes. The reduced loss values also indicate better convergence during training.

TABLE 3: Results of Myocardial Detection on Binary Label with SMOTE

Classifiers	Accuracy (%)	Loss (%)	Test Accuracy (%)	Test Loss (%)
RNN	0.9878	0.0424	0.9878	0.0424
DNN	0.9939	0.0252	0.9939	0.0252
CNN	0.9924	0.0362	0.9924	0.0362
LSTM	0.9924	0.0255	0.9924	0.0255

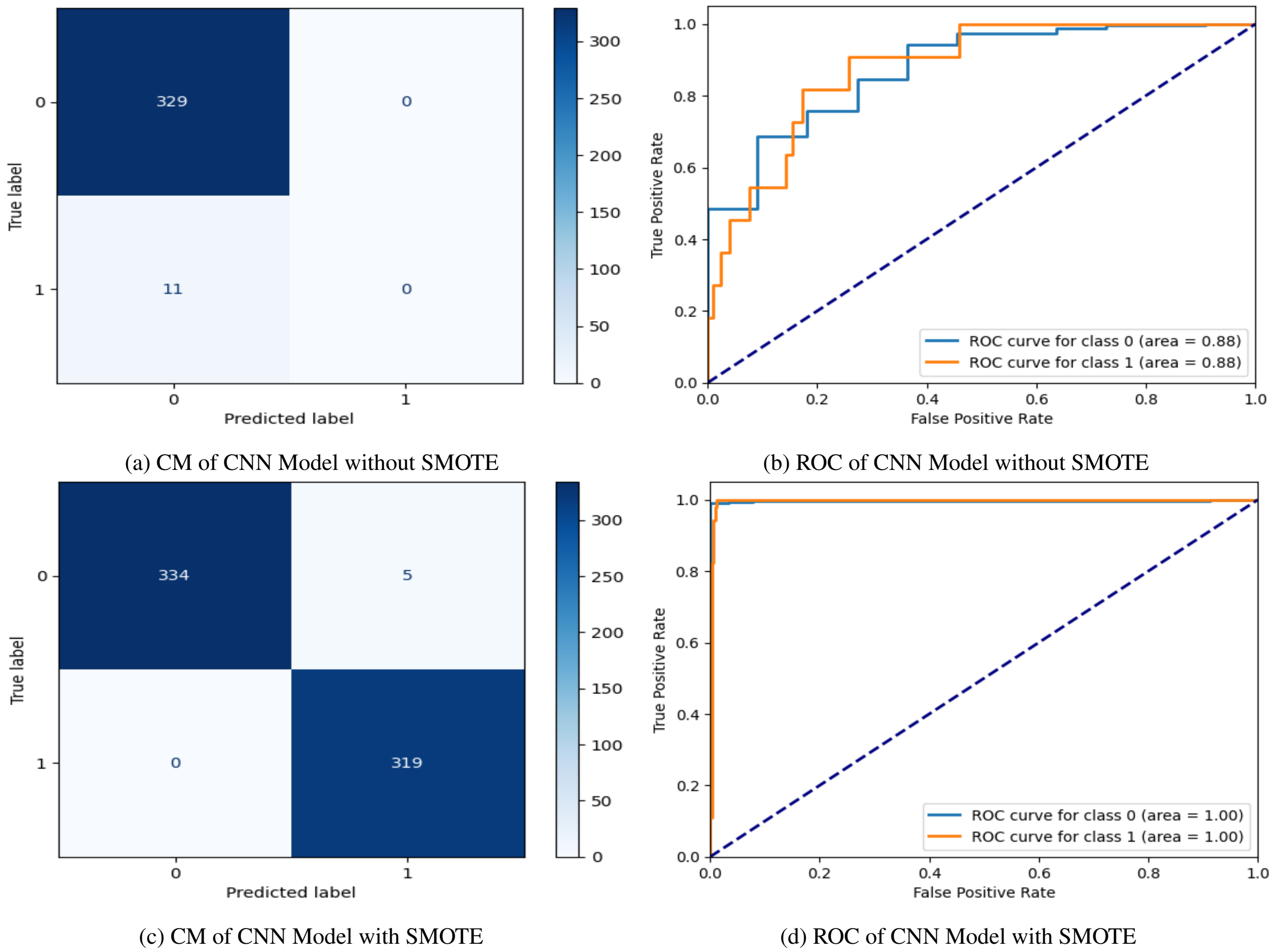
Figure 2 presents graphical visualizations in terms of the Confusion matrix (CM) and recursive operating characteristic curve (ROC) of the Convolutional Neural Network (CNN) model for binary label classification. Figure 2a presents the CM for the CNN model applied without SMOTE. CM Check for the distribution of True Positives(TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). This helps in understanding where the model is making correct or incorrect predictions. The CM typically has four quadrants: True Positive (top-left), False Negative (bottom-left), FP (top-right), and True Negative (bottom-right). Every quadrant displays values in each cell and represents the count of instances corresponding to different classification outcomes. CM operates more effectively since the proposed approach yields more continuous, superior true positive values and fewer FPs.

Figure 2b presents the ROC for the CNN model applied without SMOTE. The ROC curve is a graphical visualization of the model's performance across different classification thresholds. The x-axis represents the FP Rate (FPR), and the y-axis represents the True Positive Rate (TPR) or Sensitivity. The curve illustrates the trade-off between sensitivity and specificity, showing how well the model discriminates between classes. A performance curve that is closest to the upper-left corner is preferable. In this scenario, two labels are used for classification. Label 0 represents the blue line, and label 1 represents the orange line. The blue line shows the ROC curve with an area of 0.88, while the orange line for Class 1 expresses the curve area with the same value of 0.88. Figure 2c presents the CM for the CNN model applied with SMOTE. CM operates more effectively since the proposed approach yields more continuous, superior true positive and negative values with fewer FPs. Figure 2d presents the ROC for the CNN model applied with SMOTE. The blue line shows the ROC curve with an area of 1.00, while the orange line for Class 1 expresses the curve area with the same value of 1.00.

Figure 3 presents graphical visualizations regarding the CM and ROC of the Deep Neural Network (DNN) model for binary label classification. Figure 3a presents the CM for the DNN model applied without SMOTE. CM Check for the distribution of TP, TN, FP and FN. This helps to understand whether the proposed model makes correct or incorrect predictions. The CM typically has four quadrants: TP (top-left), FN (bottom-left), FP (top-right), and TN (bottom-right). Every quadrant displays values in each cell and represents the count of instances corresponding to different classification outcomes. CM operates more effectively since the proposed approach yields more continuous, superior true positive values and fewer FPs.

Figure 3b presents the ROC for the DNN model applied without SMOTE. The ROC curve is a graphical visualization of the model's performance across different classification thresholds. The x-axis shows the False FPR, and the y-axis shows the TPR or Sensitivity. The curve illustrates the trade-off between sensitivity and specificity, showing how well the model discriminates between classes. A performance curve that is closest to the upper-left corner is preferable. In this scenario, two labels are used for classification. The blue line represents the ROC curve with an area of 0.88, while the orange line for Class 1 expresses the curve area with the same value of 0.88. Figure 3c presents the CM for the DNN model applied with SMOTE. CM operates more effectively since the proposed approach yields more continuous, superior true positive and negative values with fewer FPs. Figure 3d presents the ROC for the DNN model applied with SMOTE. The blue line represents the ROC curve with an area of 1.00, while the orange line for Class 1 expresses the curve area with the same value of 1.00.

Figure 4 presents graphical visualizations regarding the CM and ROC of the Recurrent Neural Network (RNN) model for binary label classification. Figure 4a presents the CM for the RNN model applied without SMOTE. CM Check for the distribution of TP, TN, FP and FN. This helps in understanding where the model is making correct or incorrect predictions. The CM typically has four quadrants: TP (top-left), FN (bottom-left), FP (top-right), and TN (bottom-right). Every quadrant displays values in each cell and represents the count of instances corresponding to different classification outcomes. CM operates more effectively since the proposed approach yields more continuous, superior true positive val-



(a) CM of CNN Model without SMOTE

(b) ROC of CNN Model without SMOTE

(c) CM of CNN Model with SMOTE

(d) ROC of CNN Model with SMOTE

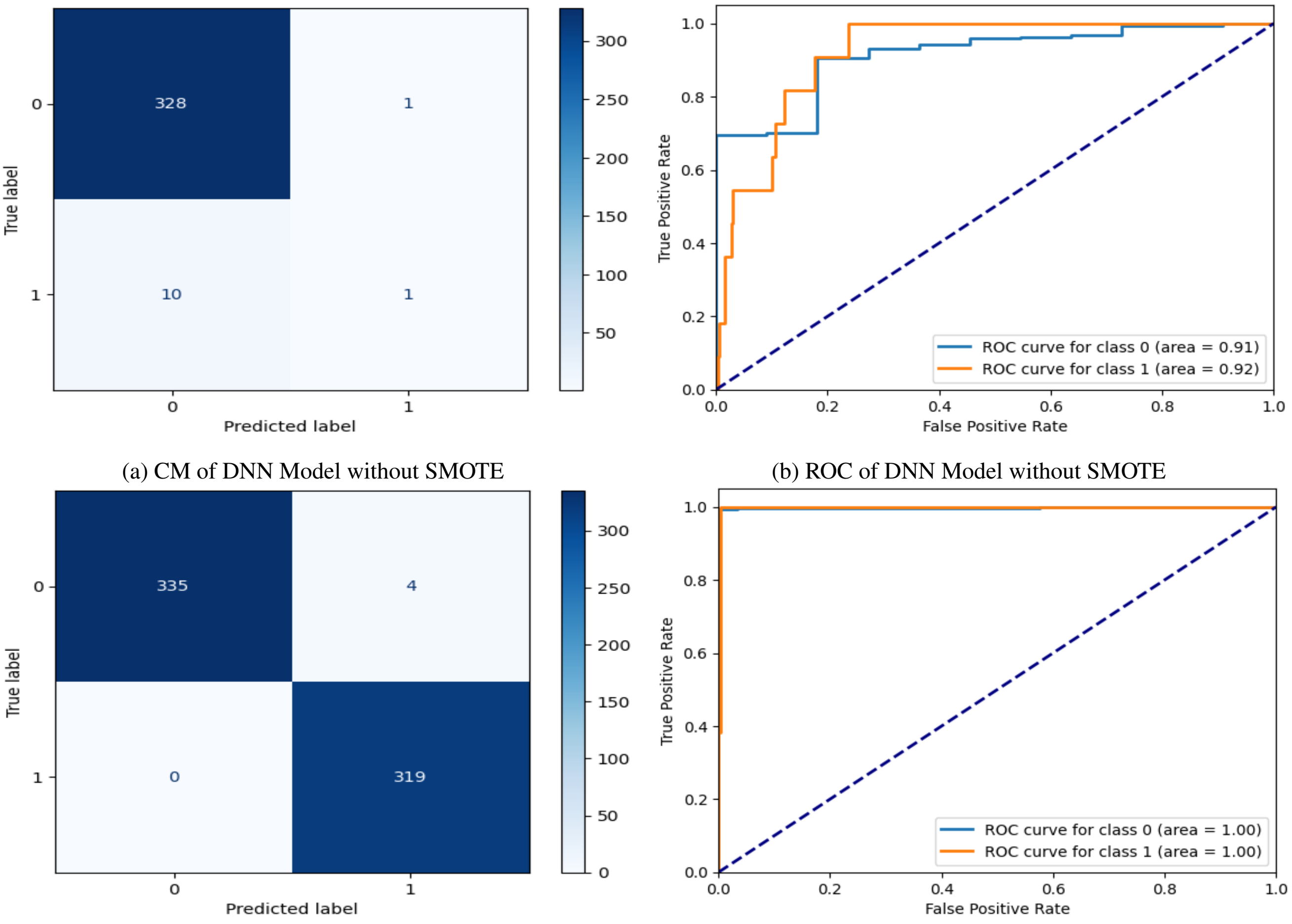
FIGURE 2: Graphical Visualization of CNN Model for binary label

ues and fewer FPs.

Figure 4b presents the ROC for the RNN model applied without SMOTE. The ROC curve is a graphical visualization of the model's performance across different classification thresholds. The x-axis shows the FPR and the y-axis shows the TPR or Sensitivity. The curve illustrates the trade-off between sensitivity and specificity, showing how well the model discriminates between classes. A performance curve that is closest to the upper-left corner is preferable. In this scenario, two labels are used for classification. The blue line represents the ROC curve with an area of 0.88, while the orange line for Class 1 expresses the curve area with the same value of 0.88. Figure 4c presents the CM for the RNN model applied with SMOTE. CM operates more effectively since the proposed approach yields more continuous, superior true positive and negative values with fewer FPs. Figure 4d presents the ROC for the RNN model applied with SMOTE. The blue line represents the ROC curve with an area of 1.00, while the orange line for Class 1 expresses the curve area with the same value of 1.00.

Figure 5 presents graphical visualizations regarding the CM and ROC of the Long-Short Term Memory (LSTM) model for binary label classification. Figure 5a presents the CM for the LSTM model applied without SMOTE. CM Check for the distribution of TP, TN, FP and FN. This helps in understanding where the model is making correct or incorrect predictions. The CM typically has four quadrants: TP (top-left), FN (bottom-left), FP (top-right), and TN (bottom-right). Every quadrant displays values in each cell and represents the count of instances corresponding to different classification outcomes. CM operates more effectively since the proposed approach yields more continuous, superior true positive values and fewer FPs.

Figure 5b presents the ROC for the LSTM model applied without SMOTE. The ROC curve is a graphical visualization of the model's performance across different classification thresholds. The x-axis shows the FPR and the y-axis shows the TPR or Sensitivity. The curve illustrates the trade-off between sensitivity and specificity, showing how well the model discriminates between classes. A performance curve that is closest to the upper-left corner is preferable. In this scenario, two labels are used for classification. The blue line represents the ROC curve with an area of 0.88, while the orange line for Class 1 expresses the curve area with the same value of 0.88.



(a) CM of DNN Model without SMOTE

(b) ROC of DNN Model without SMOTE

(c) CM of DNN Model with SMOTE

(d) ROC of DNN Model with SMOTE

FIGURE 3: Graphical Visualization of DNN Model for binary label

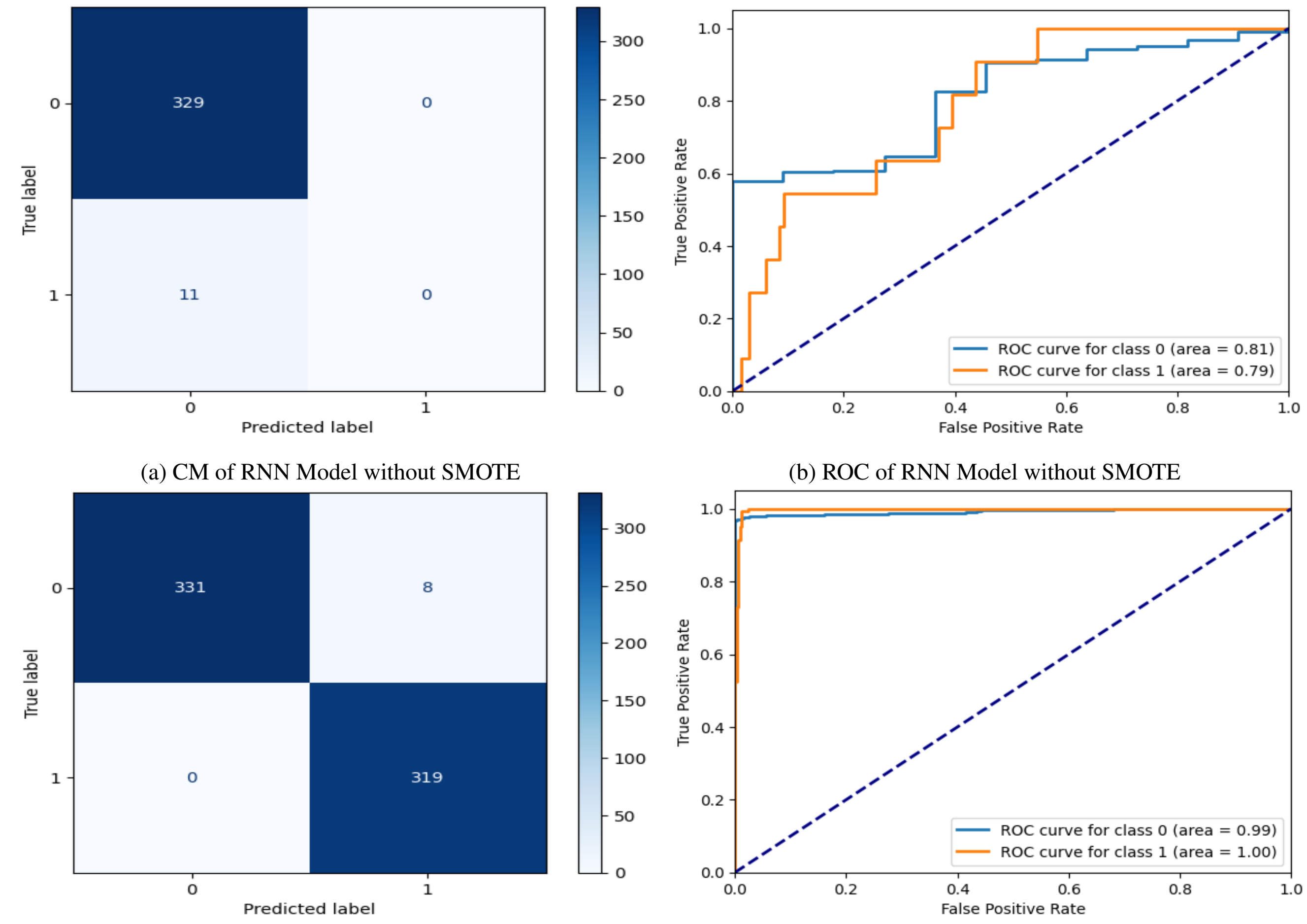
Figure 5c presents the CM for the LSTM model applied with SMOTE. CM operates more effectively since the proposed approach yields more continuous, superior true positive and negative values with fewer FPs. Figure 5d presents the ROC for the LSTM model applied with SMOTE. The blue line represents the ROC curve with an area of 1.00, while the orange line for Class 1 expresses the curve area with the same value of 1.00.

2) Results of Myocardial Diseases on Multi-Label

TABLE 4: Results of Myocardial Diseases on Multi-Label without SMOTE

Classifiers	Accuracy	Loss	Test Accuracy	Test Loss
RNN	0.8765	0.4201	0.8765	0.4201
DNN	0.8765	0.4131	0.8765	0.4131
CNN	0.8824	0.4446	0.8824	0.4446
LSTM	0.8794	0.4142	0.8794	0.4142

Table 4 provides the results of different DL classifiers (RNN, DNN, CNN, and LSTM) on classifying myocardial diseases using multi-labels without applying the SMOTE technique. The evaluation metrics included in this table are accuracy and loss for training and testing datasets. RNN attained a value training accuracy of 87.65%, with a training loss of 0.4201. DNN attained a value training Accuracy of 87.65%, with a training loss of 0.4131. CNN attained a value training Accuracy of 88.24%, with a training loss of 0.4446. LSTM attained a value training accuracy of 87.94%, with a training loss of 0.4142. RNN attained a test accuracy value of 87.65%, with a test loss of 0.4201. DNN attained a test accuracy value of 87.65%, with a test loss of 0.4131. CNN attained a test accuracy value of 88.24%, with a test loss of 0.4446. LSTM attained a test accuracy value of 87.94%, with a test loss of 0.4142. All four classifiers (RNN, DNN, CNN, LSTM) have similar accuracy values for training and testing datasets, suggesting they are generalizing reasonably well. The loss values for each classifier during training and testing are also consistent. These results might indicate good generalization of the models to unseen data, as the test accuracy is similar to the training accuracy. The results show that the classifiers. However, the accuracy is around 87-88%, which might indicate a chance for improvement. The loss values are relatively high, indicating that there might be a chance for improvement in terms of convergence during training.



(a) CM of RNN Model without SMOTE

(b) ROC of RNN Model without SMOTE

(c) CM of RNN Model with SMOTE

(d) ROC of RNN Model with SMOTE

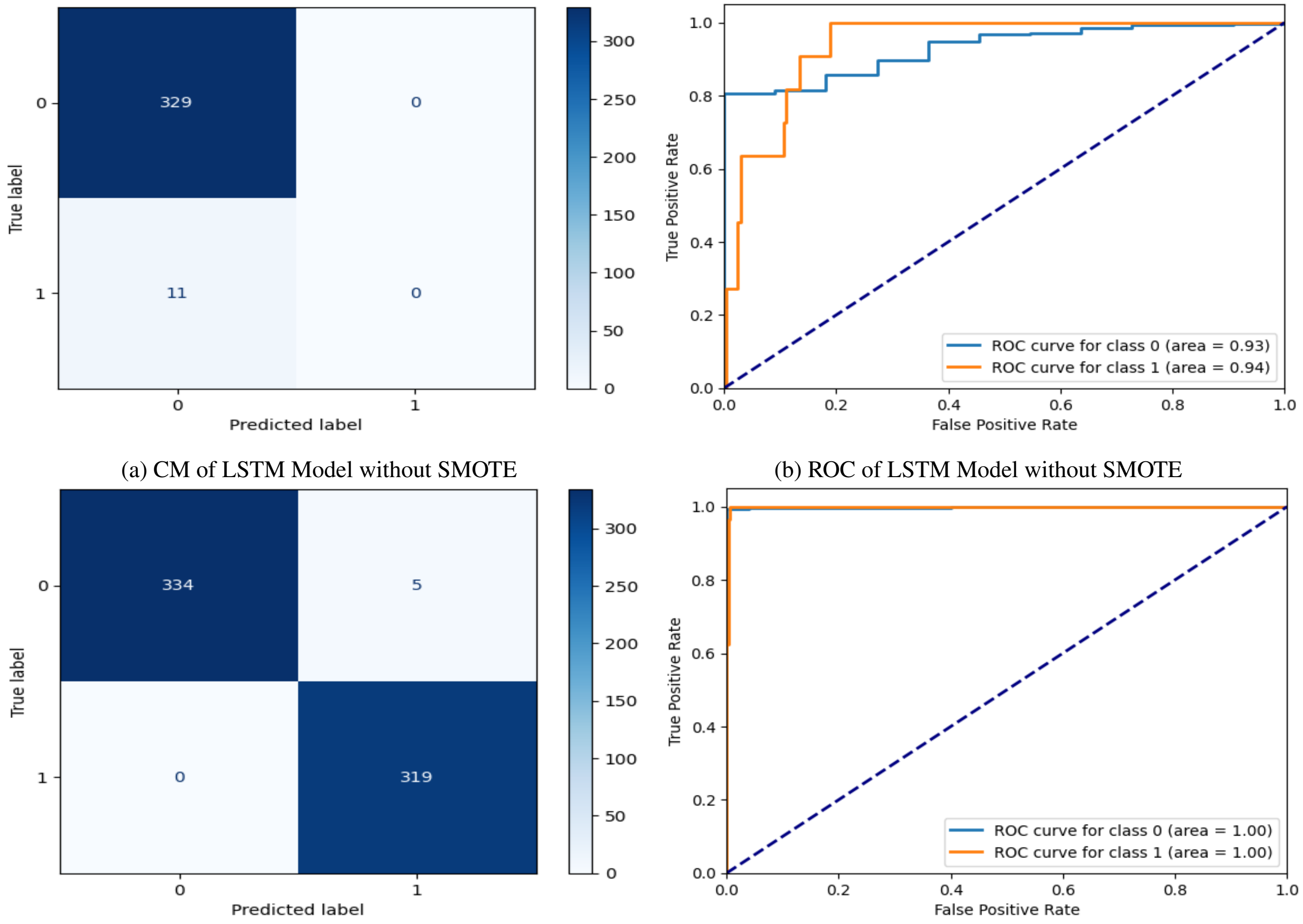
FIGURE 4: Graphical Visualization of RNN Model for Binary Label

TABLE 5: Result of Myocardial Diseases on Multi-Label with SMOTE

Classifiers	Accuracy	Loss	Test Accuracy	Test Loss
RNN	0.9930	0.0242	0.9930	0.0242
DNN	0.9974	0.0115	0.9974	0.0115
CNN	0.9952	0.0305	0.9952	0.0305
LSTM	0.9956	0.0227	0.9956	0.0227

Table 5 provides the results of different classifiers (RNN, DNN, CNN, and LSTM) on classifying myocardial diseases using multi-labels, with the addition of applying the SMOTE technique for class imbalance problems. SMOTE is a technique used to address class imbalance in the dataset by generating synthetic samples for the minority class. RNN training accuracy value increased to 99.30%, with a training loss of 0.0242. DNN training accuracy value increased to 99.74%, with a training loss of 0.0115. CNN training accuracy value increased to 99.52%, with a training loss of 0.0305. LSTM training accuracy value increased to 99.56%, with a training loss of 0.0227. RNN testing accuracy value increased to 99.30%, with a test loss of 0.0242. DNN testing accuracy value increased to 99.74%, with a test loss of 0.0115. CNN testing accuracy value increased to 99.52%, with a test loss of 0.0305. LSTM testing accuracy value increased to 99.56%, with a test loss of 0.0227. All classifiers indicate a significant improvement in both training and testing accuracies after applying SMOTE. The classifiers now achieve very high accuracy, suggesting that addressing class imbalance through oversampling has positively impacted the models' ability to classify myocardial diseases with multiple labels. The reduced loss values also indicate better convergence during training. These findings suggest that SMOTE has effectively addressed the challenges of class imbalance in the multi-label classification task, resulting in highly accurate and well-generalized models.

Figure 6 presents graphical visualizations regarding the CM and ROC of the CNN model for multi-label classification. Figure 6a presents the CM for the CNN model applied without SMOTE. CM Check for the distribution of TP, TN, FP and FN. This helps in understanding where the model is making correct or incorrect predictions. The matrix rows show the actual class labels, while the columns correspond to the predicted ones. The incorrect instances are centered on the diagonal components, whereas the correctly recognized



(a) CM of LSTM Model without SMOTE

(b) ROC of LSTM Model without SMOTE

(c) CM of LSTM Model with SMOTE

(d) ROC of LSTM Model with SMOTE

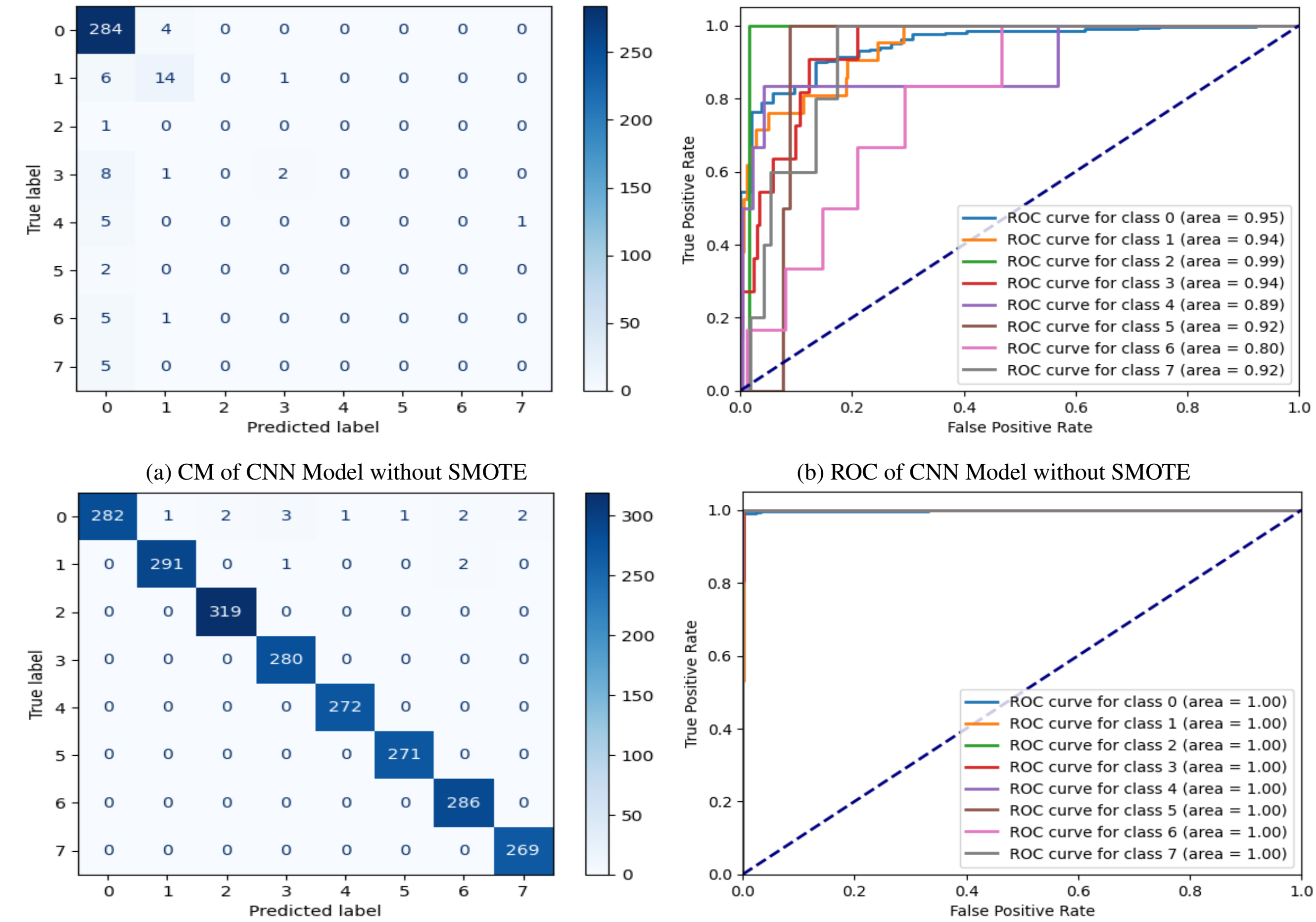
FIGURE 5: Graphical Visualization of LSTM Model for Binary Classification

instances are positioned along the diagonal. All multi-labels of the dataset from class 0 to 7, class 0 predicted 284 instances, class 1 predicted 14, and the rest from 2 to 7 classes predicted 0 instances by CNN.

Figure 6b presents the ROC for the CNN model applied without SMOTE. The ROC curve is a graphical visualization of the model's performance across different classification thresholds. The x-axis shows the FPR and the y-axis shows the TPR or Sensitivity. The curve illustrates the trade-off between sensitivity and specificity, showing how well the model discriminates between classes. A performance curve that is closest to the upper-left corner is preferable. In this scenario, multi-labels are used for classification. Label 0 represents the blue line, label 1 represents the orange line, label 2 shows the green line, label 3 shows the red line, label 4 represents the purple line, label 5 shows the brown line, label 6 represents the pink line, and label 7 represent the grey line. The blue line represents the ROC curve with an area of 0.95, the orange line for Class 1 expresses the curve area with the same value of 0.94, the green line for Class 2 expresses the curve area with the same value of 0.99, the red line for Class 3 expresses the curve area with the same value of 0.94, the purple line for Class 4 expresses the curve area with the same value of 0.89, the brown line for Class 5 expresses the curve area with the same value of 0.92, the pink line for Class 6 expresses the curve area with the same value of 0.94 and the grey line for Class 7 expresses the curve area with the same value of 0.92.

Figure 6c presents the CM for the CNN model applied with SMOTE. CM operates more effectively after applying SMOTE since the proposed approach yields more continuous, superior TP and TN values with fewer FP and FN values for all classes from 0 to 7. Figure 6d presents the ROC for the CNN model applied with SMOTE. After applying SMOTE, the ROC values of all classes 0 to 7 improve performance, and all express the curve area with the same value of 1.00.

Figure 7 presents graphical visualizations regarding the CM and ROC of the DNN model for multi-label classification. Figure 7a presents the CM for the DNN model applied without SMOTE.CM Check for the distribution of TP, TN, FP and FN. This helps in understanding where the model is making correct or incorrect predictions. The incorrect instances are centered on the diagonal components, whereas the correctly recognized instances are positioned along the



(a) CM of CNN Model without SMOTE

(b) ROC of CNN Model without SMOTE

(c) CM of CNN Model with SMOTE

(d) ROC of CNN Model with SMOTE

FIGURE 6: Graphical Visualization of CNN Model for Multi-Label Classification

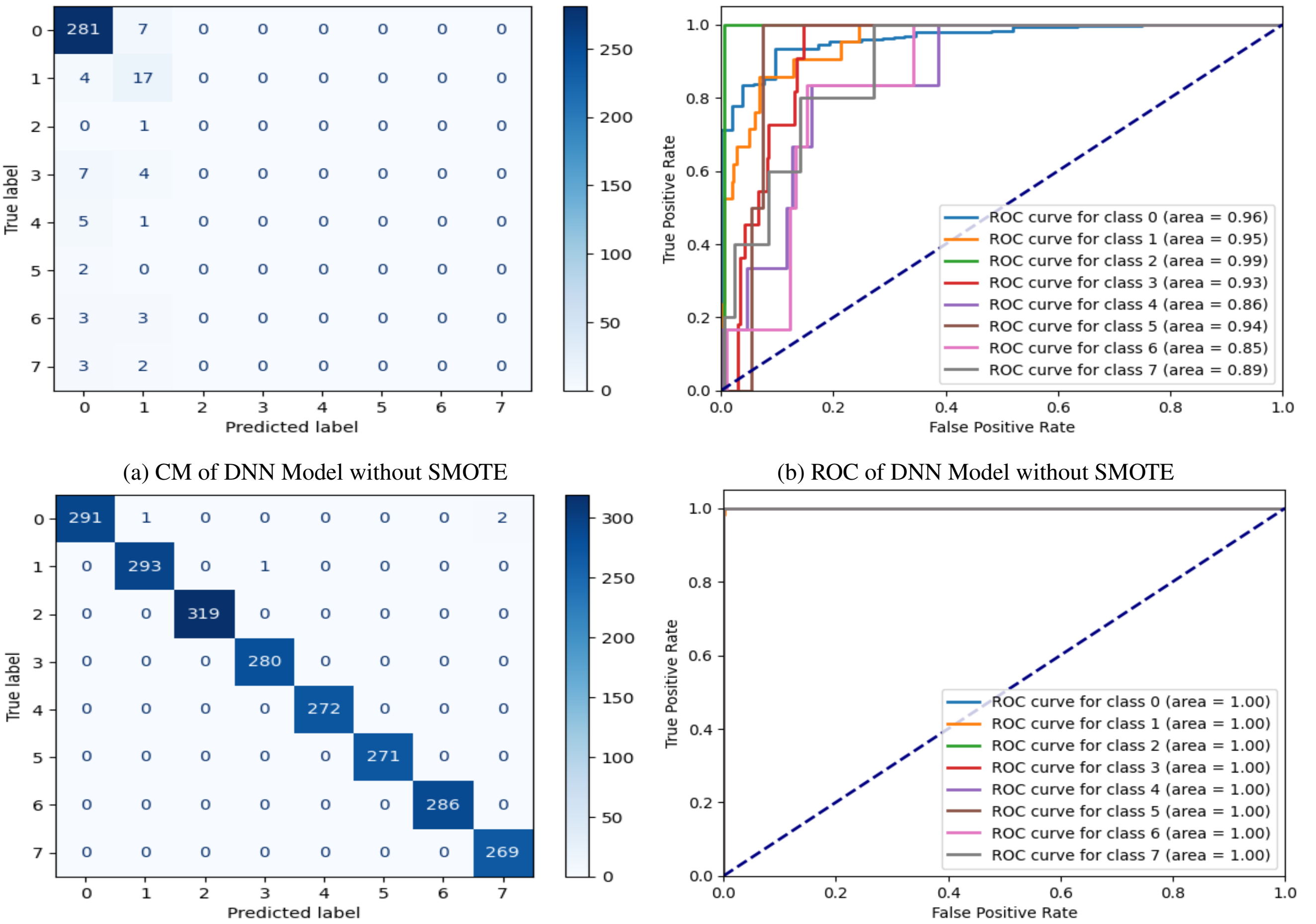
diagonal. All multi-labels of the dataset from class 0 to 7, class 0 predicted 281 instances, class 1 predicted 17, and the rest from 2 to 7 classes predicted 0 instances by CNN.

Figure 7b presents the ROC for the DNN model applied without SMOTE. The ROC curve is a graphical visualization of the model's performance across different classification thresholds. The x-axis shows the FPR and the y-axis shows the TPR or Sensitivity. The curve illustrates the trade-off between sensitivity and specificity, showing how well the model discriminates between classes. A performance curve that is closest to the upper-left corner is preferable. In this scenario, multi-labels are used for classification. The blue line represents the ROC curve with an area of 0.95, the orange line for Class 1 expresses the curve area with the same value of 0.94, the green line for Class 2 expresses the curve area with the same value of 0.99, the red line for Class 3 expresses the curve area with the same value of 0.94, the purple line for Class 4 expresses the curve area with the same value of 0.89, the brown line for Class 5 expresses the curve area with the same value of 0.92, the pink line for Class 6 expresses the curve area with the same value of 0.94 and the grey line for Class 7 expresses the curve area with the same value of 0.92.

Figure 7c presents the CM for the DNN model applied with SMOTE. CM operates more effectively after applying SMOTE since the proposed approach yields more continuous, superior TP and TN values with fewer FP and FN values for all classes from 0 to 7. Figure 7d presents the ROC for the DNN model applied with SMOTE. After applying SMOTE, the ROC values of all classes 0 to 7 improve performance, and all express the curve area with the same value of 1.00.

Figure 8 presents graphical visualizations regarding the CM and ROC of the RNN model for multi-label classification. Figure 8a presents the CM for the RNN model applied without SMOTE. CM Check for the distribution of TP, TN, FP and FN. This helps in understanding where the model is making correct or incorrect predictions. The incorrect instances are centered on the diagonal components, whereas the correctly recognized instances are positioned along the diagonal. All multi-labels of the dataset from class 0 to 7, class 0 predicted 284 instances, class 1 predicted 14, and the rest from 2 to 7 classes predicted 0 instances by CNN.

Figure 8b presents the ROC for the RNN model applied without SMOTE. The ROC curve is a graphical visualization of the model's performance across different classification



(a) CM of DNN Model without SMOTE

(b) ROC of DNN Model without SMOTE

(c) CM of DNN Model with SMOTE

(d) ROC of DNN Model with SMOTE

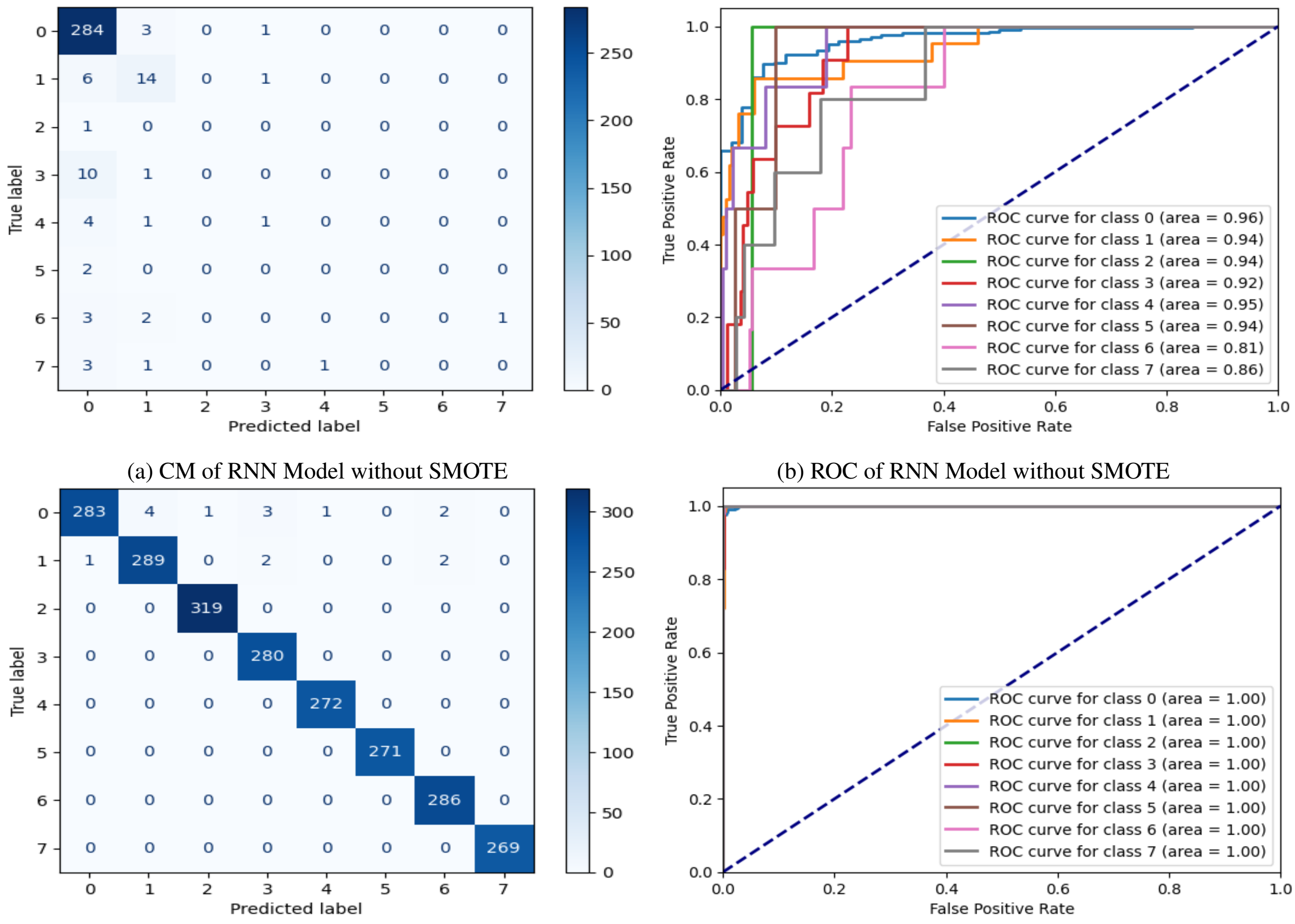
FIGURE 7: Graphical Visualization of DNN Model for Multi-Label

thresholds. The x-axis shows the FPR and the y-axis shows the TPR or Sensitivity. The curve illustrates the trade-off between sensitivity and specificity, showing how well the model discriminates between classes. A performance curve that is closest to the upper-left corner is preferable. In this scenario, multi-labels are used for classification. The blue line represents the ROC curve with an area of 0.95, the orange line for Class 1 expresses the curve area with the same value of 0.94, the green line for Class 2 expresses the curve area with the same value of 0.99, the red line for Class 3 expresses the curve area with the same value of 0.94, the purple line for Class 4 expresses the curve area with the same value of 0.89, the brown line for Class 5 expresses the curve area with the same value of 0.92, the pink line for Class 6 expresses the curve area with the same value of 0.94 and the grey line for Class 7 expresses the curve area with the same value of 0.92.

Figure 8c presents the CM for the RNN model applied with SMOTE. CM operates more effectively after applying SMOTE since the proposed approach yields more continuous, superior TP and TN values with fewer FP and FN values for all classes from 0 to 7. Figure 8d presents the ROC for the RNN model applied with SMOTE. After applying SMOTE, the ROC values of all classes 0 to 7 improve performance, and all express the curve area with the same value of 1.00.

Figure 9 presents graphical visualizations regarding the CM and ROC of the LSTM model for multi-label classification. Figure 9a presents the CM for the LSTM model applied without SMOTE. CM Check for the distribution of TP, TN, FP and FN. This helps in understanding where the model is making correct or incorrect predictions. The incorrect instances are centered on the diagonal components, whereas the correctly recognized instances are positioned along the diagonal. All multi-labels of the dataset from class 0 to 7, class 0 predicted 285 instances, class 1 predicted 14, and the rest from 2 to 7 classes predicted 0 instances by CNN.

Figure 9b presents the ROC for the LSTM model applied without SMOTE. The ROC curve is a graphical visualization of the model's performance across different classification thresholds. The x-axis shows the FPR and the y-axis shows the TPR or Sensitivity. The curve illustrates the trade-off between sensitivity and specificity, showing how well the model discriminates between classes. A performance curve that is closest to the upper-left corner is preferable. In this scenario, multi-labels are used for classification. The blue



(a) CM of RNN Model without SMOTE

(b) ROC of RNN Model without SMOTE

(c) CM of RNN Model with SMOTE

(d) ROC of RNN Model with SMOTE

FIGURE 8: Graphical Visualization of RNN Model for Multi-Label

line represents the ROC curve with an area of 0.95, the orange line for Class 1 expresses the curve area with the same value of 0.94, the green line for Class 2 expresses the curve area with the same value of 0.99, the red line for Class 3 expresses the curve area with the same value of 0.94, the purple line for Class 4 expresses the curve area with the same value of 0.89, the brown line for Class 5 expresses the curve area with the same value of 0.92, the pink line for Class 6 expresses the curve area with the same value of 0.94 and the grey line for Class 7 expresses the curve area with the same value of 0.92.

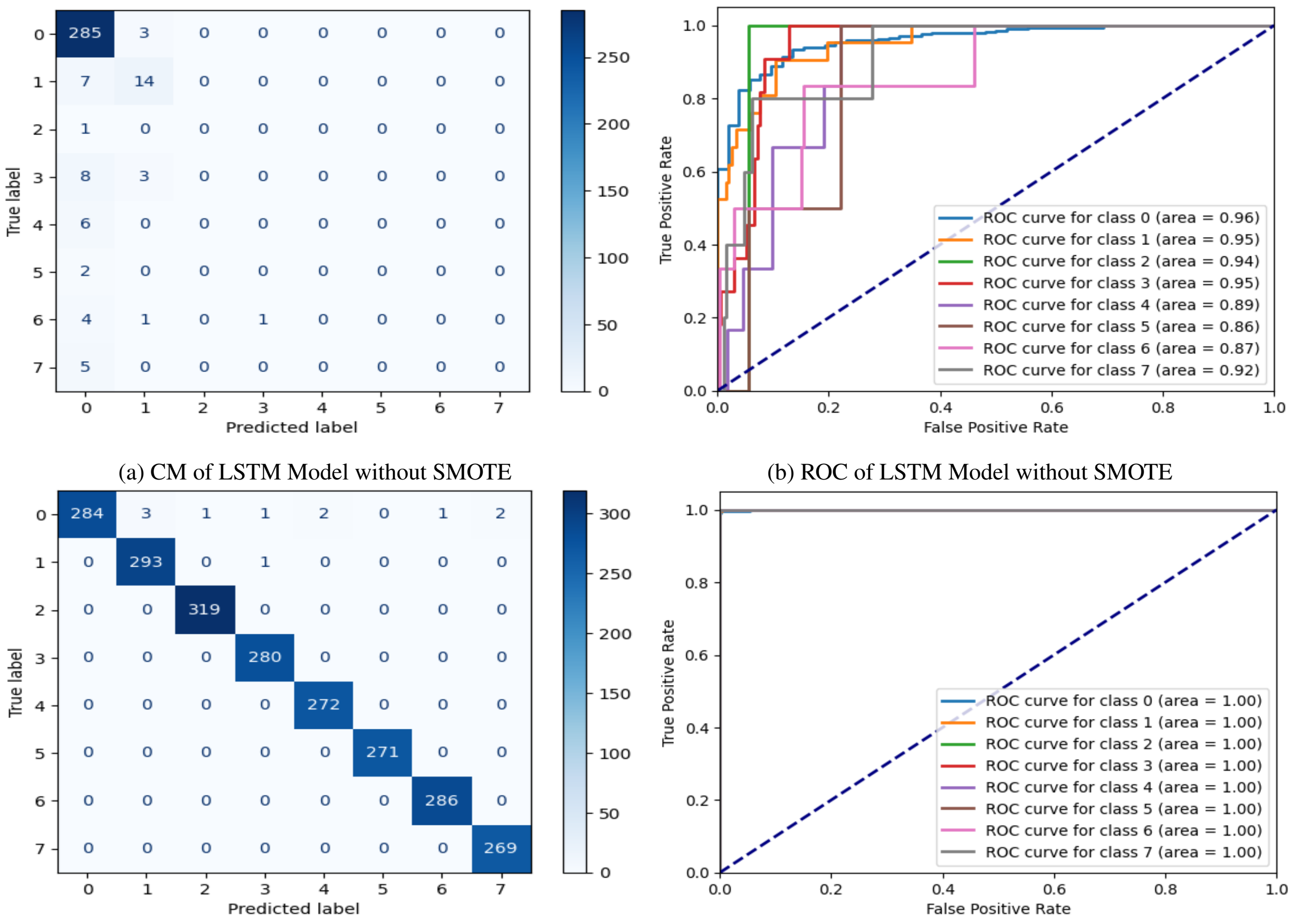
Figure 9c presents the CM for the LSTM model applied with SMOTE. CM operates more effectively after applying SMOTE since the proposed approach yields more continuous, superior TP and TN values with fewer FP and FN values for all classes from 0 to 7. Figure 9d presents the ROC for the LSTM model applied with SMOTE. After applying SMOTE, the ROC values of all classes 0 to 7 improve performance, and all express the curve area with the same value of 1.00.

TABLE 6: Comparison of proposed approach

Authors	Classifiers	Classes	Accuracy	Loss
[13]	LightGBM	Binary-Lable	84.60%	-
[21]	LR	Multi-Lable	88.67%	0.056
[27]	GB	Binary-Lable	98.79%	-
Proposed Model	DNN	Binar-Lable	99.39%	0.0252
Proposed Model	DNN	Multi-Lable	99.74%	0.0115

3) Comparison of Proposed Approach with Previous Techniques

Table 6 compares the proposed approach with previous techniques regarding classifiers used, class types, classification, accuracy, and loss. The approach proposed by [13] utilized the LightGBM classifier for binary-label classification and achieved an accuracy of 84.60%. The approach proposed by [21] used logistic regression (LR) for multi-label classification and achieved an accuracy of 88.67% with a loss of 0.056. The approach proposed by [27] employed gradient boosting (GB) for binary-label classification and achieved an accuracy of 98.79%. Ultimately, the model proposed in this research utilized a deep neural network (DNN) for binary and multi-label classification tasks. It achieved higher accuracies



(a) CM of LSTM Model without SMOTE

(b) ROC of LSTM Model without SMOTE

(c) CM of LSTM Model with SMOTE

(d) ROC of LSTM Model with SMOTE

FIGURE 9: Graphical Visualization of LSTM Model for Multi-Label

of 99.39% and 99.74%, respectively, with lower loss values of 0.0252 and 0.0115. The proposed model outperforms the previous techniques in terms of both accuracy and loss across both binary-label and multi-label classification tasks. Additionally, the proposed model achieves higher accuracy in both classification tasks than the other classifiers used in the previous approaches.

V. CONCLUSION AND FUTURE SCOPE

MI is a leading global reason of mortality. Because of that, thousands of people lost their lives every single day. A patient's life can be saved by receiving the right medical care; failing to do so might have terrible consequences. Considering this, we suggested a model that predicts MI in patients using DL classifiers (RNN, CNN, DNN, and LSTM). We used a dataset that included the medical records of 1700 MI patients. In most statistics about medicine and healthcare, class imbalance is a serious issue. We used the SMOTE sampling approach to balance the data, which performed better. Models are first trained using imbalanced data, and then the data is balanced using a sampling strategy. Another preprocessing technique uses z-score normalization to normalize the data, and data is split into training and test sets. After employing SMOTE, the proposed method improved accuracy and loss scores. The method was evaluated on unbalanced datasets to verify the proposed approach's effectiveness. This demonstrates the effectiveness of the proposed approach, which is useful for handling unbalanced classification issues. The research paper suggests a deep learning-based technique for detecting myocardial infractions (MI), yet there are a number of noteworthy drawbacks. First off, the diversity of real-world populations might need to be accurately reflected by the dataset size of 1700 MI patients. Second, even though the research discusses class imbalance, it is worthwhile to investigate methods other than oversampling and undersampling. Thirdly, model performance may be impacted by the scant attention paid to feature selection and preprocessing. Furthermore, a barrier to the clinical acceptability of deep learning models is their interpretability. Finally, for real-world applicability, comprehensive clinical validation and deployment are essential. In the future, we want to utilize datasets from various geographic areas, varying preprocessing methods, and various deep-learning classifiers for the proposed model.

ACKNOWLEDGEMENT

Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R503), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

REFERENCES

[1] J. Miah, D. M. Ca, M. A. Sayed, E. R. Lipu, F. Mahmud, and S. Y. Arafat, “Improving cardiovascular disease prediction through comparative analysis of machine learning models: A case study on myocardial infarction,” in 2023 15th International Conference on Innovations in Information Technology (IIT), pp. 49–54, IEEE, 2023.

[2] A. K. Dohare, V. Kumar, and R. Kumar, “Detection of myocardial infarction in 12 lead ecg using support vector machine,” Applied Soft Computing, vol. 64, pp. 138–147, 2018.

[3] C. F. Sing, J. H. Stengård, and S. L. Kardia, “Genes, environment, and cardiovascular disease,” Arteriosclerosis, thrombosis, and vascular biology, vol. 23, no. 7, pp. 1190–1196, 2003.

[4] U. R. Acharya, H. Fujita, V. K. Sudarshan, S. L. Oh, M. Adam, J. E. Koh, J. H. Tan, D. N. Ghista, R. J. Martis, C. K. Chua, et al., “Automated detection and localization of myocardial infarction using electrocardiogram: a comparative study of different leads,” Knowledge-Based Systems, vol. 99, pp. 146–156, 2016.

[5] K. Jafarian, V. Vahdat, S. Salehi, and M. Mobin, “Automating detection and localization of myocardial infarction using shallow and end-to-end deep neural networks,” Applied Soft Computing, vol. 93, p. 106383, 2020.

[6] A. H. Association et al., “Cardiovascular disease: A costly burden for america projections through 2035,” 2017.

[7] Z. Chen, A. Lalande, M. Salomon, T. Decourselle, T. Pommier, A. Qayyum, J. Shi, G. Perrot, and R. Couturier, “Automatic deep learning-based myocardial infarction segmentation from delayed enhancement mri,” Computerized Medical Imaging and Graphics, vol. 95, p. 102014, 2022.

[8] R. Muraki, A. Teramoto, K. Sugimoto, K. Sugimoto, A. Yamada, and E. Watanabe, “Automated detection scheme for acute myocardial infarction using convolutional neural network and long short-term memory,” PloS one, vol. 17, no. 2, p. e0264002, 2022.

[9] A. K. Chattopadhyay and S. Chattopadhyay, “Virdocd: A virtual doctor to predict dengue fatality,” Expert Systems, vol. 39, no. 1, p. e12796, 2022.

[10] S. Chattopadhyay, A. Chattopadhyay, and E. Aifantis, “Predicting case fatality of dengue epidemic: Statistical machine learning towards a virtual doctor,” Journal of Nanotechnology in Diagnosis and Treatment, vol. 7, pp. 10–24, 2021.

[11] S. Padhy and S. Dandapat, “Third-order tensor based analysis of multilead ecg for classification of myocardial infarction,” Biomedical Signal Processing and Control, vol. 31, pp. 71–78, 2017.

[12] L. V. Romaguera, F. P. Romero, C. F. F. Costa Filho, and M. G. F. Costa, “Myocardial segmentation in cardiac magnetic resonance images using fully convolutional neural networks,” Biomedical Signal Processing and Control, vol. 44, pp. 48–57, 2018.

[13] R. H. Khan, J. Miah, S. A. A. Nipun, and M. Islam, “A comparative study of machine learning classifiers to analyze the precision of myocardial infarction prediction,” in 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0949–0954, IEEE, 2023.

[14] H. Abdeltawab, F. Khalifa, F. Taher, N. S. Alghamdi, M. Ghazal, G. Beache, T. Mohamed, R. Keynton, and A. El-Baz, “A deep learning-based approach for automatic segmentation and quantification of the left ventricle from cardiac cine mr images,” Computerized medical imaging and graphics, vol. 81, p. 101717, 2020.

[15] A. S. Lundervold and A. Lundervold, “An overview of deep learning in medical imaging focusing on mri,” Zeitschrift für Medizinische Physik, vol. 29, no. 2, pp. 102–127, 2019.

[16] R. Karim, P. Bhagirath, P. Claus, R. J. Housden, Z. Chen, Z. Karimaghaloo, H.-M. Sohn, L. L. Rodríguez, S. Vera, X. Albà, et al., “Evaluation of state-of-the-art segmentation algorithms for left ventricle infarct from late gadolinium enhancement mr images,” Medical image analysis, vol. 30, pp. 95–107, 2016.

[17] M. Khened, V. A. Kollerathu, and G. Krishnamurthi, “Fully convolutional multi-scale residual densenets for cardiac segmentation and automated cardiac diagnosis using ensemble of classifiers,” Medical image analysis, vol. 51, pp. 21–45, 2019.

[18] M. Chen, L. Fang, Q. Zhuang, and H. Liu, “Deep learning assessment of myocardial infarction from mr image sequences,” Ieee Access, vol. 7, pp. 5438–5446, 2019.

[19] Z.-Y. Shu, S.-J. Cui, Y.-Q. Zhang, Y.-Y. Xu, S.-C. Hung, L.-P. Fu, P.-P. Pang, X.-Y. Gong, and Q.-Y. Jin, “Predicting chronic myocardial ischemia using ccta-based radiomics machine learning nomogram,” Journal of Nuclear Cardiology, vol. 29, no. 1, pp. 262–274, 2022.

[20] F. Commandeur, P. J. Slomka, M. Goeller, X. Chen, S. Cadet, A. Razipour, P. McElhinney, H. Gransar, S. Cantu, R. J. Miller, et al., “Machine learning to predict the long-term risk of myocardial infarction and cardiac death based on clinical risk, coronary calcium, and epicardial adipose tissue: a prospective study,” Cardiovascular research, vol. 116, no. 14, pp. 2216–2225, 2020.

[21] Z. Chen, J. Shi, T. Pommier, Y. Cottin, M. Salomon, T. Decourselle, A. Lalande, and R. Couturier, “Prediction of myocardial infarction from patient features with machine learning,” Frontiers in cardiovascular medicine, vol. 9, p. 754609, 2022.

[22] L. Marien, M. Valizadeh, W. Zu Castell, C. Nam, D. Rechid, A. Schneider, C. Meisinger, J. Linseisen, K. Wolf, and L. Bouwer, “Machine learning models to predict myocardial infarctions from past climatic and environmental conditions,” Natural Hazards and Earth System Sciences Discussions, vol. 2022, pp. 1–36, 2022.

[23] D. Mandair, P. Tiwari, S. Simon, K. L. Colborn, and M. A. Rosenberg, “Prediction of incident myocardial infarction using machine learning applied to harmonized electronic health record data,” BMC medical informatics and decision making, vol. 20, no. 1, pp. 1–10, 2020.

[24] C. Xiao, Y. Guo, K. Zhao, S. Liu, N. He, Y. He, S. Guo, and Z. Chen, “Prognostic value of machine learning in patients with acute myocardial infarction,” Journal of cardiovascular development and disease, vol. 9, no. 2, p. 56, 2022.

[25] A. R. Sulthana and A. Jaithunbi, “Varying combination of feature extraction and modified support vector machines based prediction of myocardial infarction,” Evolving systems, pp. 1–18, 2022.

[26] D. Doudesis, K. K. Lee, J. Boeddinghaus, A. Bularga, A. V. Ferry, C. Tuck, M. T. Lowry, P. Lopez-Ayala, T. Nestelberger, L. Koechlin, et al., “Machine learning for diagnosis of myocardial infarction using cardiac troponin concentrations,” Nature Medicine, pp. 1–10, 2023.

[27] B. Raghukumar, B. Naveen, and D. Lachikarathman, “Predicting the myocardial infarction from predictive analytics through supervised machine learning,” SN Computer Science, vol. 4, no. 4, p. 369, 2023.

[28] K. Kodthalu Shivashankara and R. Sameni, “A synthetic electrocardiogram (ecg) image generation toolbox to facilitate deep learning-based scanned ecg digitization,” arXiv e-prints, pp. arXiv–2307, 2023.

[29] E. Prabhakararao and S. Dandapt, “Multi-label ecg classification using temporal convolutional neural network,” arXiv preprint arXiv:2306.03844, 2023.

[30] A. A. Rawi, M. K. Elbashir, and A. M. Ahmed, “Classification of 27 heart abnormalities using 12-lead ecg signals with combined deep learning techniques,” Bulletin of Electrical Engineering and Informatics, vol. 12, no. 4, pp. 2220–2235, 2023.

[31] “Myocardial infarction complications database.” https://figshare.le.ac.uk/articles/dataset/Myocardial_infarction_complications_Database/12045261/3?file=22803572. (Accessed on 02/07/2024).

[32] A. Newaz, M. S. Mohosheu, and M. A. Al Noman, “Predicting complications of myocardial infarction within several hours of hospitalization using data mining techniques,” Informatics in Medicine Unlocked, vol. 42, p. 101361, 2023.

[33] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” Journal of artificial intelligence research, vol. 16, pp. 321–357, 2002.

[34] H. Henderi, T. Wahyuningsih, and E. Rahwanto, “Comparison of min-max normalization and z-score normalization in the k-nearest neighbor (knn) algorithm to test the accuracy of types of breast cancer,” International Journal of Informatics and Information Systems, vol. 4, no. 1, pp. 13–20, 2021.

[35] R. Jozefowicz, W. Zaremba, and I. Sutskever, “An empirical exploration of recurrent network architectures,” in International conference on machine learning, pp. 2342–2350, PMLR, 2015.

[36] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] Y. Chen, T. Luo, S. Liu, S. Zhang, L. He, J. Wang, L. Li, T. Chen, Z. Xu, N. Sun, et al., “Dadiannao: A machine-learning supercomputer,” in 2014 47th Annual IEEE/ACM International Symposium on Microarchitecture, pp. 609–622, IEEE, 2014.

[38] M. Krichen, "Convolutional neural networks: A survey," Computers, vol. 12, no. 8, p. 151, 2023.

[39] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," IEEE transactions on neural networks and learning systems, 2021.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.

[41] K. Xia, J. Huang, and H. Wang, "Lstm-cnn architecture for human activity recognition," IEEE Access, vol. 8, pp. 56855–56866, 2020.

[42] Q. V. Le, N. Jaitly, and G. E. Hinton, "A simple way to initialize recurrent networks of rectified linear units," arXiv preprint arXiv:1504.00941, 2015.

SIDRA ABBAS received her BS degree in the Department of Computer Science, COMSATS University, Islamabad, Pakistan. She has worked as a research assistant at ASET Labs, Islamabad, Pakistan. Her research interests include but are not limited to Artificial Intelligence, Healthcare, Mobile Ubiquitous Computing, computer forensics, machine learning, criminal profiling, software watermarking, intelligent systems, and data privacy protection.

STEPHEN OJO is currently an Assistant Professor in the Department of Electrical and Computer Engineering, College of Engineering, Anderson University, Anderson South Carolina, United States of America. He received a B.Sc (Honours) in Electrical and Electronics Engineering from the Federal University of Technology Akure Nigeria in 2014, an M.Sc from Girne American University, Cyprus in 2017 and a PhD in Information Systems in May 2021 from the same University. Before Joining Anderson University, he was a Lecturer at Girne American University Cyprus, where he taught courses in distributed computing, advanced programming and Electric Circuits. Dr. Ojo also worked as a research scholar at Vodafone Telecommunication Company Cyprus, where he developed a multiplicative-based model for signal propagation in wireless networks. Dr. Ojo was awarded a Mobil and full Ph.D. scholarships throughout his undergraduate program. Dr. Ojo is a full-time faculty member at Anderson University, USA, where he teaches computer programming, electric circuits, machine learning, artificial intelligence in wireless mobile networks, and biomedical applications. He has authored and co-authored several peer-reviewed journals. His research interests are wireless networks, machine learning for wireless mobile networks, machine learning, and AI in biomedical devices.

MOEZ KRICHEN (MEMBER, IEEE) received the Ph.D. degree in computer science from Joseph Fourier University, Grenoble, France, in 2007, and the HDR (Ability to Conduct Research) degree in computer science from the University of Sfax, Sfax, Tunisia, in 2018. He is currently an associate professor of computer science at Al-Baha University and a member of the Research Laboratory on Development and Control of Distributed Applications (REDCAD), Sfax. Currently, he is also working on formal aspects related to deep learning, data mining, blockchain, smart contracts, and optimization. His main research interests include model-based conformance, load, and security testing methodologies for real-time, distributed, and dynamically adaptable systems. Moreover, he works on applying formal methods to several modern technologies like smart cities, the Internet of Things (IoT), smart vehicles, drones, and healthcare systems.

DR. ALAEDDINE MIHOUB received his PhD degree in Computer Science from Grenoble-Alpes University (France, 2015) and his MSc degree in Computer Science from Paris Descartes University (France, 2012). Dr. Alaeddine Mihoub is an Associate Professor of Artificial Intelligence & Machine Learning at the College of Business and Economics at Qassim University (Saudi Arabia). Previously, he was a Research and Data Scientist at Maskott (France) and, earlier, a Postdoctoral Researcher at Orange Labs (Meylan, France). He is the co-author of many innovative patents and several publications in well-ranked international journals and conferences. His research interests include Smart Interactive Systems, Social Signal Processing, and, more largely, Artificial Intelligence, Machine Learning, and Data Science.

MEZNAH ALAMRO is an Assistant Professor in Artificial Intelligence (AI) in the College of Computer and Information Science at Princess Noura University. She holds a Ph.D. in Computer Science from Texas Tech University. Her research interests include Machine Learning, Data Mining, and IoT Security.

LUCIA VILCEKOVA is an Assistant Professor in Faculty of Management, Comenius University Bratislava, Odbojárov 10, 82005 Bratislava 25, Slovakia. She received her PhD Comenius University Bratislava. Her research interests include deep learning, data mining, healthcare, and information management.