Prediction of Coronary Artery Disease using Deep Learning – A Basic study

Ramanathan G[1], Jagadeesha S.N.[2]

[1]Research Scholar, College of Computer Science and Information Science, Srinivas University, Mangalore,Karnataka, India

ORCID: 0000-0001-9621-3294; Email Id:ramanathanmca2006@gmail.com

[2]Research Professor, College of Computer Science and Information Science,Srinivas University, Mangalore, Karnataka, India

ORCID: 0000-0002-5185-2233; Email Id: jagadeesha2012@gmail.com

Area/Section: Health Science
Type of the Paper: Conceptual/Analytical Paper
Type of Review: Peer Reviewed as per |C|O|P|E| guidance.
Indexed in: OpenAIRE.
DOI: https://doi.org/10.5281/zenodo.14233208
Google Scholar Citation: IJHSP

How to Cite this Paper:
G.Ramanathan., & S.N. Jagadeesha (2024). Prediction of Coronary Artery Disease using Deep Learning – A Basic study .*International Journal of Health Sciences and Pharmacy (IJHSP)*, *8*(2),13-50. DOI: https://doi.org/10.5281/zenodo.14233208

International Journal of Health Sciences and Pharmacy (IJHSP)
A Refereed International Journal of Srinivas University, India.

Crossref DOI : https://doi.org/10.47992/IJHSP.2581.6411.0121
Received on: 17/08/2024
Published on: 28/11/2024

International Journal of Health Sciences and Pharmacy (IJHSP), ISSN: 2581-6411, Vol. 8, No. 2, November 2024

SRINIVAS PUBLICATION

Prediction of Coronary Artery Disease using Deep Learning – A Basic study

Ramanathan G[1], Jagadeesha S.N.[2]

[1]Research Scholar, College of Computer Science and Information Science, Srinivas University, Mangalore,Karnataka, India

ORCID: 0000-0001-9621-3294; Email Id:ramanathanmca2006@gmail.com

[2]Research Professor, College of Computer Science and Information Science,Srinivas University, Mangalore, Karnataka, India

ORCID: 0000-0002-5185-2233; Email Id: jagadeesha2012@gmail.com

ABSTRACT

Purpose: *Cardiovascular disease continues to be a major contributor to global mortality and morbidity. Forecasting cardiovascular illness is regarded as a crucial concern studying clinical data. Due to the increasing volume of data, the task of analyzing and processing it has become more complex. This is particularly challenging when it comes to maintaining e-healthcare data. Furthermore, the utilization of artificial intelligence in this research field regards the prediction model to be a crucial aspect. The field of computer-assisted diagnosis is characterized by its dynamic and rapid development. Given the possible risks linked to inaccurate medical diagnoses, considerable efforts have been directed towards enhancing computer programs that aid clinicians in achieving precise diagnoses. Artificial intelligence (AI) will assist in discovering new connections between data and disease. The implementation of AI is intended to mitigate human fallibility, decrease the expenses of medical care, and streamline the exchange of professional data in diverse formats. A branch of Machine Learning (ML) called Deep learning (DL) has a set of algorithms and software that enable a machine to autonomously acquire skills and execute tasks. DL machines utilize Neural Networks (NN) to process vast quantities of data. DL employs numerous layers to represent different levels of abstraction. ML is a highly encouraging technique for creating sophisticated and automated algorithms to investigate complex and diverse bio-medical data with multiple dimensions. This research investigates two datasets pertaining to heart disease. The study employs Neural Networks and evaluates their predictive accuracy and several other performance parameters.*

Design/Methodology/Approach: *The study employs datasets sourced from Kaggle containing heart-related information. The analysis employs Dense Neural Network (DNN) models using Python, Sklearn, Pandas, Tensorflow, and Jupyter Notebook, and their results have been analyzed.*

Findings/Results: *The ANN models built for both the datasets have performed well. The performance metrics of the model for Cleveland Dataset which is a balanced dataset is extremely good with an accuracy of 0.99. The performance for Framingham Dataset which is an imbalanced dataset is lesser with an accuracy of 0.88.*

Originality/Value: *The study involved an analysis of two cardiac databases from Kaggle. The distribution of data, interdependence of multiple features, and the impact of all features on the prediction of heart disease have been observed. DNN have been utilized to build and train models. Investigation has also been conducted to comprehend the significance of feature selection and its impact on the accuracy of predictions. This is a fundamental examination of Neural Networks and their application and appropriateness for forecasting cardiovascular illnesses.*

Paper Type: *Research and Analysis with basic study*

Keywords: Artificial Intelligence, Diagnosis of Coronary heart disease, Deep Learning, Types of Neural Network, Artificial Neural Network, Performance Metrics.

1. INTRODUCTION :

The human heart is the most important and intricate organ within the human body. On an average, the heart beats 100,000 times each day which accounts to 30 million times per year. The organ consists of four chambers and four valves that collaborate to accurately regulate the process of filling, ejecting, and overall pumping function due to the interplay of mechanical and electrical forces. The electrical and mechanical domains collaborate to control the filling of the chambers, while the synchronised opening and closing of these valves governs their proper expulsion. Cardiac conditions, such as valve stenosis, arrhythmias of the ventricular system, valvular regurgitation, and cardiac failure, can cause significant physiological consequences. These diseases can result in a range of symptoms and adverse effects, spanning from slight discomfort to potentially fatal emergency. [1].

Cardiovascular disease is a significant contributor to mortality in the present era. Significant risk factors encompass bad lifestyle choices such as smoking, consuming alcohol, and maintaining a diet rich in fat. Annually, over 10 million individuals succumb to cardiovascular disease. Adopting a health-conscious way of living and undergoing regular medical examinations are the sole dependable methods to avert cardiovascular disease.

The primary obstacle in today's healthcare system lies in rendering prompt and precise diagnoses along with effective therapy to patients. Cardiovascular disease is the leading cause of mortality worldwide; however, it is also among the most manageable conditions. Early diagnosis is the key aspect in achieving successful disease therapy.

Identifying and diagnosing coronary heart disease in its early stages might pose challenges, thereby prompting the development of computer-aided approaches. Artificial Intelligence is gaining popularity as a computer-aided detection approach in medical facilities. It helps process and analyse clinical data to generate diagnoses for medical disorders. Artificial intelligence (AI) is set to greatly transform various facets of human existence, including the management of cardiovascular disease. With the ability of modern computers to perform millions of calculations per second, it is now feasible to develop more intricate machine learning systems, which in turn brings artificial intelligence closer to human-level intelligence. [2].

Electrocardiograms (ECGs or EKGs), exercise stress tests, echocardiograms, cardiac catheterization, chest X-rays, coronary artery calcium scans, coronary angiograms, and cardiac magnetic resonance imaging (MRI) are some of the tests that doctors use today to determine the severity of disease in a person. [3]. AI helps doctors make diagnoses by examining, processing, and understanding data about patients. A lot of different AI approaches have been relied on to look at heart disease data from clinical studies. These encompass evolution classifier, decision trees, artificial neural networks, ensemble machine learning, fuzzy neural networks, binary particle swarm optimisation, random forest classifier, Bayesian algorithms. [2].

Extensive research and study have been conducted and published in this domain thus far. The objective of this investigation is to attain a thorough knowledge of the basic principles of Deep Learning (DL), particularly the Artificial Neural Network (ANN) technique, and to evaluate its effectiveness in predicting coronary artery disease. This will be achieved by implementing the concepts on two of cardiac disease datasets sourced from Kaggle. Based on several performance parameters of the models the best model is determined.

2. OBJECTIVES:

The primary objectives of the investigation are as listed below.

(1) Gaining an understanding the basic principles of Neural Networks and their hyperparameters
(2) Accessing the appropriateness of different ML algorithms in the forecast of cardiac disease based on their principle of working.
(3) Learning the various performance parameters used to evaluate a prediction model and their usability.
(4) Identifying and understanding the different attributes of the collected data.
(5) Pre-processing the dataset to address missing values and imbalance in the data distribution.
(6) Constructing dense neural networks to predict heart disease using the provided datasets. Optimizing the hyperparameters and selecting the most effective features for the models.

(7) Accessing the efficacy of the models developed based on the accuracy and other performance metrics like f1 score, roc and auc values for the datasets considered for study.

3. LITERATURE REVIEW :

A huge amount of research has been performed on how neural networks can be used to forecast Coronary Artery Disease (CAD), and the results have been published in many journals. The following table presents brief summaries of a few published articles.

Table 1: Scholarly Literature on Deep Learning for prediction of CAD published between 2005 - 2021

S. No.	Area & Focus of the Research	Outcome of the Research	Reference
1	Prediction of CAD using NN and Feature Correlation Analysis	The authors have come up with a way to use NN to identify the risk factors of CAD based on feature correlation analysis (NN-FCA). The suggested model had a bigger ROC curve and made better judgements about the risk factors in the Korean population than the Framingham Risk Score.	Kim, J. K et al., (2017). [4]
2	Data Mining Techniques to forecast heart disease and identification of the important Features	Different sets of features and seven classification methods were used to make prediction models. The results indicate that the model created with the most important features found and the data mining method (Vote - a mix of Logistic Regression and Naïve Bayes) is 87.4% accurate at predicting the disease.	Amin, M. S et al., (2019). [5]
3	Heart Failure Prediction with Sequential Data Modelling of Electronic Health Records (EHR)	People with heart failure benefit from a service that could accurately and systematically diagnose everyone. This paper comes up with a new way to do this using a data-driven methodology and an improved LSTM (long short-term memory networks) method.	Jin, B et al., (2018). [6]
4	Heart Failure risk-prediction model using both Biomarkers and clinical parameters	The paper proposes a model that could predict the chance of an incident HF based on biomarkers from patients with stable coronary artery disease, alongside clinical parameters, derived from the LIPID (Long-Term Intervention with Pravastatin in Ischaemic Heart Disease) project.	Driscoll, A et al., (2017). [7]

5	Forecast of Mortality rate in Heart Failure patients	This article builds a system to predict fatality by analyzing a real-world dataset with the goal of predicting death in people with heart failure. Getting better at predicting mortality can help with allocating resources and improving patient results.	Wang, Z et al., (2018). [8]
6	Heart Disease Prediction with feature selection using Differential Evolution (DE)	The study proposed a method for selecting key aspects for treating cardiac disease. A modified DE model was used to select features for heart disease. The work uses the 7th concept from the ten available principles for conventional DE algorithm for relative analysis. Presented system demonstrated superior performance and accuracy compared to conventional methods.	Vivekanandan, T et al., (2017). [9]
7	Role of Artificial Neural Network in paediatric cardiology.	Cardiac murmur is a major issue in paediatric cardiology because 77–95% of children have heart murmurs, of which only a tiny percentage are caused by congenital heart disease. The research proposes a neural network-based diagnostic classifier for distinguishing pathological and innocent heart murmurs from diagnostic information of phonocardiogram.	Bhatikar, S. R et al., (2005). [10]
8	Recognition and categorization of Arrhythmia using Neural Network	The study uses end-to-end neural network to sort a lot of different arrhythmias from a single-lead ECG with accuracy about the same as a cardiologist. This method might lower the number of inaccurate interpretations of computerized ECG and boost accuracy of ECG interpretations of expert human by correctly identifying the crucial conditions and setting them in order of importance.	Hannun, A. Y et al., (2019). [11]
9	Analysis of Cardiovascular images using Convolutional Neural Networks	The study tested a CNN for ventricular contouring by training on hearts which are	Tandon, A et al., (2021). [12]

		structurally normal. In rTOF patients, the MSN algorithm performed better for the Left Ventricle than the Right Ventricle across multiple measures. Using these cases, they developed an enhanced method, MSN+rTOF. This revised method improves spatial and volumetric metrics.	
10	Identification of Cardiac Rhythm Device using Deep Learning	This article talks about the development, testing, and release to the public of a new NN-based system that aims to identify from a chest x-ray the model group and producer of a defibrillator or pacemaker.	Howard, J. P et al., (2019). [13]

Kim, J. K et al [4] in their study with the KNHANES-VI dataset, tested a Neural Network-based prediction of CHD risk method on the basis of feature correlation analysis (NN-FCA). The suggested strategy aims to enhance decision-making regarding optimal treatment strategies for coronary heart disease. Haemoglobin, triglycerides, H_B, thyroid disease, cirrhosis, and H_C were not linked to CHD, but triglycerides and CRF were. In addition, BMI and Diastolic BP are linked, as are Diastolic BP and total cholesterol as well as Systolic BP and Diastolic BP. When compared to the Framingham Risk Score evaluation for the Korean population considered for study, the proposed new model had a bigger ROC curve along with a better ability to predict CHD risk.

According to Amin et al., [5] they came up with a new way to employ data mining to find the significant features. This made it easier to predict heart disease. Seven classification methods, such as Decision Tree, SVM, k-NN, LR, and NN, were used, along with prediction models that used different features. The experiments showed that it was easier to forecast the disease when the important biomarkers and the right data mining model were used, as shown by the presented method.

In their study, Jin et al. [6] created a useful strategy for predicting heart failure. The crucial thing that was added was using a NN model to find the heart problem. Specifically, one-hot encoding and word vectors were used to model the occurrence and detect events of heart disease with the aid of the basic ideas of a short-term and long-term network model. The established method has demonstrated accuracy in predicting the risk of heart failure based on calculations utilising a real dataset.

Andrea et al. [7] came up with a strategy to fix the issue of not having enough risk-prediction models that consider both biomarkers and clinical factors. So, to solve these problems, an incident heart disease forecast model was created. In this plan, a number of different risk factors were used to figure out when the first hospitalization or heart failure would happen. A risk score was derived, that put the population into 5 groups based on their level of risk, that was from less than 5% to more than 20%.

Zhe et al. [8] have offered a method based on mortality to figure out how likely it is that HF patients will die. Predicting mortality rates more accurately has improved clinical results and resource allocation, keeping high-mortality patients from not getting enough care and low-mortality patients from getting too much care. There are three goals in this plan for predicting death rates: (i) predicting fatalities that will happen in hospitals; (ii) predicting fatalities that will happen in the next year; and (iii) predicting fatalities that will happen in the next 30 days. Researchers have found that the given model works better at predicting mortality rates than other models they have looked at.

Vive and Sriman [9] in their paper try to figure out how to diagnose heart disease and pick out the most important features from the huge number of available features. A varied version of the differential evolution (DE) method is utilized to pick out features for cardiovascular disease and make the best use of those features. Fuzzy AHP and a feed-forward NN are used to identify heart disease based on the chosen critical features. The suggested combination model is 83% accurate, which surpasses few other models that are already out there. It is also tested to see how long the suggested hybrid model can make predictions, and the results look good.

Bhatikar, S. R et al. [10] aimed in their study to make a dependable screening tool for finding murmurs in the heart in children. This is a difficult task in paediatric cardiology as heart murmurs are prevalent in this age group (77–95%), but only a small part of them are caused by heart disease that was present at birth. So, today, people with heart murmurs are often checked out by both a visit and an echocardiogram. The study shows a neural network-based diagnostic classifier that can tell the difference between heart sounds that aren't harmful and those that are, using diagnostic information from the phonocardiogram.

Hannun, A. Y. et al. [11] extensively evaluated a deep-learning approach for ECG analysis across a wide range of diagnostic classes. A DNN was developed to categorise 91,232 single-lead ECGs obtained from 53,549 patients utilising an ambulatory single-lead ECG monitoring device into 12 distinct rhythm groups. The proposed DNN model obtained an area under the ROC curve of 0.97.

Ventricular contouring on cardiac MRI is the best way to do volumetric analysis for corrected tetralogy of Fallot (rTOF), but it can take a longer duration and results can vary. A CNN ventricular contouring method was created to create contours for hearts that are mostly structurally normal. In their study, Tandon, A. et al. [12] wanted to make this programme better for use in rTOF and come up with a better way to test how well it works. They made a new CNN by including rTOF training use cases and tested how well the proposed algorithm worked by using new data to make outlines for the right and left ventricles (LV and RV). In people with rTOF, the original Mostly Structurally Normal (MSN) method did a better job of shaping the LV than the RV. When the algorithm was trained again, the new MSN+rTOF algorithm did better on testing data for the LV epicardial and RV endocardial outlines.

In their investigation, Howard, J. P et al. [13] took radiographic pictures of 1,676 devices, including 45 different models from 5 different manufacturers. CNN was developed in order to categorise the photos. When it came to determining the model group from a radiograph, the neural network performed with 99.6% accuracy and the maker of a device with 96.4% accuracy. This method is accessible online to the public and may enhance the speed of identifying and treating individuals with cardiac rhythm devices.

4. MACHINE LEARNING ALGORITHMS IN PREDICTION OF CORONARY ARTERY DISEASES – A GLIPMSE :

Machine learning (ML), an aspect of artificial intelligence (AI) is becoming more commonly utilized in cardiovascular medicine. Computational processing involves the interpretation of data by computers to perform tasks, either autonomously or with human oversight. The fundamental structure of ML is built upon models which take input data (text or images) and, utilizing an integration of statistical analysis and mathematical optimization, make predictions about outcomes (such as favourable, unfavourable, or neutral) [14].

A broad range of ML techniques have been created to assist in the resolution of intricate problems that arise in practical settings. ML algorithms, which are self-improving and automated, are in a state of continuous evolution to address new challenges. The process of selecting a ML model is a crucial task that relies largely on choosing the suitable hyperparameters. These hyperparameters are required for training the model. It corresponds to the set of values of the selected model, which are not capable of being acquired from the dataset and must be supplied prior to the model commencing the training stage. The effectiveness of the machine learning model ultimately improves with a more appropriate choice

of hyperparameter altering strategies. The following table lists the details few of the ML algorithms along with their suitability in assisting the forecast and identification of coronary artery disease (CAD).

Table 2: Understanding the suitability of Machine Learning (ML) Methods for prediction of CAD

Name of the Algorithm	Principle of Working	Strength	Weakness	Hyper Parameters	Suitability for CAD
Logistic Regression	• Performs linear transformations on combinations of independent variables. • Particularly effective for forecasting probabilities.	• Trains quickly on large datasets. • Good accuracy values for simple data sets. • Good performance on linearly separable datasets. • Interprets model coefficients as feature importance indicators.	• Not recommended for datasets with more features than observations. • Overfitting is a potential issue with datasets with a large number of dimensions.	Hyperparameters used: C, penalty, solver.	Algorithm performs fairly well, but complex dataset with non-linearly separable parameters reduces accuracy.
Decision Tree	• The algorithm divides the training dataset into subsets repeatedly, on the basis of values of attributes. • It chooses the best attribute by using a metric such as Gini impurity or Entropy. • The aim is to determine the attribute that provides maximum information gain or minimises	• Considers all possible problem outcomes. • Provides clear indication of crucial fields for prediction or classification. • Can handle large datasets and parallelize for improved processing time. • Addresses imbalanced data by assigning weights to individual nodes depending on	• Decision trees exhibit a high level of computational complexity due to their multiple layers. • Deep or complex trees may overfit training data, causing poor performance on new data. • Decision trees are subject to misclassification errors when dealing with numerous classes and limited	Hyperparameters used: criterion, min_samples_split, min_samples_leaf, max_depth,	• Cases in the dataset are represented by pairs of attributes and values. Output values of the target function are discrete. • Uses a rule-based methodology for prediction. A set of prediction rules is used to make the final choice. • Takes into account a wide range of risk factors.

	impurity after the split.	the class distribution.	training samples.		
Random Forest	• Combination of the predictions of multiple decision trees for final prediction. • Each tree uses a unique bootstrap sample and hence each tree is unique. • Predictions are aggregated by majority vote to decide the final classification.	• Overfitting issue is reduced as predictions of multiple trees are averaged. • Crucial predictor attributes can be identified using feature importance. • Method is very stable and has high precision. Also it manages non-linear correlation between attributes and target variable.	• In case of large datasets, the training and prediction speed is less. • Method has high computational cost.	Hyper parameters used: max_depth, n_estimators.	• Comprehensive for classification problem as it identifies the key attributes and reduces overfitting. • The method is less sensitive to noise and outliers in the data.
Support Vector Machine	• Finds the best hyperplane in a N-dimensional space. • A hyperplane divides data points into separate groupings. • Increases the margin between nearest points across classes. • The number of features determines the hyperplane's dimension.	• Effective in high-dimensional cases; manages high-dimensional data and nonlinear relationships; efficient memory because support vectors are used. • Decision function kernel functions that can be customized.	• Not appropriate for excessively large datasets. • Performance reduces for noisy data and when the target classes overlap.		• This is basically a binary classifier where there are only 2 target classes. • Converts the training set into a linear equation in many dimension spaces from a non-linear decision surface.
K-Nearest Neighbor	• Works on the concept of similarity.	• Lower in complexity, hence easier to implement.	• Scalability is difficult as the method requires	Hyper parameters used : algorithm, metric,	• Efficient in case of medical dataset analysis and

	• Predicts the value of new data point based on the location of the nearest class of the training dataset. • Works by identifying the nearest class for new features.		computing power and data storage. This is time consuming and resource intensive. • Difficult to classify data points when the dimensions are high. • Over fitting is a common issue and requires feature selection and dimensionality reduction.	n_neighbors, weights.	prediction as it works on the concept of pattern recognition. • Uses a non-parametric strategy to avoid identifying specific parameters. • Makes no assumptions regarding feature or dataset output.
Adaptive Boosting	• AdaBoost algorithm iteratively trains weak classifiers on training datasets, assigning higher importance to misclassified data points. The final model is formed by combining all trained weak classifiers, weighting them according to their accuracies. • Decision stumps are employed as the base classifiers.	• The algorithm employs decision stumps as individual models, which necessitates comparable preprocessing steps as decision trees. This approach minimises the risk of overfitting. • Lowers both bias and variance. • Improves the precision of underperformi ng classifiers.	• The quality of the dataset is the key. The model is sensitive to outliers and noise in data. Hyperparamete r optimization is challenging.	Hyper parameters used: learning_rate, n_estimators.	• Detects model weaknesses by evaluating data points with high weights. • Each classifier gives weights to the final prediction based on its performance. • Captures the most amount of variation in the data.
Gradient Boosting	• Consolidates weak learners into strong ones.	• Facilitates the processing of categorical features.	• Susceptible to overfitting which can be handled using	Hyper parameters used: learning_rate, max_depth, n_estimators.	• Determines shortcomings with previous models by

	• Gradient descent is used to reduce the loss function (MSE - mean squared error) for each new model. • Classification and Regression Trees (CART) serve as the framework for gradient boosted trees. • Determines the weights by calculating the loss function's negative gradient. • Decision trees and linear models may be employed as base learners.	• Trains more rapidly on larger datasets • Model has greater accuracy compared to other models. Certain models can inherently handle missing values.	L1 and L2 regularization. • Models tend to have high computational cost and take longer duration to train.		using gradient. • All classifiers are given equal weightage. • New trees are built on the residuals of the previous classifiers.
Extreme Gradient Boosting	• This is a better version of the gradient boosting method. Parallel processing at the node level makes it faster and more powerful. • Regularizatio n techniques reduces over fitting issue. • This is a linear model that performs	• Model utilises both L1 and L2 regularization techniques to ensure reliable and precise predictions. • It performs quickly and efficiently, even when dealing with large amounts of data.	• Recognised as a "black box" algorithm, which complicates the interpretation and improving of predictions. • Necessitates technical skill for efficient execution and optimization. • Hyperparamete r tuning can be time consuming.	Hyper parameters used: eval_metric, learning_rate, max_depth, n_estimators.	• Includes functionality for doing cross-validation and highlighting the importance of features. • Discovers the path for missing values through various approaches. • Divides the tree to the max_depth specified, and then prunes it

	tree learning through parallel processing.				in reverse. Excludes divisions in which there is no positive gain.
Light Gradient Boosting	• A histogram-based algorithm employed in all gradient-based decision tree (GBDT) frameworks. • The model employs a leaf-wise tree growth algorithm to achieve faster convergence. • It utilizes two techniques: gradient-based on side sampling (GOSS) and exclusive feature bundling (EFB). • Continuous values are divided into bins to enhance training speed and optimize memory usage.	• Built-in support for categorical features. • Automatic handling of missing values with default values. • Improved speed and accuracy. Reduced memory usage	• Growth of the tree Leaf-Wise makes the model more complicated and could cause it to overfit small datasets.	Hyper parameters used: learning_rate, boosting_type, n_estimators, max_depth.	• Model is faster, memory efficient and suitable for large scale datasets. • Categorical features can be used directly without one-hot encoding.

5. EVALUATION OF A MODEL – PERFORMANCE METRICS :

The process of designing a ML or NN models relies on the basic idea of constructive feedback. Evaluation metrics provide a quantifiable rating of the model's performance. The evaluative measures' ability to distinguish between different outcomes is an important factor to consider. While examining a ML model, it is important to analyse its capability to predict accurately, generalisation ability, and the quality on the whole. Evaluation metrics provide definitive criteria for assessing these attributes. Choosing the right evaluation metrics is contingent upon the domain of the interest, the data, and result that is visualized [15].

Predictive models can be having a continuous output (a regression model) or having a binary output (a classification model). The parameters to access the performance of these models are different.
Classification algorithms can be categorized into two groups based on their output:
Class output: Techniques like Support Vector Machine and K-Nearest Neighbor a class output is created. For example, for binary classification, the model's output can be 0 or 1.
Probability output: Few algorithms such as Random Forest, Gradient Boosting, Logistic Regression, Adaboost give probability outputs. Transforming probability outputs into class outputs just involves establishing a threshold probability.
In regression problems the model's output is always continuous. This output need not be transformed further.
Multiple metrics are available for the assessment of models. The selection of the metric is dependent upon the model's nature and the model's implementation strategy. After constructing the model, one can employ the following 8 measures to judge the model's performance [15].

1. Accuracy
2. Recall
3. Precision
4. Confusion Matrix
5. F1 score
6. Log Loss
7. AUC – ROC Curve
8. RMSE (Root Mean Squared Error)

Confusion Matrix

A confusion matrix, also referred to as an error matrix, is a tabular representation that displays the counts of both accurate and inaccurate predictions made by the model, compared with the actual classifications found in the test set. This metric is used to access the accuracy of classification tasks with 2 or more class outcomes. The matrix has a size of N × N, N denoting the number of potential classes. Below is a tabular representation that shows the four possible combinations of actual and predicted class outputs. This matrix assists in evaluating accuracy, recall, specificity, precision, and plotting AUC-ROC curves.

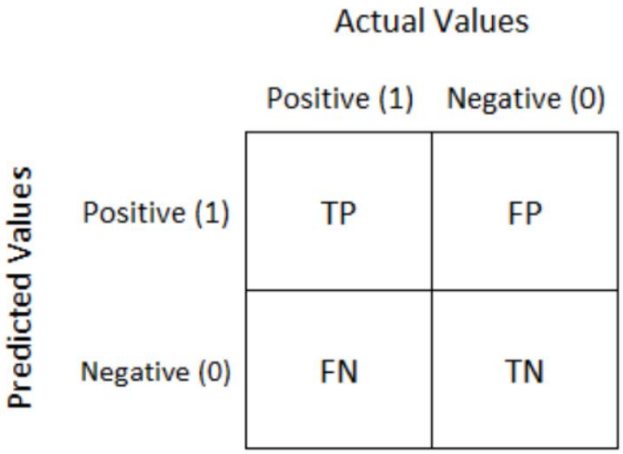


Figure 1: Confusion Matrix [16]

There can be 4 possible outcomes while performing a classification prediction as listed below:

- True Positives (TP) – Total occurrences which are truly positive and predicted to be positive.
- True Negatives (TN) - Total occurrences which are truly negative and predicted to be negative.

- False Positives (FP) – Total occurrences which are truly negative but predicted positive. FP is categorized as Type 1 Error.
- False Negatives (FN) - Total occurrences which are truly positive but predicted negative. FN is categorized as Type 2 Error.

The terms "positive" and "negative" pertain to the actual prediction being made. The terms "True" and "False" pertain to the accuracy of the prediction.

The diagonal members of the matrix indicate the count of correctly predicted instances, whereas the off-diagonal elements show the count of mislabelled instances by the classifier. A larger value in the diagonal of the confusion matrix indicates a better performance, suggesting a greater instances of accurate predictions.

Accuracy

Accuracy quantifies the instances of correctly identified observations, including the positive and negative outcomes. This metric quantifies the instances in which the classifier accurately predicts outcomes. Accuracy is the quotient obtained by dividing the instances of correct predictions by the total instances of predictions.
A model exhibiting an accuracy rating of 99% may not be performing optimally and can be false in some scenarios. This metric is not suitable for imbalanced datasets. Attaining a high accuracy score can be effortlessly achieved by identifying all observations as belonging to the majority class. For instance, when identifying fraudulent transactions, an accuracy rate of above 0.9 can be attained by categorising all transactions as non-fraudulent.

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

Precision

The number of positive findings that were perfectly calculated out of all the positive observations expected is called precision. This measure exhibits how many out of all the positive events predicted by the model were actually verified to be true. It can help to improve the imbalanced and skewed dataset. When reducing the total instances of false positives is more critical than reducing the total instances of false negatives, precision becomes very valuable.

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

Recall

Recall refers to the proportion of correctly identified positive instances relative to the total count of actual positive instances. The statistic denotes the proportion of accurately identified positive cases relative to the total count of positive instances in the dataset. This metric assesses the model's capability to recognise positive samples. A higher count of False Negatives predicted by the model, leads to a reduction in recall. In medical scenarios, the detection of true positive instances is crucial, even though a false alarm is acceptable.

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

- Precision examines the classification of both positive and negative samples, while the main concern of recall is on the classification of positive samples. To clarify, the accuracy is influenced by both the negative and positive samples, whereas the completeness is solely influenced by the positive samples and is not influenced by the negative samples.
- Precision examines the accuracy of categorising a sample as Positive, without considering the proper classification of all positive samples. The recall metric prioritises accurately identifying all positive samples, while disregarding the misclassification of negative samples as positive.

F1 Score

When Precision and Recall are harmonically averaged, the result is the F1 Score. It is a composite measure that integrates Precision and Recall. The model has a better performance when the F1 score is higher. The F1-score can take on values between 0 and 1.

$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

The F1 Score serves as a valuable evaluation metric in various scenarios:
- When FP and FN's expenses match.
- Addition of data does not meaningfully modify the result.
- High True Negative

AUC – ROC Curve

The following table can help in understanding the various metrics.

Table 3: Understanding the various metrics using Confusion Matrix

Confusion Matrix		Actual			
		Positive	Negative		
Predicted	Positive	a	b	Positive Predictive Value	a/(a+b)
	Negative	c	d	Negative Predictive Value	d/(c+d)
		Sensitivity	Specificity	Accuracy = (a+d) / (a+b+c+d)	
		a/(a+c)	d/(b+d)		

The relationship between Sensitivity and Specificity constitutes the Receiver Operating Characteristic (ROC) curve.

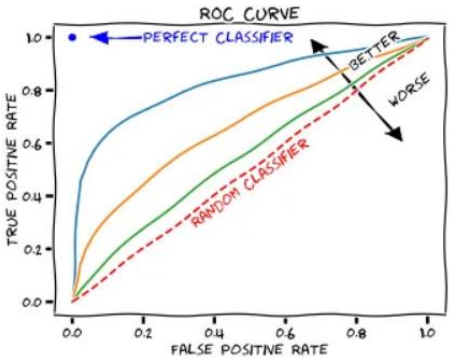


Figure 2: Receiver Operating Characteristic Curve [17]

Sensitivity can be referred as the true positive rate (TPR). Specificity can be referred as the false positive rate (FPR). An ideal classifier will have a high TPR and a low FPR.

- Any model which has a ROC curve above the random classifier line can be regarded as a good performing model.
- Any model that has a ROC curve below the line of the random guessing classifier can be clearly rejected.

The ROC curve shows TPR and FPR at various classification thresholds, but this is not the best way to do it since we need to test our model at many levels. The AUC algorithm, which is built on sorting, is a fast way to get this information.

AUC refers to Area Under the ROC Curve. The AUC value quantifies the capacity of the classifier to differentiate between different classes. The ROC AUC score, affords a single number that indicates the quality of a curve.

A higher AUC indicates superior model performance across various threshold points distinguishing positive as well as negative classes. When the value of AUC is 1, it indicates that the classifier can perfectly differentiate the positive from negative classes. When the AUC value is equal to 0, the classifier will be incorrectly identifying all negative cases as positive and vice versa. A classifier where AUC value is 0.5 will be unable to differentiate between the positive class and negative class.

Log Loss

The most crucial probability-based classification metric is log loss. A well-known evaluation metric for models with binary classification is log loss, which is often referred to as logarithmic loss or cross-entropy loss. By quantifying the discrepancy between expected probabilities and actual values, it evaluates a model's performance. Log-loss penalises inaccurate predictions with higher values and displays the degree to which the prediction probability is similar to the corresponding actual value (0 or 1 for binary classification). Improved model performance is indicated by a lower log-loss.

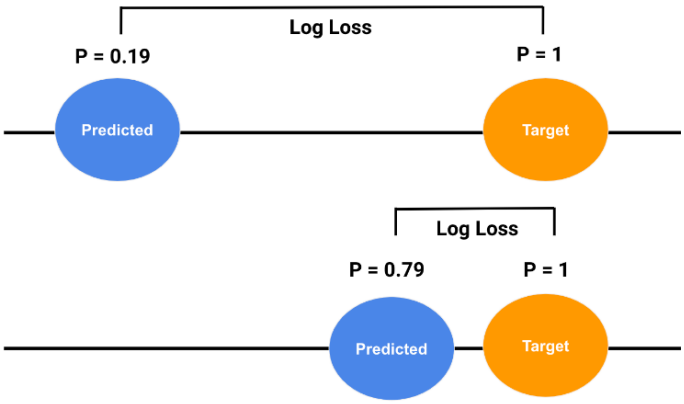


Figure 3: Representation of Log Loss [18]

The above figure represents the concept of log loss.

- The first image illustrates a wrong prediction due to the significant disparity between the observed and projected probability, which result in a substantial log loss. In this scenario, the function imposes a penalty on the incorrect answer which the model is extremely certain about.
- The lower figure illustrates an accurate prediction since the anticipated probability closely aligns with the actual probability, resulting in a minimal log loss. In this case, the function is providing a reward for a right answer which the model is extremely certain about.

Log loss can be derived using the below formula.

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

- $p(y_i)$ = Predicted probability for a positive class
- $1 - p(y_i)$ = Predicted probability for a negative class
- y_i is the actual value. $y_i = 1$ (Positive class) $y_i = 0$ (Negative class)

Log loss is a non-negative quantity that exists within the interval [0, ∞]. By minimising log loss, the classifier achieves enhanced accuracy.

Root Mean Squared Error (RMSE)

A commonly utilised metric in statistics and ML for evaluating the accuracy of predictive models is the RMSE value. This value assesses the magnitude of the deviation between the expected and actual values by doing the following procedures: squaring the errors, computing the mean, and subsequently taking the square root. RMSE offers an accurate performance evaluation of the model, wherein reduced values signify enhanced predictive accuracy.

RMSE computes the average of the discrepancies from the actual value and is focused with determining the extent of the errors. The model having an RMSE value of zero signifies a flawless fit. As RMSE decreases, the performance of the model and its predictions improve. A larger value of RMSE signifies a substantial discrepancy between the predicted and the actual truth.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- y_i = actual value
- $\hat{y}_i$ = predicted value
- n = number of samples

The above are few of the metrics employed for the assessment of classification or prediction models. These metrics have been used for assessing of the prediction model trained as part of this research.

6. DEEP LEARNING – WORKING PRINCIPLE OF A NEURAL NETWORK :

Both deep learning (DL) and machine learning (ML) are methods for teaching computers to learn and make decisions. ML is similar to instructing a computer to identify patterns and generate predictions. Algorithms are employed to analyse data and identify relationships. ML is a broad methodology that entails the training of models to generate predictions by identifying patterns within datasets. DL is an application of ML that draws inspiration from the human brain. A comprehensive artificial neural network comprising interconnected layers is utilized to acquire knowledge and comprehend intricate patterns. DL is highly effective at managing intricate and detailed tasks; however, it demands increased computational and data resources. The selection between ML and DL is dependent upon the problem at hand and the aspects of the data that is accessible.

DL enables the extraction of more complex features from the data through a neural network (NN) that is modelled based on the operation of the neural system in the human body. A neural network consists of multiple layers, each layer containing several units. These layers are categorized as below.

- Input layer
- Hidden layer(s)
- Output layer

The various components of NN include neurons, connections, weights, bias, propagation function and learning rule.

Inputs are sent to neurons, and linear function and activation function determine what happens. Weights and biases control how information is sent through connections. Learning, which involves changing weights and biases, happens in three steps: input processing, output generation, and continuous refinement, which makes the network better at many different tasks.

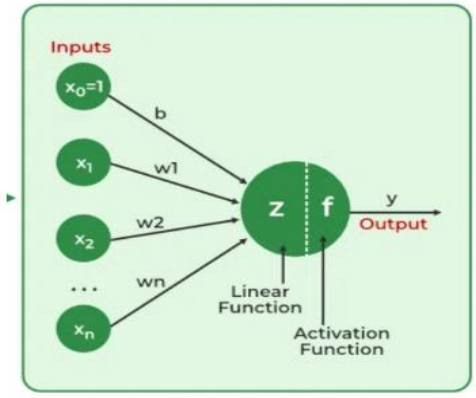


Figure 4: Representation of a simple Neural Network [19]

In a dense neural network, each node in one layer is connected to every node in the next layer by connections. The process of training a NN happens in two steps such as Forward Propagation and Back Propagation.

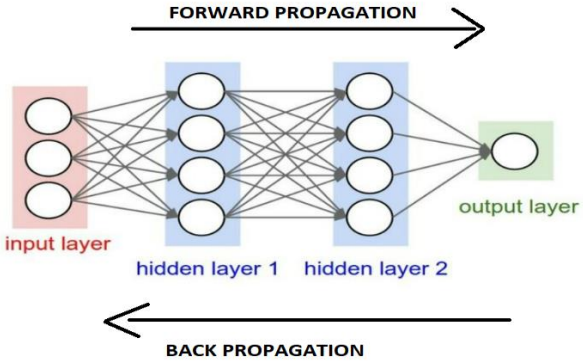


Figure 5: Representation of a Dense Neural Network [19]

Forward Propagation

The following are the components and steps involved in forward propagation:

- Input Layer – Every parameter of the dataset is defined by a neuron in this layer of the network.
- Weights and Connections - Every neuronal link has a weight that shows how strong it is. While training, the values of weights are modified in every iteration based on the result obtained.

- Hidden Layers - Multiplying inputs by their corresponding weights, summing the results, and then passing them through an activation function are the operations performed by each hidden layer neuron. This results in the introduction of non-linearity, which helps the network to determine complex patterns.
- Output - By iterating through the steps until the output layer is reached, the ultimate outcome is generated.

Back Propagation

The back propagation involves the following components and steps:

- Loss Calculation – A loss function calculates the difference between network output and target values. RMSE (Root Mean Square Error) is the loss function used generally for classification problem.
- Gradient Descent - Following this, the network implements gradient descent to decrease the loss. To reduce the level of error, the weights are adjusted in accordance with the derivative of the loss pertaining to each weight.
- Weight Adjustment - By employing this backpropagation, or iterative process, in reverse throughout the network, the weights are modified at each connection.

Forward propagation, loss calculation, and backpropagation are all done over and over again while the network is being trained with different sets of data. This lets the network to evolve and learn patterns from the data.

Activation Function

A neural network employs an activation functions to compute the weighted total of inputs and biases; this sum helps determine whether a neuron can be activated. The algorithm performs data manipulation on the provided input and generates a neural network output that comprises the parameters of the input.

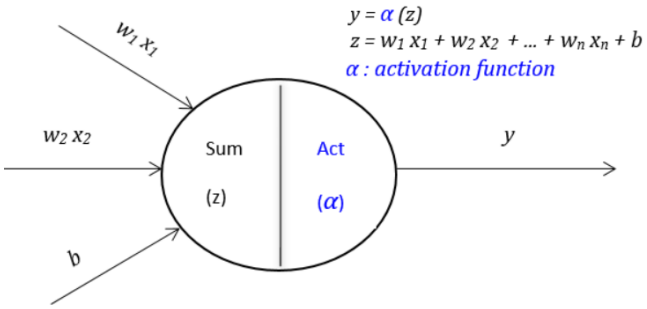


Figure 6: Representation of Activation Function [20]

On the basis of the function they represent, these may be linear or nonlinear and are utilised to regulate the output of neural networks across domains.

- Linear Activation Function - A linear function's activation is directly proportional to its input, which consists of the weighted sum of the values received from each neurone. This function is also called a straight-line function. It has a simple equation.

$$f(x) = ax + b$$

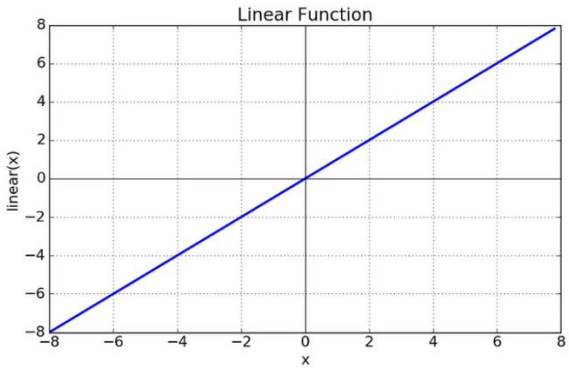


Figure 7: Curve of Linear Function [21]

- Nonlinear Activation Function - Nonlinear functions are the most frequently employed activation functions. It facilitates the adaptation of neural network models to diverse data sets and the differentiation of outcomes. These functions are categorised according to their curves.

i. ***Sigmoid or Logistic Activation Function*** – The curve of a Sigmoid function has the shape of letter S. The range of a Sigmoid function is confined between 0 and 1. Hence it is utilized in models where the chance of the output must be anticipated.

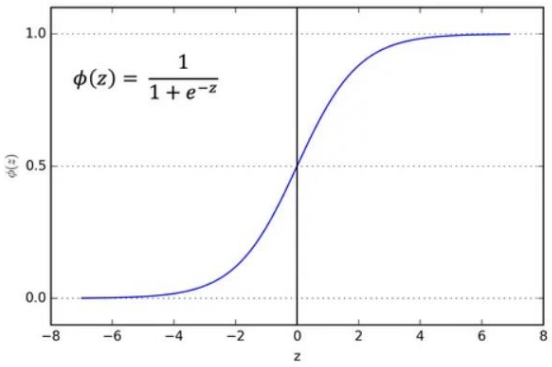


Figure 8: Sigmoid Function [21]

ii. ***Tanh or hyperbolic tangent Activation Function*** - The hyperbolic tangent (tanh) function is an effective option for a non-linear activation function in neural networks, used to connect layers. It exhibits certain similarities with the sigmoid function. The Tanh function differs from a sigmoid function in that it maps input values to an array of -1 to 1, rather than 0 to 1. This function is primarily applied in classification problems involving 2 classes.

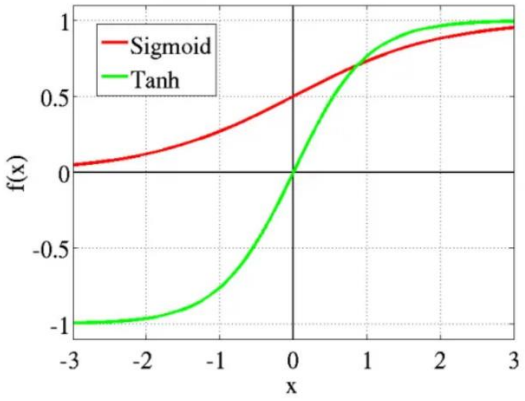


Figure 9: Curves of Tanh and Sigmoid [21]

iii. ***Rectified Linear Unit Activation Function (ReLU)*** – Deep Learning models predominantly employ the Rectified Linear Unit (ReLU) function. The function outputs 0 in case of negative input, while for any positive input x, it returns x itself. Hence the function can be represented as $f(x) = \max(0, x)$.

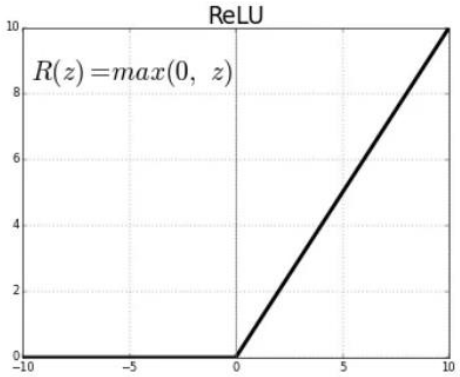


Figure 10: Curve of ReLU function [21]

One disadvantage of the ReLU function is that it instantly sets all negative values to zero, hence reducing the model's capacity to effectively fit or train on the facts. Hence, any negative input provided to the ReLU activation function will promptly be converted to zero on the graph. This, in turn, impacts the resultant graph by failing to accurately map the negative values.

iv. ***Leaky ReLU*** - Leaky ReLU can be described as a modified version of the ReLU. A leaky ReLU represents a variant of the rectified linear unit that permits a modest, non-zero, constant gradient α (often α=0.01). The magnitude to which the benefit is consistent across tasks is currently uncertain. Leaky ReLUs aim to tackle the problem of "dying ReLUs".

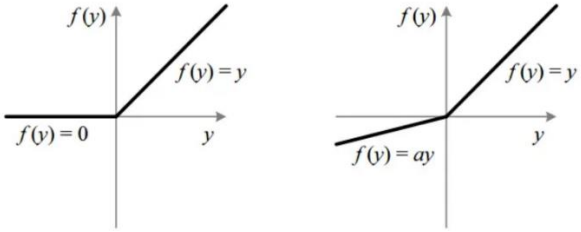


Figure 11: ReLU vs Leaky ReLU [21]

v. ***Maxout*** - The Maxout activation function serves as an advancement of the ReLU and leaky ReLU functions. The function is a linear function with piecewise operation that gives the highest value among the given inputs as output. ReLU and leaky ReLU are both subsets of the Maxout function. The Maxout neuron possesses all the advantages of a Rectified Linear Unit (ReLU) while it does not have the drawbacks such as the issue of dying ReLU. However, this results in a twofold increase in the overall number of parameters per neuron, necessitating the training of a larger total number of parameters.

vi. ***ELU*** - **The Exponential Linear Unit (ELU)** is a mathematical function that exhibits faster convergence and provides more precise outcomes. ELU, unlike other activation functions, requires an additional alpha constant, which has to be a positive numerical value. ELU is identical to ReLU, with the exception that it handles negative inputs differently. Both functions exhibit the characteristics of the identity function when the inputs are non-negative. However, the ELU function gradually gets smooth until its output reaches -α, while the ReLU function rapidly smooths.

$$f(x) = x, when\ x \geq 0$$
$$f(x) = \alpha(e^x - 1), when\ x < 0$$

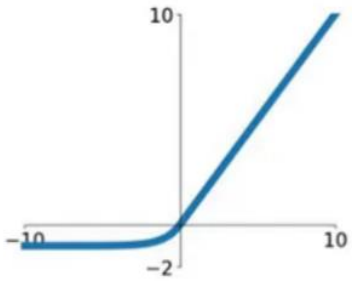


Figure 12: Curve of ELU [22]

vii. ***Softmax*** - The Softmax function determines the event's probability distribution across 'n' distinct events. This function will compute, in a general sense, the probabilities associated with each target class out of all potential target classes. The probability values that have been calculated will subsequently aid in identifying the target class given the inputs. In other words, the Softmax activation function converts the unprocessed neural network outputs into a probability vector, which is a distribution of probabilities across the input classes.

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

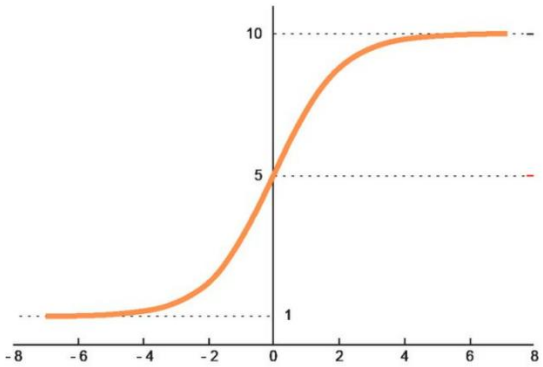


Figure 13: Curve of Softmax Function [23]

Choice of the Activation Function

The selection of the activation function is reliant upon the type of problem and the range of values anticipated for the output. For instance, when predicting values greater than 1, the usage of tanh or sigmoid in the output layer is not appropriate. Instead, ReLU can be employed. However, if the desired output values must fall within the range of (0,1) or (-1, 1), ReLU is not an optimal option, and instead, sigmoid or tanh functions can be employed. When doing a classification job and utilising a neural network to forecast a probability distribution across the distinct class labels, it is advisable to employ the Softmax activation function in the final layer. Regarding the hidden layers, the general recommendation is to use Rectified Linear Unit (ReLU) as the activation function for these layers. For

binary classification, the Sigmoid function is to be employed in the final layer, whereas ReLU or Leaky ReLU may be utilised in the hidden layers.

Neural Networks Types

Numerous varieties of neural networks exist. The majority of previously developed neural network models in deep learning are based on the following varieties of neural networks.

i. ***Perceptron*** - The perceptron is a basic form that is employed for tasks involving the categorization of data into two categories. A type is made up of a singular layer of artificial neurons, often referred to as perceptrons, which receive input values, apply weights to them, and produce an output. The perceptron is commonly employed in datasets which can be linearly separated, where it acquires the capability to categorise inputs into two distinct groups by utilising a decision boundary. But the perceptron is limited in its ability to process complicated data that cannot be separated linearly. Its straightforward architecture renders it unsuitable for complicated tasks. This network is utilised for the implementation of Logic Gates such as AND, OR, or XOR.

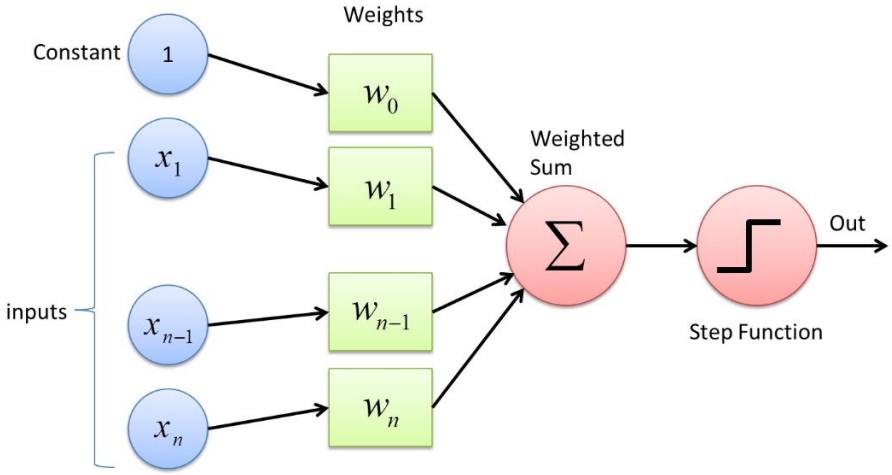


Figure 14: A Perceptron Model [24]

ii. ***Artificial Neural Network (ANN)*** - This is a type of feedforward neural network that is made up of three or more layers: an input layer, either one or several hidden layers, and an output layer. The system employs nonlinear activation functions. ANNs typically refer to neural networks that consist of fully connected layers. Put simply, every neuron in one layer is linked to every neuron in the neighbouring layers. Therefore, an ANN possesses greater computational capacity in comparison with a perceptron. Nevertheless, the interconnectedness of the networks results in them susceptible to overfitting data. Common methods for minimising overfitting involve implementing early stopping, incorporating dropout layers, and introducing regularization terms. ANNs are employed in data compression for social networks, speech recognition, recognition of hand-written character, computer vision applications, and data prediction systems.

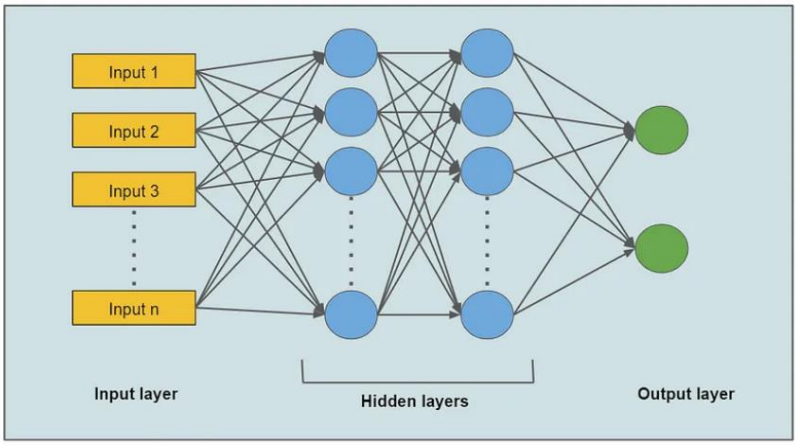


Figure 15: An Artificial Neural Network [25]

iii. ***Convolutional Neural Network (CNN)*** – This type is a specialised category of ANN that is specifically created for the purpose of image processing. The system utilises convolutional layers to actively acquire hierarchical characteristics from input pictures, facilitating efficient recognition and categorization of images. CNNs have brought about an important transformation in the domain of computer vision and play a crucial role in activities such as identifying objects and analysing images. CNNs are composed of kernels which are filters. Kernels are employed to extract suitable information from the input using the convolution operation. CNNs, initially developed for picture data, exhibit remarkable performance when applied to sequential inputs. The distinctive feature of a CNN is its convolution layer. This layer does the dot product operation, which involves multiplying each component of two vectors and then summing the results. During the early stages of a CNN, the filters are randomly initialised and do not yield any meaningful outcomes. By employing a loss function, the filters are fine-tuned, and via numerous iterations, the network improves its ability to accomplish its objective. They usually need an extensive quantity of training data.

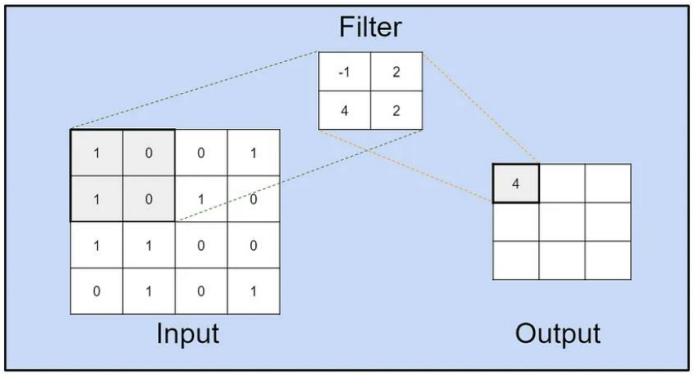


Figure 16: Convolution operation using a filter in a CNN [25]

iv. ***Recurrent Neural Network (RNN)*** - This is a method of ANN designed specifically for processing progressive facts. It is applicable for tasks that rely on contextual dependencies, such as time series prediction and natural language processing, as it utilises feedback loops to retain information within the network. RNNs are specifically built to process and understand data that is presented in a temporal or sequential manner. RNNs utilise additional data points within a sequence to improve the accuracy of their predictions. This is achieved by receiving input and utilising the activations of preceding nodes in the sequence to impact the output. A RNN consists of a recurrent module that takes input from the preceding stage and feeds its output as input to the subsequent stage.

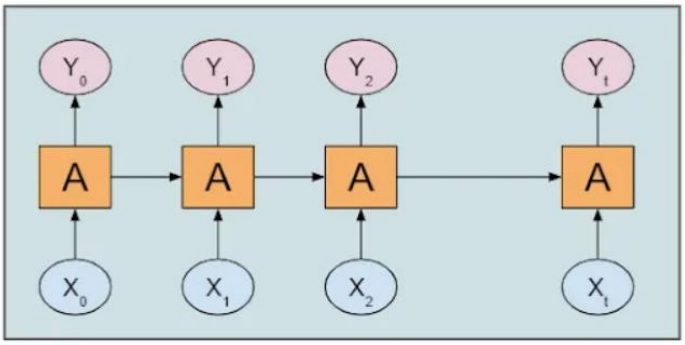


Figure 17: Basic working architecture of RNN [25]

v. ***Long Short – Term Memory (LSTM)*** - RNNs have the ability to maintain information from the most recent stage. However, when it comes to a complex task such as language translation, a far higher level of retention is necessary. In order to acquire knowledge of long-term dependencies, the neural network requires the ability to store and recall information. Long Short-Term Memory (LSTM) models are a specific type of RNNs that have the capability to perform that task. LSTMs possess a similar sequential arrangement as RNNs, however with a distinct recurring module configuration. The utilisation of this recurring module structure enables the network to preserve a significantly greater quantity of past stage data. LSTM is specifically engineered to address the issue of the problem of vanishing gradient which occurs while training of RNNs. The system employs memory cells and gates to selectively access, save, and delete data.

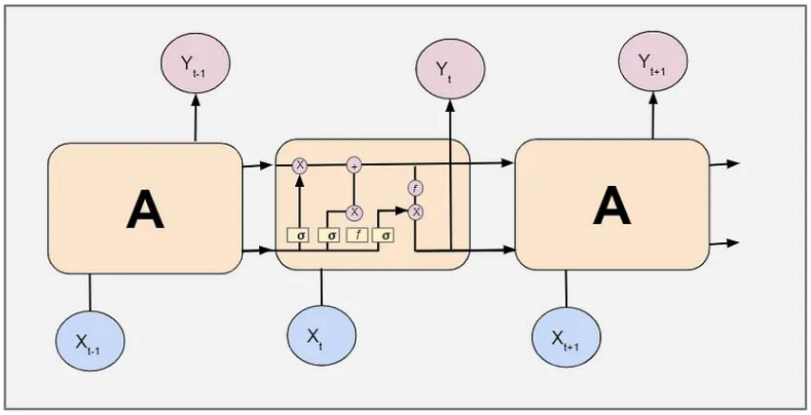


Figure 18: LSTM network [25]

ML vs DL – Basic Comparison

Deep Learning (DL) and Machine Learning (ML) are terms that can be used interchangeably, but they represent different concepts. The following are the key differences between them.

Table 4: Machine Learning (ML) vs Deep Learning (DL)

S.No	Machine Learning (ML)	Deep Learning (DL)
1	The ML algorithms are fed lots of data and allowed to adapt their operations based on the patterns and links they find.	Computers can handle and analyse vast volumes of complicated, unstructured data using DL techniques, which imitate the human brain.
2	ML algorithms require structured data and the quantity of data required is relatively lesser than DL.	DL specializes in processing huge quantity of complex unstructured data.

3	ML algorithms can normally run on traditional computers.	DL requires powerful hardware as they process more complex data. GPU or Graphics Processing Units can help in meeting the computational demands of DL.
4	ML algorithms are easier to interpret as they work on traditional statistical methods.	DL models are difficult to interpret and explain as they work based on complex neural network concept.

Raw data cannot be utilised by ML algorithms for learning purposes. Strict engineering and domain expertise in highly classified information are required to extract features from unprocessed data. Patterns are then found by using these features in internal representations. The initial stage of the ML process is omitted in DL. This process is carried out automatically by DL. DL is capable of automatically mining new features from unprocessed facts.

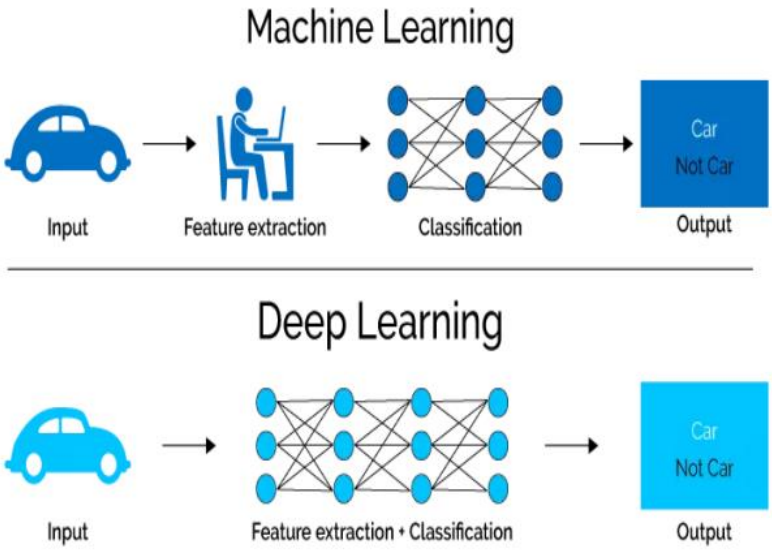


Figure 19: Comparison between ML and DL processing [26]

7. PREDICTION OF CORONARY ARTERY DISEASE USING NEURAL NETWORK:

The prediction of coronary artery disease (CAD) by artificial neural networks is the primary goal of this study. Two heart disease-related datasets with various sets of biological factors have been examined for the purpose of CAD prediction as part of the study. The steps involved in the process are listed below:

i. Analysis and Cleaning of Dataset
ii. Pre-processing of Dataset
iii. Splitting into Training and Testing set
iv. Training using a dense neural network
v. Evaluating the performance with Testing set

The details of the software used in the analysis are as follows:

- Python 3.9
- Pandas 1.5
- Jupyter Notebook
- Scikit-learn
- Seaborn

- Numpy 1.23
- Matplotlib 3.6
- Tensorflow 2.11

Details of the Study population

The study researches 2 different datasets obtained from Kaggle for CAD prediction. The following section details the statistics of the datasets.

Cleveland dataset for heart disease prediction

This dataset has a total study population of 1025 records. The dataset has no null values and a balanced dataset with nearly equal population for both the target values (0 – no disease, 1- presence of disease). The dataset has a set of 13 parameters and 1 target parameter. The details of the parameters are listed below:
Categorical parameters – Sex, ChestPain, FastingBS, RestECG, ExerAng, slope, ca, thal, target
Continuous parameters – Age, RestingBPS, Cholesterol, MaxHRate, STDepress
The study population include 499 patients who do not have heart disease and 526 patients who have heart disease. The following graph represents the distribution on the basis of target parameter.

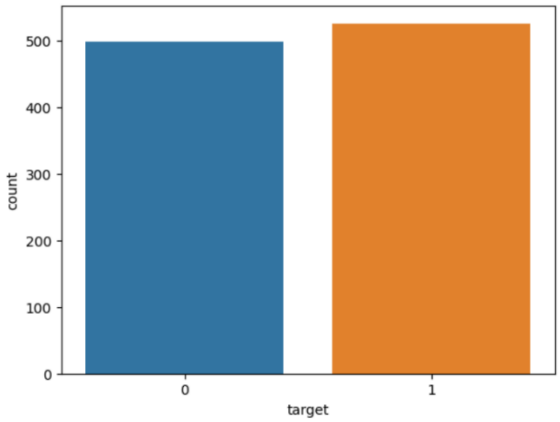


Figure 20: Histogram of target parameter (Cleveland Dataset)

Correlation heatmap has been plotted to understand how closely related certain parameters are to the target parameter. The correlation heatmap (Figure 21) of the dataset indicates that the 'target' parameter is influenced by all the other parameters of the dataset.

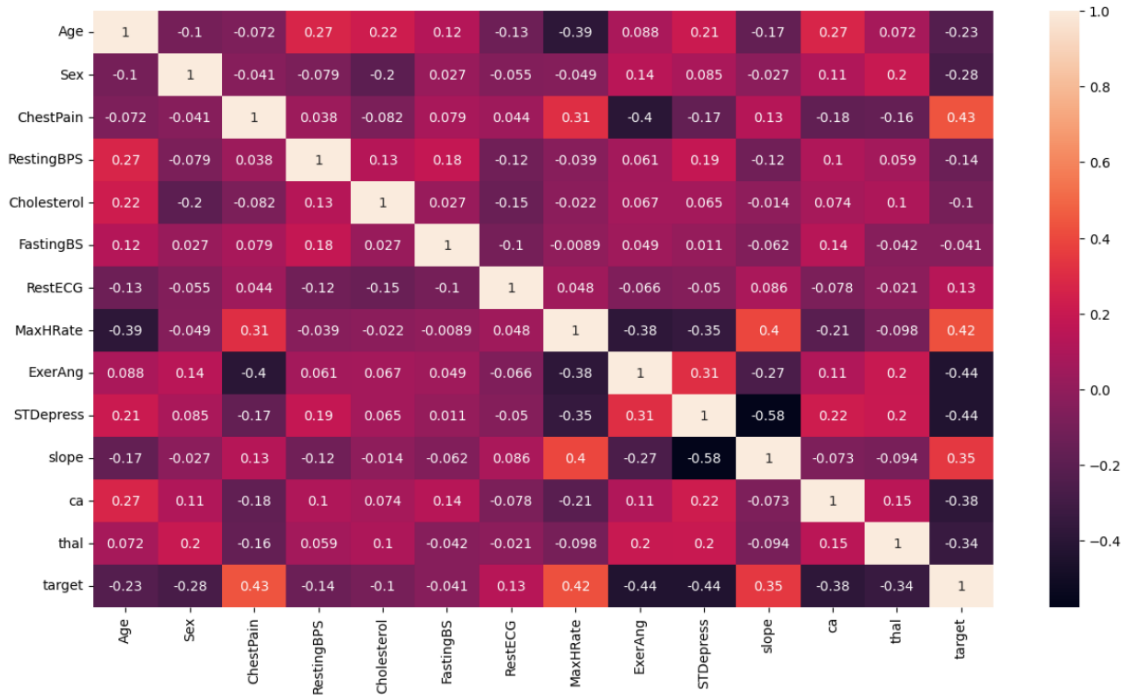


Figure 21: Correlation heatmap of Cleveland Dataset

During preprocessing, the dataset features must be scaled to a similar scale. This will facilitate a faster convergence of the gradient descent towards the minima. Feature scaling facilitates optimization within the framework of neural network algorithms by: It expedites the training process. It keeps the optimisation from being slowed down in nearby optima. The following parameters of the dataset have been scaled using the MinMaxScaler of Scikit - learn library – Age, ChestPain, RestingBPS, Cholesterol, RestECG, MaxHRate, STDepress, slope, ca, thal.

The train_test_split module of the Scikit-learn toolkit has been used to divide the dataset into training set and testing set. There are 13 input parameters and 1 output (target) parameter. The training set has a size of 820 records and testing set has a size of 205 records. The dense neural network for the Cleveland dataset has the following structure:

- Input Layer – 13 neurons (activation – ReLU)
- Hidden Layer – 128 neurons (activation – ReLU)
- Dropout – 0.2
- Hidden Layer – 64 neurons (activation – ReLU)
- Dropout – 0.2
- Hidden Layer – 32 neurons (activation – ReLU)
- Dropout – 0.2
- Hidden Layer – 64 neurons (activation – ReLU)
- Dropout – 0.2
- Hidden Layer – 32 neurons (activation – ReLU)
- Dropout – 0.2
- Output Layer – 1 neuron (result) (activation – Sigmoid)

Optimizer – Adam

Loss Function – Binary Cross Entropy

Epochs – 2000

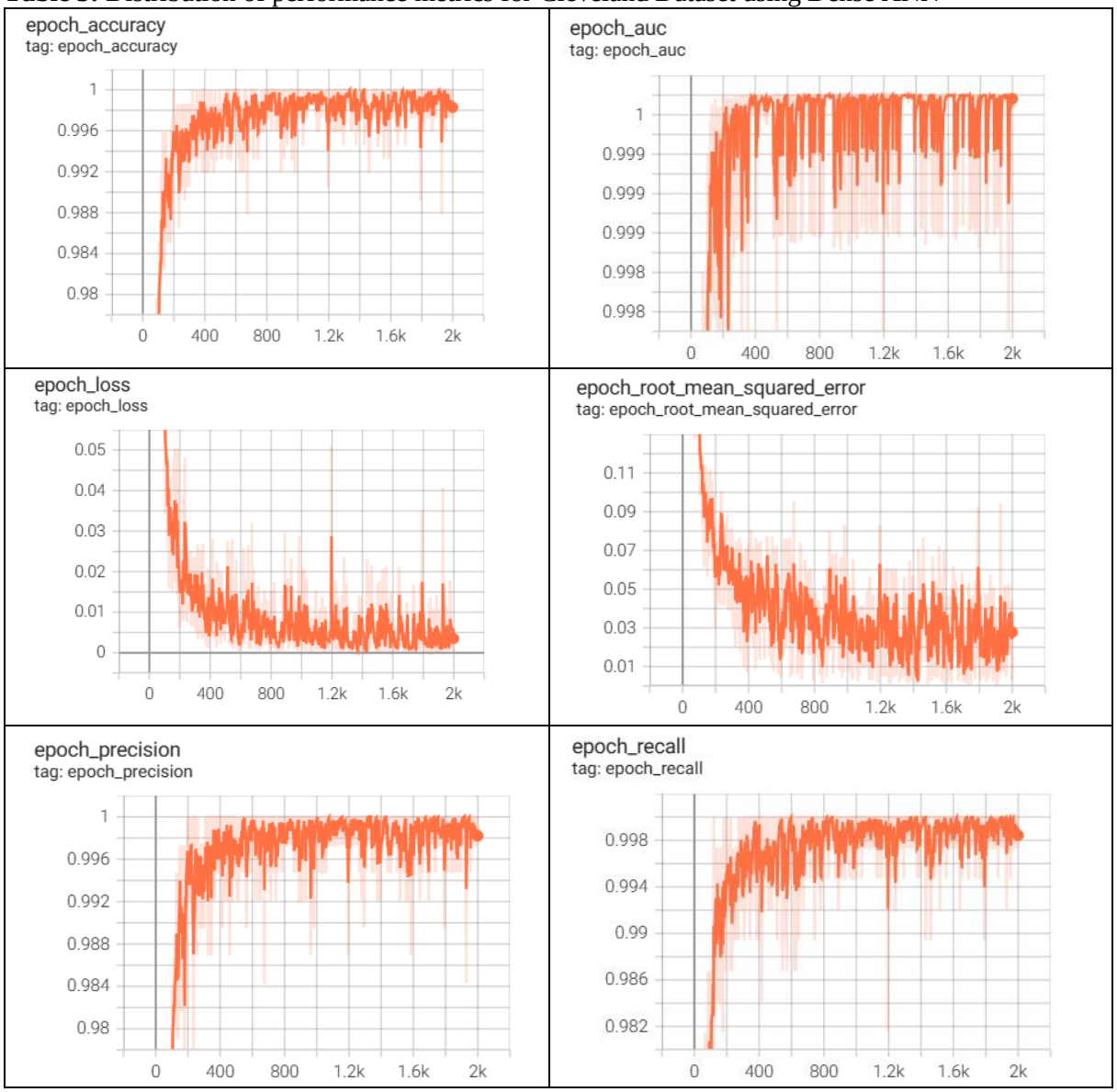
After training the dense neural network, the evaluation of its performance is conducted using the testing set. The following result is achieved.

- Accuracy – 0.99

- AUC – 1.0
- RMSE – 0.093
- Loss – 0.0193
- Precision – 0.98
- Recall – 1

The graphs of the various performance parameters through the epoch of 2000, has been obtained from Tensor board. They are as shown below.

Table 5: Distribution of performance metrics for Cleveland Dataset using Dense ANN



The distribution and values of Accuracy and AUC show that the model trained for the balanced dataset has good performance.

Framingham dataset for heart disease prediction

The Framingham dataset has a study population of 4240 individuals. There are a total of 16 parameters for every person in the dataset. The details of the parameters are as below:
Categorical parameters: male, education, currentSmoker, BPMeds, prevalentStroke, prevalentHyp, diabetes, TenYearCHD
Continuous parameters: age, cigsPerDay, totChol, sysBP, diaBP, BMI, heartRate, glucose
The parameter *education* has no role in the prediction and so is dropped from analysis. Hence the total number of parameters in the dataset are 14 input and 1 output parameter. The dataset has a total of 12.74% of missing or null values. Entries that contain null or missing values are removed from the dataset. The dataset has a total of 3751 entries after the removal of null entries. The dataset is an imbalanced dataset with the population representation being high for individuals without the heart disease. Out of 3751 individuals, 572 have the disease and 3179 do not have the disease. The following graph shows the distribution.

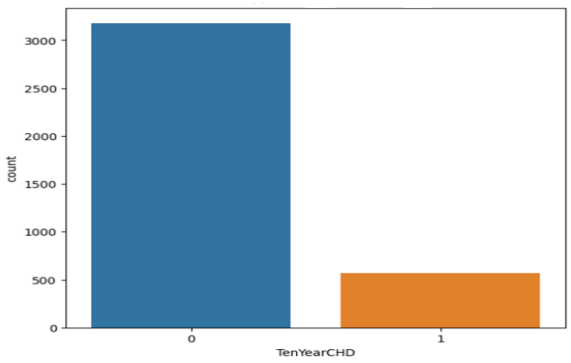


Figure 22: Histogram of the TenYearCHD parameter in Framingham Dataset

To determine the effect of various parameters on the target variable 'TenYearCHD', the correlation matrix has been plotted.

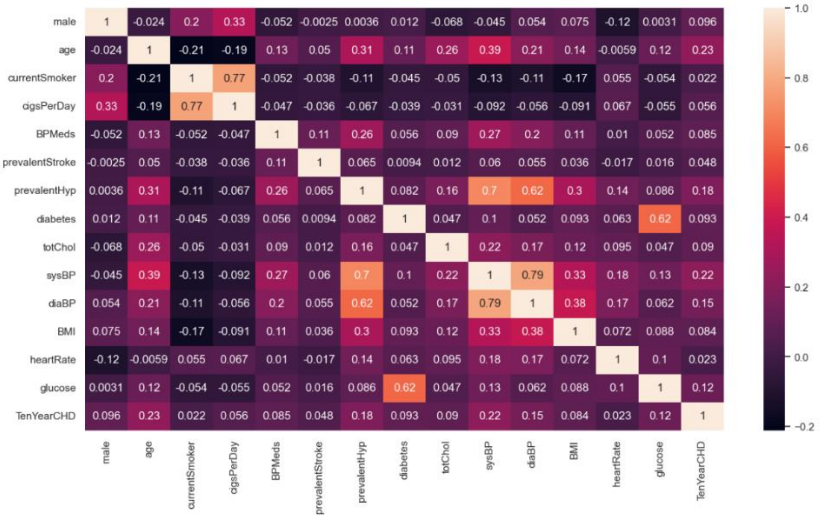


Figure 23: Correlation heatmap of Framingham Dataset

The correlation heatmap shows that none of the parameters have a relation with the variable 'TenYearCHD' of greater than 0.50. Hence for this dataset, training the neural network requires either Feature Selection or Feature Extraction to get more accurate and better performing model. As part of this research, the following two approaches have been tried to address this issue:

- Principal Component Analysis (PCA) for Feature Extraction
- Feature Selection using SelectKBest

Principal Component Analysis (PCA) for Feature Extraction

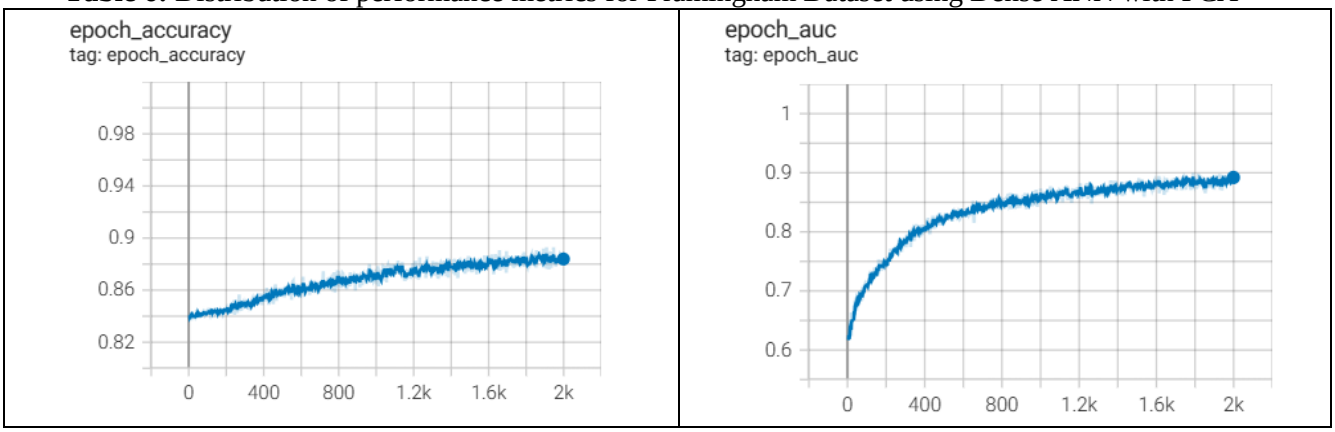
PCA or Principal Component Analysis is a method utilised to reduce the dimensionality of extensive data sets while maintaining essential information. It simultaneously enhances interpretability and reduces the loss of information. It aids in identifying the most important characteristics of a dataset. PCA helps in the discovery of a series of linear variable combinations. This is accomplished by converting the original variables into principal components, that represent a set of new, uncorrelated variables. The variance-based ranking of principal components facilitates efficient feature selection.
Using PCA, the extraction of features has been performed to get a correlation of 95% with the target variable 'TenYearCHD'. This resulted in 5 new features. These features have been used for building the dense neural network. The neural network has the following dense layers:

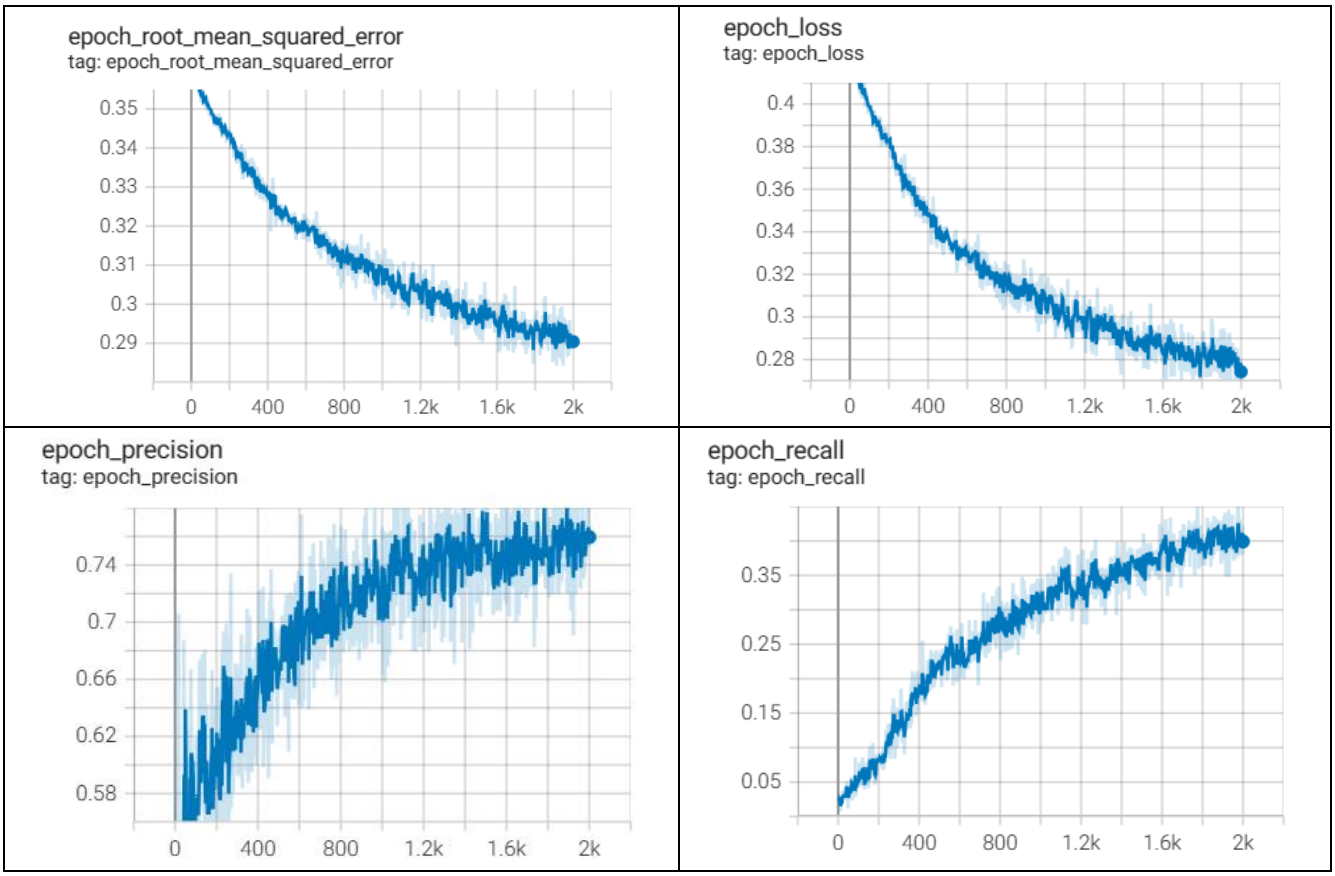
- Input Layer – 5 neurons (activation – ReLU)
- Hidden Layer – 128 neurons (activation – ReLU)
- Dropout – 0.2
- Hidden Layer – 64 neurons (activation – ReLU)
- Dropout – 0.2
- Hidden Layer – 32 neurons (activation – ReLU)
- Dropout – 0.2
- Hidden Layer – 64 neurons (activation – ReLU)
- Dropout – 0.2
- Hidden Layer – 32 neurons (activation – ReLU)
- Dropout – 0.2
- Output Layer – 1 neuron (result) (activation – Sigmoid)

Optimizer – Adam
Loss Function – Binary Cross Entropy
Epochs – 2000
The training of the network has been performed with the training set of size 3000. With a testing set of size 751, the result achieved are as follows.

- Accuracy – 0.88
- AUC – 0.703
- RMSE – 0.32
- Loss – 0.3549
- Precision – 0.44
- Recall – 0.04

Table 6: Distribution of performance metrics for Framingham Dataset using Dense ANN with PCA





Feature Selection using SelectKBest

SelectKBest is among the most frequently utilised feature selection techniques. This is a category of a feature selection method that utilises filters. Filter-based feature selection methods function without reliance on a particular machine learning algorithm to perform the feature selection process. In contrast, the scoring and ranking of the features is determined by statistical measures. SelectKBest ranks and scores the features according to their correlation with the output variable using statistical tests. It then determines which K features received the highest scores and would be incorporated into the final feature subset. When handling large datasets, it is critical to have SelectKBest to rapidly decrease the number of features to a manageable level.

Two parameters comprise SelectKBest: the score function and k. The score function helps to assess the significance of a feature. The numerical value denoted as "K" signifies the maximum number of features to be selected. As part of the research, the score function used is “f_classif” and value of K is 12. The f_classif score function works based on analysis of variance (ANOVA). The algorithm calculates the F-value, which quantifies the linear relationship between two variables, for every feature in relation to the target variable. Features that exhibit a strong reliance on the target variable will receive higher scores.

The Figure 25 is screenshot of the result obtained for the function during analysis. All the features having a score of more than 20 have been taken for the analysis. A total of 10 features have a score exceeding 20. These features include male, age, BPMeds, prevalentHyp, diabetes, totChol, sysBP, diaBP, BMI, glucose.

	Feature_Name	Score
0	male	34.872284
1	age	212.456580
2	currentSmoker	1.767745
3	cigsPerDay	11.760252
4	BPMeds	27.115544
5	prevalentStroke	8.543608
6	prevalentHyp	123.546636
7	diabetes	32.865910
8	totChol	30.349831
9	sysBP	190.876744
10	diaBP	85.114061
11	BMI	26.896986
12	heartRate	1.925700
13	glucose	58.657469

Figure 24: Scores for features using SelectKBest (Framingham Dataset)

The values of the columns having continuous values namely age, totChol, sysBP, diaBP, BMI, glucose have been scaled using MinMaxScaler from Scikit - learn library to improve the accurateness of the model. A dense neural network with the following layers has been built to train with a training set of 3000 records.

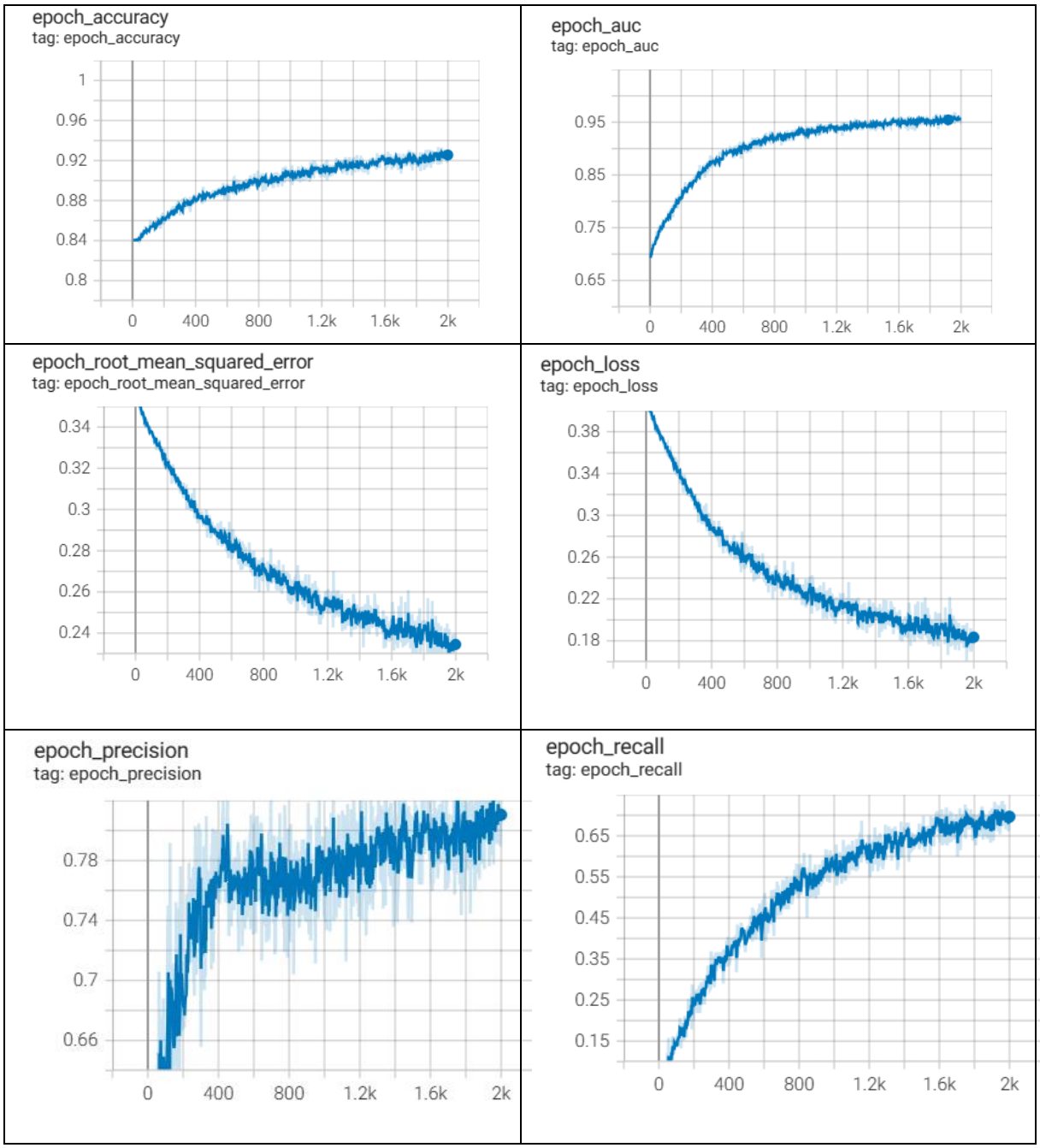
- Input Layer – 10 neurons (activation – ReLU)
- Hidden Layer – 128 neurons (activation – ReLU)
- Dropout – 0.2
- Hidden Layer – 64 neurons (activation – ReLU)
- Dropout – 0.2
- Hidden Layer – 32 neurons (activation – ReLU)
- Dropout – 0.2
- Hidden Layer – 64 neurons (activation – ReLU)
- Dropout – 0.2
- Hidden Layer – 32 neurons (activation – ReLU)
- Dropout – 0.2
- Output Layer – 1 neuron (result) (activation – Sigmoid)

Optimizer – Adam
Loss Function – Binary Cross Entropy
Epochs – 2000

The performance evaluation parameters of the neural network while tested with a testing set of 751 records are as below.

- Accuracy – 0.87
- AUC – 0.72
- RMSE – 0.32
- Loss – 0.3514
- Precision – 0.3
- Recall – 0.03

Table 7: Distribution of performance metrics for Framingham Dataset using Dense ANN with SelectKBest



8. RESULTS AND DISCUSSION:

The simulation study was conducted on HP Laptop with following Hardware and Software Specifications:

Hardware Specification:

Device name: LAPTOP-K9QUO1GT
Processor: 11th Gen Intel(R) Core (TM) i3-1115G4 @ 3.00GHz
Installed RAM: 8.00 GB (7.65 GB usable)
System type: 64-bit operating system, x64-based processor

Software Specification:

Windows Version: Windows 10 Home Single Language
OS build: 19045.3086
Experience Windows Feature Experience Pack 1000.19041.1000.0

The constructed neural network models have undergone evaluation based on several performance indicators, including Accuracy, AUC, RMSE, loss, precision, and recall. The table below presents the results obtained from the Cleveland and Framingham datasets.

Table 8: Performance Metrics of the Neural Network built in the research

Performance Metric	Cleveland Dataset	Framingham Dataset (Feature Extraction using PCA)	Framingham Dataset (Feature Selection using SelectKBest)
Accuracy	0.99	0.88	0.87
AUC	1.0	0.703	0.72
RMSE	0.093	0.32	0.32
Loss	0. 0193	0. 3549	0. 3514
Precision	0.98	0.44	0.3
Recall	1	0.04	0.03

The Cleveland Dataset and Framingham Dataset have been investigated using various machine learning (ML) techniques and the results have been published in our article titled **"Prediction of Coronary Artery Disease using Machine Learning – A Comparative study of Algorithms"** [27]. The results of the ML models from the article published and the results of the DL models for both the datasets have been listed in the table below.

Table 9: Result of ML and DL models for Cleveland Dataset

Model	Accuracy	AUC	RMSE	Precision	Recall
Logistic Regression	0.81	0.872	0.44	0.76	0.84
Decision Tree	0.98	1.0	0.2	1.0	0.912
Random Forest	0.97	0.995	0.12	0.94	0.97
AdaBoost	0.96	0.969	0.231	0.94	0.922
Gradient Boost	0.97	1.0	0.12	0.94	0.97
Extreme Gradient Boost	0.94	0.997	0.12	0.94	0.97
Light Gradient Boost	0.93	1.0	0.12	0.94	0.97
K – Nearest Neighbors	0.97	1.0	0.171	0.94	0.94
Neural Network (ANN)	**0.99**	**1.0**	**0.093**	**0.98**	**1**

Table 10: Result of ML and DL models for Framingham Dataset

Model	Accuracy	AUC	RMSE	Precision	Recall
Logistic Regression	0.66	0.705	0.57	0.65	0.56
Decision Tree	0.75	0.779	0.47	0.76	0.79
Random Forest	0.88	0.945	0.35	0.88	0.86
AdaBoost	0.79	0.845	0.48	0.79	0.72
Gradient Boost	**0.88**	**0.952**	**0.31**	**0.88**	**0.87**
Extreme Gradient Boost	0.88	0.945	0.35	0.88	0.84
Light Gradient Boost	0.88	0.939	0.33	0.88	0.87
K – Nearest Neighbors	0.84	0.878	0.38	0.84	0.87
Neural Network (PCA)	0.88	0.703	0.32	0.44	0.04
Neural Network (SelectKBest)	0.87	0.72	0.32	0.3	0.03

From the above table the following points can be inferred:

- Both datasets show good performance from the Deep Learning (DL) models.
- The results of the neural network model demonstrate superiority over the results produced by machine learning models when applied to the Cleveland dataset.
- With respect to the Framingham dataset, the neural network gives an accuracy value equal to the Gradient Boost model, but the values of AUC and RMSE show that the model performance is lower. This is an imbalanced dataset and so the results are lower compared to the machine learning models.

It can be concluded that the Artificial Neural Network model demonstrated superior performance on the Cleveland dataset, while the Gradient Boosting model excelled on the Framingham dataset.

9. LIMITATIONS AND FUTURE WORK:

The Artificial Neural Network (ANN) has been used to analyse two datasets as part of the study. Even while the network performs remarkably well on both datasets, providing extremely high accuracy, the method's status as a "black box technique" remains a significant drawback. Even if the prediction accuracy is good, this method does not allow for the identification of the main contributing features. The actual dataset features that significantly influence the method's ability to predict whether a person has heart disease have not been defined directly. Additionally, there are not many datasets available for predicting cardiac disease, which limits the scope of research. Exploring the availability of datasets with diverse biological characteristics that may aid in improved disease prediction will be the next stage of this study. The following step will be to investigate Explainable AI techniques, which can improve clinical decision-making and increase transparency in disease prognosis.

10. CONCLUSION:

This research involved the analysis of the Cleveland Dataset and the Framingham Dataset, leading to the development of prediction models utilizing Artificial Neural Networks (ANN). The model for Cleveland Dataset which is a well-balanced dataset with nearly equal representation for both the positive and negative classes, performs extraordinarily well with an accuracy of 0.99 and AUC value of 1. The Framingham Dataset is an imbalanced dataset with very high representation for negative class. The parameters of the dataset exhibit minimal correlation with the target parameter. As part of the investigation, Feature Selection as well as Feature Extraction techniques were employed, and separate models were constructed for each approach. Both the models have given nearly equal accuracy and other performance metrics. Hence it can be stated that ANN performs extremely well for balanced datasets. In case of imbalanced datasets, further performance tuning of the network has to be done to achieve better results.

REFERENCES:

[1] Baillargeon, B., Rebelo, N., Fox, D. D., Taylor, R. L., & Kuhl, E. (2014). The living heart project: a robust and integrative simulator for human heart function. *European Journal of Mechanics-A/Solids*, *48*(1), 38-47. Google Scholar↗

[2] Miao, K. H., & Miao, J. H. (2018). Coronary heart disease diagnosis using deep neural networks. *International journal of advanced computer science and applications*, *9*(10), 1-9. Google Scholar↗

[3] About Coronary Artery Disease (CAD) https://www.cdc.gov/heart-disease/about/coronary-artery-disease.html?CDC_AAref_Val=https://www.cdc.gov/heartdisease/coronary_ad.htm . Retrieved on 15/01/2024.

[4] Kim, J. K., & Kang, S. (2017). Neural Network-Based Coronary Heart Disease Risk Prediction Using Feature Correlation Analysis. *Journal of healthcare engineering*, *2017*(1), 1-13. Google Scholar↗

[5] Amin, M. S., Chiam, Y. K., & Varathan, K. D. (2019). Identification of significant features and data mining techniques in predicting heart disease. *Telematics and Informatics*, *36*(1), 82-93. Google Scholar↗

[6] Jin, B., Che, C., Liu, Z., Zhang, S., Yin, X., & Wei, X. (2018). Predicting the risk of heart failure with EHR sequential data modeling. *Ieee Access*, *6*(1), 9256-9261. Google Scholar↗

[7] Driscoll, A., Barnes, E. H., Blankenberg, S., Colquhoun, D. M., Hunt, D., Nestel, P. J., ... & Tonkin, A. (2017). Predictors of incident heart failure in patients after an acute coronary syndrome: the LIPID heart failure risk-prediction model. *International Journal of Cardiology*, *248*(1), 361-368. Google Scholar↗

[8] Wang, Z., Yao, L., Li, D., Ruan, T., Liu, M., & Gao, J. (2018). Mortality prediction system for heart failure with orthogonal relief and dynamic radius means. *International journal of medical informatics*, *115*(1), 10-17. Google Scholar↗

[9] Vivekanandan, T., & Iyengar, N. C. S. N. (2017). Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease. *Computers in biology and medicine*, *90*(1), 125-136. Google Scholar↗

[10] Bhatikar, S. R., DeGroff, C., & Mahajan, R. L. (2005). A classifier based on the artificial neural network approach for cardiologic auscultation in pediatrics. *Artificial intelligence in medicine*, *33*(3), 251-260. Google Scholar↗

[11] Hannun, A. Y., Rajpurkar, P., Haghpanahi, M., Tison, G. H., Bourn, C., Turakhia, M. P., & Ng, A. Y. (2019). Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, *25*(1), 65-69. Google Scholar↗

[12] Tandon, A., Mohan, N., Jensen, C., Burkhardt, B. E., Gooty, V., Castellanos, D. A., ... & Hussain, T. (2021). Retraining convolutional neural networks for specialized cardiovascular imaging tasks: lessons from tetralogy of fallot. *Pediatric cardiology*, *42*(3), 578-589. Google Scholar↗

[13] Howard, J. P., Fisher, L., Shun-Shin, M. J., Keene, D., Arnold, A. D., Ahmad, Y., ... & Francis, D. P. (2019). Cardiac rhythm device identification using neural networks. *JACC: Clinical Electrophysiology*, *5*(5), 576-586. Google Scholar↗

[14] Krittanawong, C., Virk, H. U. H., Bangalore, S., Wang, Z., Johnson, K. W., Pinotti, R., ... & Tang, W. W. (2020). Machine learning prediction in cardiovascular diseases: a meta-analysis. *Scientific reports*, *10*(1), 16057. Google Scholar↗

[15] 12 Important Model Evaluation Metrics for Machine Learning Everyone Should Know https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/. Retrieved on 25/01/2024.

[16] Metrics to Evaluate your Classification Model to take the Right Decisions. https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/. Retrieved on 25/01/2024.

[17] Receiver operating characteristic. https://en.wikipedia.org/wiki/Receiver_operating_characteristic. Retrieved on 25/01/2024.

[18] Logarithmic Loss. https://emilyswebber.github.io/LogLoss/. Retrieved on 26/01/2024.

[19] What are Neural Networks? https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/. Retrieved on 31/01/2024.

[20] Activation Functions Neural Networks: A Quick & Complete Guide. https://www.analyticsvidhya.com/blog/2021/04/activation-functions-and-their-derivatives-a-quick-complete-guide/. Retrieved on 31/01/2024.

[21] Activation Functions in Neural Networks. https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6. Retrieved on 05/02/2024.

[22] Introduction to Exponential Linear Unit. https://medium.com/@krishnakalyan3/introduction-to-exponential-linear-unit-d3e2904b366c. Retrieved on 05/02/2024.

[23] Es-Sabery, F., Hair, A., Qadir, J., Sainz-De-Abajo, B., García-Zapirain, B., & De La Torre-Díez, I. (2021). Sentence-level classification using parallel fuzzy deep learning classifier. *IEEE Access*, *9*(1), 17943-17985. Google Scholar↗

[24] Analyzing Types of Neural Networks in Deep Learning. https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/. Retrieved on 20/02/2024.

[25] 6 Types of Neural Networks Every Data Scientist Must Know. https://towardsdatascience.com/6-types-of-neural-networks-every-data-scientist-must-know-9c0d920e7fce. Retrieved on 20/02/2024.

[26] Chahal, A., & Gulia, P. (2019). Machine learning and deep learning. *International Journal of Innovative Technology and Exploring Engineering*, *8*(12), 4910-4914. Google Scholar↗

[27] Ramanathan, G., & Jagadeesha, S. N. (2023). Prediction of Coronary Artery Disease using Machine Learning–A Comparative study of Algorithms. *International Journal of Health Sciences and Pharmacy (IJHSP)*, *7*(2), 180-209. Google Scholar↗