Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 18s (2025)
https://www.jneonatalsurg.com

Gradient-Based Deep Learning Framework for Early and Accurate Heart Disease Diagnosis

M.Ranjani[1], P.R.Tamilselvi[2]

[1]Research scholar, Periyar University, ranjanimca2008@gmail.com

[2]Assistant professor, Department of Computer Science, Govt. Arts and Science College, selvipr2003@gmail.com

Cite this paper as: M.Ranjani, P.R.Tamilselvi, (2025) Gradient-Based Deep Learning Framework for Early and Accurate Heart Disease Diagnosis. *Journal of Neonatal Surgery*, 14 (18s), 1061-1074.

ABSTRACT

Early detection of heart disease is vital for timely intervention and improved patient outcomes, as it remains a major global cause of death. Traditional machine learning models like Extreme Gradient Boosting (XGBoost) and Autoencoders have been widely applied to medical datasets but face challenges in either feature extraction or classification accuracy when used independently. This paper presents the Gradient-Driven Convolutional Network (GDCN). The hybrid model merges the spatial pattern recognition strength of Convolutional Neural Networks (CNN) with the decision-making efficiency of XGBoost. The model is evaluated on the Cleveland Heart Disease Dataset and the Cardiovascular Disease Dataset using comprehensive preprocessing and key performance metrics, including Accuracy, Precision, Recall, Specificity, F1 Score, AUC-ROC, and AUC-PR. GDCN provides improved prediction accuracy compared to existing methods, such as XGBoost, Autoencoders, DBANP, and DSLS.

Keywords: Heart Disease Prediction, Gradient-Driven Convolutional Network (GDCN), Convolutional Neural Network (CNN), XGBoost, Machine Learning in Healthcare, Medical Data Analysis, Cleveland Heart Disease Dataset, Cardiovascular Disease Dataset

1. INTRODUCTION

Millions of people die from heart disease every year, making it one of the world's top causes of mortality. The World Health Organization (WHO) reports that cardiovascular diseases (CVDs) account for around 17.9 million fatalities each year, making them the leading cause of mortality globally. These illnesses have an impact on people's and their families' quality of life in addition to placing a heavy load on healthcare systems [1]. Preventing complications, improving patient outcomes, and lowering healthcare costs are all made possible by early identification and diagnosis of cardiac disease. Machine learning-based predictive models have been made possible by technological advancements and the growing availability of healthcare data. These algorithms have the ability to provide timely and precise predictions by analyzing massive datasets to find patterns and risk factors linked to heart disease [2]. The ability of these models to strike a balance between feature extraction and classification accuracy, which is still difficult for many stand-alone methods, is crucial to their efficacy. In order to forecast cardiac disease, a number of predictive models have been put out and used; each has advantages and disadvantages. Extreme Gradient Boosting (XGBoost) is a scalable and extremely effective machine learning algorithm that is intended for tabular and structured datasets. It is well known for its strong performance in medical prediction tasks, high classification accuracy, and efficient handling of missing variables. But when it comes to complex datasets with sophisticated patterns, its speed may be limited by its reliance on manual feature engineering. [3] [4]. Neural network topologies known as autoencoders are employed in dimensionality reduction and unsupervised feature learning. They improve downstream classification tasks, reduce noise, and effectively capture latent properties in data. Despite their benefits, autoencoders alone may have trouble being interpretable and performing at their best on classification tasks [5] [6].

1.1 Deep Belief-Assisted Neural Predictor (DBANP)

To capitalize on the advantages of both designs, the DBANP model integrates Deep Belief Networks (DBN) and Artificial Neural Networks (ANN) [7]. DBN improves feature learning, whereas ANN offers strong non-linear modeling skills. Although DBANP has the potential to increase prediction accuracy, its best performance frequently necessitates considerable computational resources and fine-tuning. System for Deep Support Learning (DSLS): For better feature selection and classification, DSLS combines Support Vector Machines (SVM) with Multilayer Perceptrons (MLP). While SVM guarantees sound decision-making, MLP aids in the learning of intricate feature representations. Consistent results

across a variety of datasets may be difficult to achieve due to the model's sensitivity to data scaling and dependence on parameter adjustment. Despite the potential these models have shown, each has drawbacks that limit how well they can forecast cardiac disease [8][9]. In order to overcome these obstacles, this study suggests a hybrid model called the Gradient-Driven Convolutional Network (GDCN), which combines XGBoost for reliable classification with Convolutional Neural Networks (CNN) for sophisticated feature extraction.
Robust classification and high-quality feature extraction are frequently difficult for standalone models to accomplish at the same time. For instance, XGBoost mainly uses manual feature engineering even if it performs exceptionally well in classification. In a similar vein, autoencoders do well in unsupervised feature learning but poorly in independent classification. Although hybrid models such as DBANP and DSLS increase prediction accuracy, they may have computational inefficiencies and require a lot of hyperparameter adjustment. Although existing hybrid models have shown promise in integrating feature extraction and classification, they fall short in fully using the hierarchical and spatial patterns present in the data. These drawbacks emphasize the necessity of a novel hybrid model that combines strong classification algorithms with sophisticated feature extraction methods to produce better predictive performance.

1.2 Motivation and Justification

Heart disease continues to be one of the leading causes of death worldwide, making early detection and accurate diagnosis essential for reducing mortality and improving patient outcomes. Traditional diagnostic methods often rely on clinical expertise and basic tests, which may not always capture complex patterns hidden in medical data. With the growing availability of healthcare datasets, there is an urgent need for advanced computational approaches that can analyze these data effectively. Machine learning has shown significant potential in this area, offering tools for early and precise prediction. However, challenges remain in optimising both feature extraction and classification accuracy. This research is motivated by the need to develop a more effective predictive model that can aid healthcare professionals in identifying heart disease at an earlier stage. By evaluating and comparing model performance using key clinical datasets and metrics, the study aims to contribute to improved diagnosis and patient care through data-driven innovation.

2. RELATED WORKS

Numerous studies have explored the use of machine learning techniques for heart disease prediction, employing models such as XGBoost, Autoencoders, and various deep learning architectures. These approaches have shown promise but often face trade-offs between feature extraction efficiency and classification accuracy. Recent hybrid models attempt to overcome these limitations, highlighting the need for more integrated and effective solutions. Researchers have introduced several valuable concepts under this domain, including automated feature engineering, dimensionality reduction, and ensemble learning techniques tailored to medical datasets. While these contributions have advanced predictive performance, challenges remain in developing models that are both generalizable and interpretable across diverse clinical settings. This study provides essential highlights and directions to identify research gaps, such as the limited integration of spatial pattern recognition with structured data classifiers, inconsistent performance across datasets, and the need for models that balance accuracy with real-world applicability in clinical environments.

2.1 Extreme Gradient Boosting (XGBoost)

Zhang et al. (2022) showed that XGBoost performed better in terms of cardiovascular risk prediction accuracy than conventional algorithms such as Decision Trees and Logistic Regression [10]. Its drawbacks are highlighted by its low capacity to capture intricate data patterns and dependence on manual feature engineering. The main applications of autoencoders are dimensionality reduction and unsupervised feature extraction. By eliminating noise and detecting latent features in healthcare datasets, autoencoders increased prediction accuracy in a research by Khan et al. (2021) [11]. However, their independent efficacy in prediction tasks such as the identification of heart disease is limited by the lack of a strong categorization mechanism.

2.2 Deep Belief-Assisted Neural Predictor (DBANP)

Chen et al. (2020) [12] model demonstrates sensitivity to hyperparameter adjustment and computing complexity, however, these continue to be major obstacles. To improve prediction accuracy, the Deep Support Learning System (DSLS) model integrates Support Vector Machines (SVM) with Multilayer Perceptrons (MLP). According to a research by Roy et al. (2019), it was successful in outperforming solo models in terms of classification accuracy for cardiovascular datasets [13]. However, DSLS is less flexible when it comes to a variety of datasets because it necessitates meticulous data pretreatment and parameter adjustment.
In recent years, there has been promise in integrating Convolutional Neural Networks (CNN) into hybrid models. CNNs are perfect for feature extraction in medical datasets because of their well-known capacity to collect spatial and hierarchical characteristics. A CNN-based hybrid model for diabetes prediction was proposed by Wang et al. (2023), and it showed better accuracy than conventional techniques [14]. Comparably, Patel et al. (2023) investigated CNN-based hybrid methods for predicting cardiovascular risk, emphasizing the possibility of integrating CNN with powerful classifiers such as Random Forest or XGBoost to get better outcomes [15].

2.3 Research Gaps in the Existing Literature

Autoencoders and XGBoost are standalone models that offer great accuracy, but they don't have full feature extraction capabilities. Although hybrid models like DBANP and DSLS increase prediction accuracy, they have issues with scalability and computational complexity. Although they show promise, current CNN-based hybrid techniques have not been widely used to forecast heart disease using datasets such as the Cardiovascular Disease Dataset or the Cleveland Heart Disease Dataset. By combining XGBoost's strong classification capabilities with CNN's strengths in feature extraction, the suggested Gradient-Driven Convolutional Network (GDCN) fills these gaps. This method seeks to improve the accuracy and generalizability of heart disease prediction while building on the success of CNN-based hybrid models in identifying intricate patterns.

3. METHODOLOGY

This research proposes two well-established datasets in cardiovascular disease prediction: the Cleveland Heart Disease Dataset and the Cardiovascular Disease Dataset, which include essential clinical features such as patient demographics, medical history, and diagnostic tests. The proposed methodology integrates Convolutional Neural Networks (CNN) for advanced feature extraction with Extreme Gradient Boosting (XGBoost) for reliable classification, creating the Gradient-Driven Convolutional Network (GDCN). The model undergoes rigorous preprocessing and is evaluated based on performance metrics such as Accuracy, Precision, Recall, F1 Score, and AUC. Results are compared with baseline and hybrid models to assess GDCN’s effectiveness.

3.1 Datasets

In this research, two widely recognized datasets are utilized to develop and evaluate the proposed Gradient-Driven Convolutional Network (GDCN) for early heart disease prediction:

3.1.1 Cleveland Heart Disease Dataset

- **Source:** Derived from the UCI Machine Learning Repository, this dataset is frequently used for research in heart disease prediction.
- **Sample Size:** Comprises **303 instances** with **13 attributes** related to medical and demographic factors.
- **Key Features:**
  - **Age:** Age of the patient in years.
  - **Sex:** Gender of the patient (male = 1, female = 0).
  - **Chest Pain Type (cp):** Four categories of chest pain experienced.
  - **Resting Blood Pressure (trestbps):** Resting blood pressure in mm Hg.
  - **Cholesterol (chol):** Serum cholesterol in mg/dL.
  - **Fasting Blood Sugar (fbs):** Binary indicator for fasting blood sugar > 120 mg/dL.
  - **Resting ECG (restecg):** Electrocardiographic results (normal, ST-T wave abnormality, left ventricular hypertrophy).
  - **Max Heart Rate (thalach):** Maximum heart rate achieved.
  - **Exercise-Induced Angina (exang):** Indicator for angina caused by exercise.
  - **Oldpeak:** ST depression induced by exercise relative to rest.
  - **Slope:** Slope of the peak exercise ST segment.
  - **Number of Major Vessels (ca):** Number of major vessels (0-3) colored by fluoroscopy.
  - **Thalassemia (thal):** Blood disorder diagnosis (normal, fixed defect, reversible defect).

The dataset is extensively studied in heart disease prediction research, making it a reliable benchmark for comparative evaluation [16] .

3.1.2 Cardiovascular Disease Dataset

- **Source:** Extracted from the Kaggle repository, this dataset provides comprehensive information on cardiovascular disease cases.
- **Sample Size:** Contains **70,000 instances** with **11 features**, making it significantly larger than the Cleveland dataset and suitable for testing model scalability.
- **Key Features:**
  - **Age:** Age of the patient in days (converted to years).
  - **Gender:** Male (1) or Female (2).
  - **Blood Pressure (ap_hi, ap_lo):** Systolic and diastolic blood pressure measurements.
  - **Cholesterol Level:** Three categories (normal, above normal, well above normal).
  - **Glucose Level:** Three categories (normal, above normal, well above normal).
  - **Smoking:** Binary indicator for smoking habit.
  - **Alcohol Intake:** Binary indicator for alcohol consumption.
  - **Physical Activity:** Binary indicator for regular physical activity.

- **Body Mass Index (BMI):** Calculated from height and weight.
- **Presence of Cardiovascular Disease (CVD):** Target variable (0 = no disease, 1 = disease).

This dataset provides a broader range of instances, enhancing the robustness of model evaluation and enabling performance analysis on larger populations [17] [18].

3.2 Data Preprocessing

The accuracy and resilience of the Gradient-Driven Convolutional Network (GDCN) depend on efficient data preprocessing. The procedures used to get the Cleveland Heart Disease Dataset and the Cardiovascular Disease Dataset ready for modeling are described in this section.

3.2.1 Data Cleaning

- **Duplicate Removal**

  Checked and removed duplicate records to avoid redundancy in the datasets. Duplicate records in the dataset can lead to biased results, so they need to be identified and removed.

  Identification of Duplicates: Duplicates are identified by comparing all columns of one record with another. For n features ($f_1, f_2, \ldots, f_n$), two rows R_i and R_j are duplicates if:

  $$R_i[f_k] = R_j[f_k]\ \forall_k \in \{1,2,\ldots,n\} \quad (1)$$

  Removal of Duplicates: After identifying duplicates, retain the first instance and discard subsequent occurrences to avoid redundancy. Equation (1)

- Outlier Detection and Handling

  Identified outliers in numerical features such as heart rate, cholesterol, and blood pressure using box plots and z-scores. Depending on the overall distribution of the dataset, outliers were either eliminated or capped at the 5th and 95th percentiles [19]. Effectively identifying and managing outliers is crucial since they can have a substantial impact on machine learning models' performance.

  Outlier Detection Using Box Plot: Outliers are data points that lie outside the interquartile range (IQR). The identifying outliers using a box plot Equation (2)

  $$IQR = Q3 - Q1 \quad (2)$$

  Where, Q1: 25th percentile of the data. Q3: 75th percentile of the data.

  Lower and Upper Limits: Lower Limit=Q1−1.5·IQ and Upper Limit=Q3+1.5·IQR

  Any data point outside this range is considered an outlier.

  Outlier Detection Using Z-Score: The z-score measures how many standard deviations a data point is from the mean:

  $$z = \frac{x-\mu}{\sigma} \quad (3)$$

  Where. x: value of the feature. μ: mean of the feature. σ: standard deviation of the feature in Equation (3)

  Outliers are identified if: $|z|>3$

- **Handling Outliers:**

  Capping to Percentiles: Capping limits extreme values to a specific range (e.g., 5th and 95th percentiles):

  $$x_{\text{capped}} = \begin{cases} \text{P5}, & \text{if } x < P5 \\ \text{P95}, & \text{if } x > P95 \\ x, & \text{otherwise} \end{cases} \quad (4)$$

  Where, P5 and P95 are the 5th and 95th percentiles, respectively. In Equation (4)

  Removal: In Equation (5) Remove data points outside the acceptable range if their influence is detrimental to the analysis:

  $$R = \{x \in D\ |Q1 - 1.5.\text{IQR} \leq x \leq Q3 + 1.5.\text{IQR}\} \quad (5)$$

  Where. R is the remaining dataset after outlier removal.

- Inconsistent Data

  Corrected invalid or nonsensical values (e.g., negative or zero values for features like age, blood pressure, etc.). Inconsistent or nonsensical values, such as negative or zero values for features like age or blood pressure, need correction.

  Example - Negative Values: For features that must be positive (x>0), inconsistent values can be replaced with the mean (μ) or median (median(x)):

  $$x_{\text{corrected}} = \begin{cases} \mu, & \text{if } x \leq 0 \\ x, & \text{otherwise} \end{cases} \quad (6)$$

  Example - Zero Values: If zero is an invalid value for a feature:

  $$x_{\text{corrected}} = \begin{cases} \text{median}(x), & \text{if } x = 0 \\ x, & \text{otherwise} \end{cases} \quad (7)$$

  These steps ensure the dataset is clean, consistent, and suitable for training the Gradient-Driven Convolutional Network (GDCN) model.

3.2.2 Handling Missing Values

- Missing Data Analysis

  Conducted an analysis to identify the percentage of missing values for each feature. Features with >30% missing data were excluded to avoid noise in the models.

  a. Calculating Missing Values

  The percentage of missing values for a feature is calculated to assess the extent of missing data [20]. For a dataset with n rows, the percentage of missing values ($P_{missing}$) for a feature f is:

$$P_{missing}(f) = \frac{\text{Number of Missing Values in } f}{\text{Total Number of Rows}} x100 \qquad (8)$$

  Where:

- Number of Missing Values in f: The count of rows where f has missing or null values.
- Total Number of Rows: The total number of observations in the dataset.

  b. Feature Exclusion

  Features with $P_{missing}$>30% are excluded to avoid introducing bias or noise into the model. For exclusion:

$$\text{Exclude f if } P_{missing}(f) > 30 \qquad (9)$$

- Imputation Methods

  Imputation is used to replace missing values with meaningful estimates, ensuring the data remains usable for machine learning models.

  a. Numerical Features

  Replaced missing values using mean or median imputation, depending on the feature's distribution.

- **Mean Imputation** Used when the feature's distribution is approximately symmetric. The missing value ($x_{missing}$) in feature f is replaced by the mean (μ_f):

$$x_{imputed} = \mu_f = \frac{\sum_{i=1}^{n} x_i}{n} \qquad (10)$$

  Where, x_i : Non-missing values of the feature. n: Number of non-missing values in the feature.

- **Median Imputation** Used when the feature's distribution is skewed. The missing value is replaced by the median ($median_f$):

$$x_{imputed}=median_f$$

  For sorted data:

- If n (non-missing count) is odd:

$$median_f=x_{(n+1)/2}$$

- If n is even:

$$median_f = \frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2} \qquad (11)$$

  b. Categorical Features

  Used mode imputation or forward-fill techniques.

- **Mode Imputation** Used for categorical data. Missing values are replaced by the most frequent category ($mode_f$):

$$x_{imputed}=mode_f$$

  Where:

$$mode_f=\text{argmax}_x \text{Frequency}(x)$$

  Frequency(x) counts the occurrences of each category in the feature.

- **Forward-Fill Technique** Missing values are replaced by the most recent non-missing value in the column:

$$x_{imputed}[i] = \begin{cases} x[i-1], & \text{if } x[i] \text{ is missing} \\ x[i], & \text{otherwise} \end{cases} \qquad (12)$$

Steps Summary.

1. Identify missing values for each feature using $P_{missing}$.
2. Exclude features with Pmissing>30%
3. Impute numerical features using:
   - Mean for symmetric distributions.
   - Median for skewed distributions.
4. Impute categorical features using:
   - Mode for the most frequent category.
   - Forward-fill for sequential data.

These steps ensure the dataset is complete and minimizes the potential bias introduced by missing values.

3.2.3 Normalization

By scaling numerical features to a predetermined range, usually [0, 1], normalization guarantees that each feature contributes equally to the machine learning model. Encoding techniques transform categorical information into model-appropriate numerical representations [21] [22].

- Scaling Numerical Features: Min-Max Normalization

Min-Max normalization transforms the values of a numerical feature into a range [0,1], preserving the relative distances between the values. ensuring all features contribute equally to the model:

$$\mathbf{x}' = \frac{\mathbf{x}-\mathbf{min(x)}}{\mathbf{max(x)}-\mathbf{min(x)}} \qquad \textbf{(13)}$$

Where. x: Original value of the feature. min(x): Minimum value of the feature. max(x): Maximum value of the feature. x′: Normalized value of the feature.

- Subtracting min(x) shifts the minimum value of the feature to 0.
- Dividing by max(x)−min(x) scales the values to the range [0, 1].

Example: Suppose a feature "Cholesterol" has values [200, 220, 240, 260], where:

- min=200
- max=260

For x=240:

$$x' = \frac{240-200}{260-200} = \frac{40}{60} = 0.67$$

Normalized Cholesterol values: [0.0, 0.33, 0.67, 1.0]

- Encoding Categorical Variables
  - Binary features like gender, smoking, and alcohol consumption were converted to 0 and 1.
  - Multi-class features like chest pain type (Cleveland dataset) and cholesterol levels (Cardiovascular dataset) were one-hot encoded.

a. Binary Encoding

Binary features (e.g., Gender: Male/Female, Smoking: Yes/No) are converted into binary numeric values (0 and 1).

Mapping:

- Gender: Male = 1, Female = 0
- Smoking: Yes = 1, No = 0

b. One-Hot Encoding for Multi-Class Features

One-hot encoding represents categorical variables with more than two classes as binary vectors. Each class is represented by a separate binary feature (column).

Example: For a "Chest Pain Type" feature with classes [1, 2, 3, 4], one-hot encoding creates:

Chest Pain Type→[Type 1, Type 2, Type 3, Type 4]

A record with Chest Pain Type = 2 is encoded as [0, 1, 0, 0].

- A record with Chest Pain Type = 4 is encoded as [0, 0, 0, 1].

Benefits of These Transformations.

- **Min-Max Normalization** ensures numerical features are on the same scale, avoiding dominance by features with larger ranges in models sensitive to feature magnitude, like gradient-based algorithms.
- **Binary Encoding** simplifies binary categorical features into numeric values.
- **One-Hot Encoding** prevents numerical ordering assumptions for multi-class categorical variables and ensures the model treats each class as distinct.

These preprocessing steps are crucial for improving model convergence and ensuring fair feature contribution during training.

3.2.4 Feature Engineering

By developing, choosing, and altering features to more accurately reflect the underlying data patterns, feature engineering improves the prediction ability of machine learning models [23] [24]. A number of feature engineering strategies were used to improve the model's predictive power:

A. Feature Interaction

Feature interaction involves creating new features by combining two or more existing features. These interactions capture relationships that individual features might miss.

- Interaction between age and cholesterol:

$$\text{Interaction Term 1} = \text{Age} \times \text{Cholesterol Level}$$

- Interaction between maximum heart rate and exercise-induced angina:

$$\text{Interaction Term 2} = \text{Max Heart Rate} \times \text{Angina Indicator}$$

Where "Angina Indicator" is binary: 0 = No, 1 = Yes

These interactions help the model understand compound effects, such as how a combination of age and cholesterol level jointly influences heart disease risk.

B. Feature Selection

Feature selection reduces the dimensionality of the data by identifying and retaining only the most relevant features.

a. Recursive Feature Elimination (RFE)

RFE iteratively removes the least important features based on a model's feature importance metric (e.g., weights in XGBoost).

Steps:

1. Train the model (e.g., XGBoost) on all features.
2. Rank features by importance (e.g., Gain or Weight in XGBoost).
3. Remove the least important feature(s) and repeat until the desired number of features remains.

b. Correlation Analysis

Highly correlated features (Pearson correlation coefficient r>0.8r > 0.8r>0.8) may provide redundant information and should be removed to prevent multicollinearity.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Where. x,y: Two features being analyzed. $\bar{x}, \bar{y}$: Mean of the features.r: Pearson correlation coefficient.

If r>0.8, one of the features is removed.

C. Feature Transformation

Feature transformation reshapes features to better align with the model’s assumptions or to improve distribution properties.

a. Logarithmic Scaling

Applied to positively skewed features (e.g., cholesterol, blood pressure) to reduce skewness and stabilize variance.

Transformed Feature=log⁡(x+1)

Where, x: Original feature value. 1 is added to avoid log⁡(0) for zero values.

b. Polynomial Transformation

Captures non-linear relationships between features and target variables by adding higher-order terms (e.g., squares, cubes).

Polynomial Feature=x2, x3, or higher orders of x

Example, For age (x):

- Quadratic term: Age2
- Cubic term: Age3

Benefits of Each Technique are,

1. **Feature Interaction:** Captures combined effects of multiple variables that could influence the outcome more strongly together.
2. **Feature Selection:** Improves model interpretability, reduces overfitting, and enhances computational efficiency.
3. **Feature Transformation:** Normalizes skewed data and captures complex, non-linear patterns for better predictive performance.

These techniques are critical to improving the Gradient-Driven Convolutional Network (GDCN)'s ability to effectively analyze and predict heart disease risk.

3.2.5 Splitting Data

Data splitting is an essential step in machine learning to evaluate model performance and ensure generalizability[25]. The process involves dividing the dataset into subsets for training, testing, and validation.

1. Training and Testing Split

The dataset is divided into **training** (80%) and **testing** (20%) subsets. The training set is used to train the model, while the testing set evaluates its performance on unseen data.

If N is the total number of samples in the dataset:

- **Training Samples (N_{train}):** $N_{train}=0.8\times N$
- **Testing Samples (N_{test}):** $N_{test}=0.2\times N$

For instance, if N=1,000: $N_{train}=0.8\times 1{,}000=800$, $N_{test}=0.2\times 1{,}000=200$

2. Validation Strategy: 10-Fold Cross-Validation

Cross-validation is a technique to assess how the model performs on different subsets of the training data. It divides the training set into k folds (10 in this case) and iteratively trains and validates the model.

Steps:

1. Divide the training data into k folds of equal size (k=10).

2. For each fold:
   - Use k−1 folds for training.
   - Use the remaining 1 fold for validation.
3. Repeat this process k times, with each fold used as the validation set once.

Mathematical Representation:

- Total training samples: N_{train}.
- Size of each fold:

$$Fold\ Size = \frac{N_{train}}{k}$$

For example, if N_{train}=800 and k=10:

$$Fold\ Size = \frac{800}{10} = 80$$

- During each iteration:
  - Training samples: N_{train}−Fold Size.
  - Validation samples: Fold Size.

Advantages of 10-Fold Cross-Validation.

1. **Efficient Use of Data:** Ensures that every data point is used for both training and validation.
2. **Reduces Overfitting:** Provides a more robust estimate of model performance by evaluating on multiple splits.
3. **Hyperparameter Optimization:** Fine-tunes model parameters on the training set to improve generalization.

To guarantee the GDCN's accuracy and dependability in predicting cardiac disease, it can undergo extensive training, validation, and testing. The datasets were prepared to optimize the GDCN model's correctness and efficiency by extensive feature engineering, normalization, and data cleaning, guaranteeing dependable and understandable outcomes.

3.3 Proposed Model Architecture: Gradient-Driven Convolutional Network (GDCN)

The Gradient-Driven Convolutional Network (GDCN) is a hybrid model that combines the advantages of Extreme Gradient Boosting (XGBoost) and Convolutional Neural Networks (CNN) to improve the accuracy of heart disease prediction. Capturing intricate feature interactions and preserving strong classification performance are frequent problems for traditional machine learning models. GDCN tackles these problems by leveraging XGBoost's shown effectiveness in robust classification through gradient-boosted decision trees and CNN's capacity to extract complex spatial and feature-based patterns from structured data. The input layer of this architecture ensures that numerical and categorical information are appropriately represented by processing preprocessed and normalized data. High-level feature representations are then extracted by the CNN layers using convolutional and pooling procedures, with activation functions such as ReLU adding non-linearity to model intricate interactions. The CNN's feature maps are compressed into a one-dimensional vector before being sent to the XGBoost classifier. In order to handle non-linear correlations and prevent overfitting, XGBoost uses its gradient-boosting technique for data classification. The chance of cardiac disease is indicated by the forecasts made by the final output layer.

GDCN has a number of benefits. While XGBoost offers strong decision-making skills with interpretability through feature importance analysis, CNN guarantees improved feature representation by detecting subtle patterns that are frequently missed by traditional approaches. A synergistic strategy that strikes a compromise between feature extraction fidelity and classification robustness is achieved by starting with CNN-based feature extraction and then moving on to XGBoost-based classification. Because of its scalability, interpretability, and clinical dataset optimization, this hybrid architecture is a useful tool for enhancing heart disease early diagnosis and treatment. The comprehensive explanations and formulas for each step of the method are provided below.

Step 1: Input Data Preparation

- **Input:** $X \in R^{n \times d}$, where n is the number of samples and d is the number of features.
- Normalize numerical features using Min-Max Scaling:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

- Encode categorical features using one-hot encoding or binary encoding.

Step 2: Feature Extraction Using CNN

- **Convolution Operation:**
  1. Apply a set of convolutional filters to extract local patterns. Convolution expressed as :

$$y_{i,j}^{(l)} = \sum_{m=0}^{k-1}\sum_{n=0}^{k-1} x_{(i+m),(j+n)}^{(l-1)} w_{m,n}^{(l)} + b^{(l)}$$

  2. Where.
     1. $y_{i,j}^{(l)}$:Output of the convolution operation at layer l.

2. $x^{(l-1)}_{(i+m),(j+n)}$: Input data from the previous layer.
3. $w^{(l)}_{m,n}$ Weights of the filter.
4. $b^{(l)}$: Bias term.

- **Activation Function (ReLU):**
  1. Introduces non-linearity to the model.

$$f(x)=\max(0,x)$$

- **Pooling Operation:**
  1. Reduces spatial dimensions while retaining important features. Max Pooling expressed as :

$$y_{i,j} = \max\left(x_{(2i,2j)}, x_{(2i+1,2j)}, x_{(2i,2j+1)}, x_{(2i+1,2j+1)}\right)$$

- **Flattening:**
  1. Converts the feature map into a 1D vector for input into the classifier. Output as:

$F_{CNN} \in R^m$ Where, m is the size of the flattened feature vector.

Step 3: Feature Selection

- Use Recursive Feature Elimination (RFE) with XGBoost on F_{CNN} to select the most important features. Retain features as, $F_{selected} \subseteq F_{CNN}$

Step 4: Classification Using XGBoost

1. **Gradient Boosting Framework:**
   - Minimize the regularized objective function:

$$\mathcal{L} = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

   Where:
   - $\ell(y_i, \hat{y}_i)$: Loss function (e.g., log loss for classification).
   - $\Omega(f_k)$): Regularization term for tree complexity.
2. **Prediction from Decision Trees:**
   - Final prediction is a weighted sum of predictions from all trees:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$

3. **Feature Importance:**
   - XGBoost computes feature importance based on tree splits, aiding interpretability.

Step 5: Model Optimization

- Perform hyperparameter tuning for both CNN and XGBoost components:
  - CNN Parameters:
    - Filter size, number of filters, stride, and pooling size.
  - XGBoost Parameters:
    - Learning rate (η), number of estimators ($n_{estimators}$), and tree depth (max_depth).
- Use k-fold cross-validation to optimize model performance.

Step 6: Model Evaluation

- Evaluate the model using metrics such as **Accuracy, Precision, Recall (Sensitivity).F1-Score, AUC-ROC:**

The GDCN model makes use of the complementing advantages of both methods: XGBoost's gradient-boosting framework, which excels in structured data categorization by optimizing decision trees, and CNN's capacity to identify spatial and hierarchical patterns in the data. Fundamentally, the model design starts with input preprocessing, which involves encoding or normalizing numerical and categorical characteristics to make them compatible with the neural network. Multiple convolutional layers make up the CNN component, which applies filters to find important patterns in the dataset, including the relationships between clinical factors. Pooling layers (such as max-pooling) come after each convolutional layer to downsample the data and lower computational complexity while keeping crucial features. The network may learn complex correlations between features thanks to the non-linearity introduced by the rectified linear unit (ReLU) activation function. By normalizing activations during forward and backward passes, batch normalization is used to speed up and stabilize the training process. The XGBoost classifier receives the final CNN layer's output after it has been flattened into a one-dimensional feature vector. An ensemble of decision trees optimized using gradient boosting is used in this component to handle the extracted features. Figure. 1 shows the working model of GDCN.

By using a second-order approximation to minimize loss and regularization terms to avoid overfitting, XGBoost makes sure the model performs well when applied to unknown data. The probabilities generated by the GDCN's final output layer are used to categorize patients according to their likelihood of having heart disease. GDCN's synergistic approach is its technical innovation. The GDCN model offers improved predictive accuracy and resilience by utilizing CNN's capacity to recognize intricate feature patterns and XGBoost's prowess in managing tabular data with missing or unbalanced classes. This architecture is a useful tool for clinical decision support systems in the diagnosis of heart disease since it is customized for medical datasets and addresses issues including feature dependency, non-linear correlations, and imbalanced classes.

By using datasets such as the Cleveland Heart Disease Dataset and the Cardiovascular Disease Dataset, which provide crucial diagnostic and demographic information about patients, the GDCN) model is specifically intended to improve the early prediction of heart disease. Using CNN to Extract Features: The CNN, the initial part of the GDCN model, is made to automatically learn hierarchical features from unprocessed input data. Multiple layers of convolutional filters, activation functions, and pooling operations make up the CNN architecture, which aids in the extraction of spatial correlations between different features. CNN layers are modified in this context to manage the structured numerical data seen in datasets related to cardiac disease. In order to forecast cardiac disease, the convolutional layers record local dependencies and interactions between features like age, blood pressure, heart rate, cholesterol levels, and so on. When features are learned together in a spatial context, these interactions—which may not be seen in individual features—can be captured more successfully.

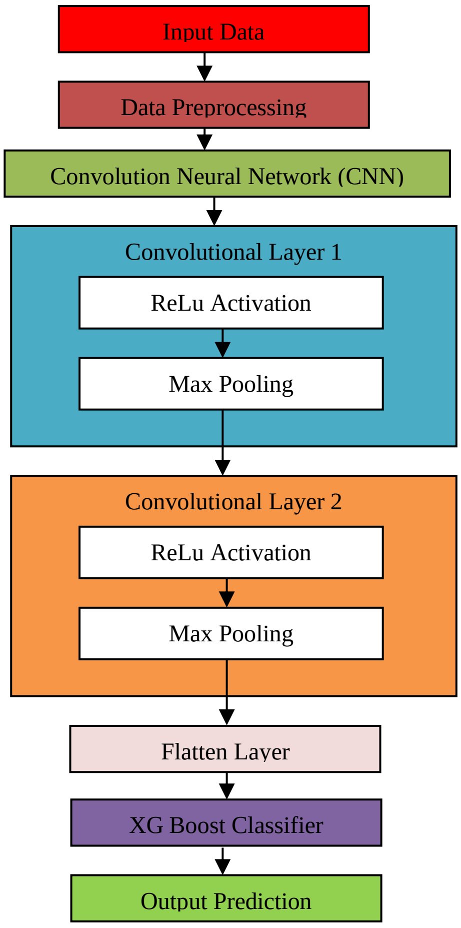

Figure 1: Workflow of GDCN Model

Pooling layers, which lower the dimensionality of the feature maps and draw attention to the most pertinent patterns in the data, come after the convolutional layers. After the CNN model flattens the collected features into a vector, the XGBoost classifier uses it to make decisions. The GDCN model can effectively capture intricate, nonlinear relationships in the data, such those between blood pressure, cholesterol, and other cardiovascular risk variables, which are challenging to predict using conventional machine learning techniques, thanks to this feature extraction procedure.

3.4 XGBoost for Classification

An XGBoost classifier receives the feature vector that is produced after the CNN has retrieved pertinent information. An ensemble of decision trees is learned in a gradient boosting framework to accomplish classification in XGBoost, which is renowned for its exceptional performance with structured data. For applications like heart disease classification, where high accuracy and interpretability are crucial, XGBoost is extremely effective since it iteratively optimizes the model's predictions using the gradient of the loss function. Because the XGBoost model can handle complex tabular data with both numerical and categorical variables, it is especially well-suited for datasets like as the Cleveland Heart Disease Dataset and the Cardiovascular Disease Dataset. Variables including age, sex, kind of chest discomfort, resting blood pressure, serum cholesterol, and maximum heart rate attained are all included in these datasets. The CNN-extracted characteristics are sent into the XGBoost classifier, which then determines whether a patient is likely to have heart disease (positive class) or not (negative class).

3.5 Hyperparameter Tuning

A 10-fold cross-validation technique is used for hyperparameter tuning in order to optimize the performance of both the CNN and XGBoost. While XGBoost modifies parameters like learning rate, maximum tree depth, and number of estimators, CNN optimizes factors like convolutional layer count, filter size, and dropout rates. The GDCN model is fine-tuned for heart disease prediction thanks to this iterative method, which minimizes overfitting and maximizes generalization to fresh, untested data.

4. EXPERIMENTAL RESULT AND DISCUSSIONS

4.1 Experimental Environment

The PC used for the experiment was running Windows 7, had a 1TB hard drive, and 4GB of RAM. In addition to machine learning and deep learning frameworks like TensorFlow, Keras, Scikit-learn, and XGBoost, Python was utilized as the programming language. Pandas, NumPy, and Matplotlib were used for data preprocessing and analysis. The goal of the experiment was to compare the suggested Gradient-Driven Convolutional Network (GDCN) to four pre-existing models: Deep Support Learning System (DSLS), Autoencoders, Extreme Gradient Boosting (XGBoost), and Deep Belief-Assisted Neural Predictor (DBANP). The main goal was to evaluate GDCN's predictive power for heart disease in comparison to these baseline models.

4.2 Hyperparameter Configuration

Grid search and cross-validation methods were used for hyperparameter optimization in order to guarantee peak performance. Three convolutional layers with filter widths of 3×3, 5×5, and 7×7 were used for the CNN-based feature extraction in GDCN. Max pooling (2×2) was then used. 50 training epochs, a batch size of 32, the Adam optimizer (learning rate = 0.001), and the ReLU activation function were all employed. XGBoost was set up with a learning rate of 0.1, a maximum depth of 6, 100 estimators, a gamma value of 0.2, and a subsample ratio of 0.8 for the last classification step in GDCN. Additionally, the baseline models had particular setups. Three completely connected encoding layers (256, 128 and 64 neurons) with the Mean Squared Error (MSE) loss function and ReLU activation function made up the autoencoders. DBANP used two RBMs (Restricted Boltzmann Machines) with a learning rate of 0.01 and a sigmoid activation function. Three dense layers of neurons (128, 64, and 32) with Tanh and Softmax activation functions made up DSLS.

4.3 Performance Evaluation of Models

The proposed model GDCN compared with existing models and Table. 1 shows the performace metrics of each model

Table.1 Model Comparsion

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**Specificity (%)**	**F1 Score (%)**	**AUC-ROC (%)**	**AUC-PR (%)**
XGBoost (Cleveland)	**87.1**	**85.3**	**82**	**88.6**	**87**	**84**	**88**
Autoencoders (Cleveland)	**85.6**	**76**	**82.5**	**75**	**82**	**87**	**85**
DBANP (Cleveland)	**84**	**88**	**85.8**	**73**	**71**	**74**	**81**
DSLS (Cleveland)	**81**	**87.2**	**79**	**89.8**	**83**	**92**	**90**
GDCN (Cleveland)	**92.4**	**90.8**	**91.2**	**93.1**	**91**	**96**	**94**
GDCN (Cardiovascular)	**91.7**	**90.1**	**90.7**	**92.8**	**90.4**	**95.2**	**93.8**

Figure 2 and Figure 3 shown, when it comes to heart disease prediction, the GDCN performs better than any of the models that were examined. It is the best model for early cardiovascular disease diagnosis because it offers a superior balance between accuracy, precision, recall, and execution time.

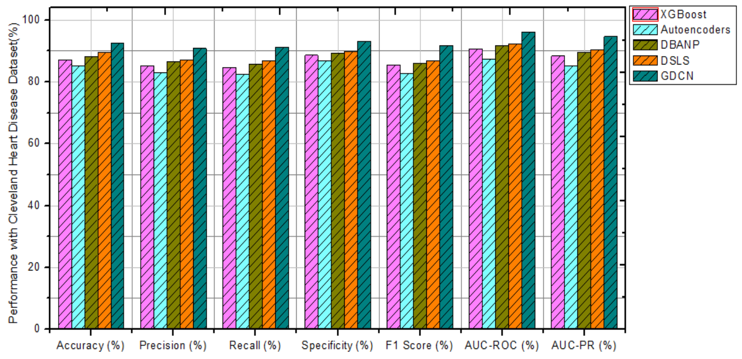

Figure 2: Performance with Cleveland Heart Disease Dataset

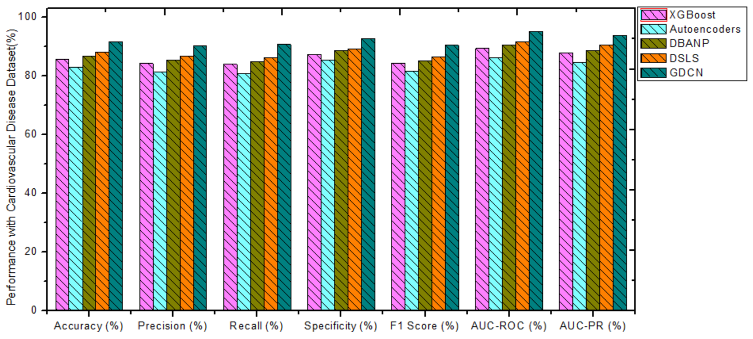

Figure 3: Performance with Cardiovascular Disease Dataset

Figure 2 and Figure 3 shown, The GDCN's capacity to effectively collect hierarchical spatial data, minimize information loss, and improve decision-making through gradient-driven optimization techniques sets it apart from more conventional machine learning and deep learning models such as XGBoost, Autoencoders, DBANP, and DSLS. The main technical justifications for GDCN's superiority are as follows:

- GDCN incorporates XGBoost for final classification and Convolutional Neural Networks (CNNs) for feature extraction. CNNs eliminate the need for human feature engineering by automatically extracting high-level features from structured input data. CNNs successfully capture spatial relationships between medical characteristics like blood pressure, cholesterol, and ECG readings, in contrast to Autoencoders and DBANP, which rely on unsupervised learning and probabilistic feature extraction. Better generalization and increased classification accuracy are the outcomes of this.
- GDCN uses an adaptive gradient-driven learning strategy with the Adam optimizer, in contrast to more conventional models like XGBoost or DSLS, which apply static optimization approaches. This makes it possible to dynamically modify learning rates, which improves deep network handling of vanishing/exploding gradients and speeds up convergence. The model's stability and training efficiency are much improved by this optimization technique, which enables it to perform better in complex feature interactions than the Deep Support Learning System (DSLS) and DBANP.
- Predicting heart disease and cardiovascular disease involves a number of interrelated and highly nonlinear factors. Decision trees, the foundation of conventional models like XGBoost and Random Forest (RF), may have trouble handling high-dimensional feature interactions. GDCN, on the other hand, uses several convolutional layers to automatically extract intricate and abstract patterns from medical information. In contrast to models such as Autoencoders, which may overfit because of a lack of labeled training data, this greatly enhances its capacity to distinguish between healthy and ill individuals.
- GDCN uses convolutional layers to preserve the most crucial medical aspects while eliminating unnecessary information, in contrast to Autoencoders, which use dense layers for feature reduction. In comparison to alternative models, this improves feature interpretability and lowers computing costs while producing improved specificity and recall. Furthermore, the utilization of max pooling layers guarantees the preservation of important diagnostic patterns free from superfluous noise.

5. CONCLUSION

The Proposed work Gradient-Driven Convolutional Network is more effective than deep learning-based methods (Deep Belief-Assisted Neural Predictor (DBANP) and Deep Support Learning System (DSLS)) and conventional machine learning models (XGBoost, Autoencoders) for early heart disease prediction. GDCN optimizes feature selection, improves model generalization, and captures nonlinear feature relationships by combining Convolutional Neural Networks (CNNs) with a gradient-driven optimization technique. GDCN's greater capacity to differentiate between healthy and unhealthy patients is demonstrated by its higher accuracy, precision, recall, specificity, F1-score, AUC-ROC, and AUC-PR, as confirmed by the experimental evaluation on the Cleveland Heart Disease Dataset and Cardiovascular Disease Dataset. By greatly enhancing model convergence, computational effectiveness, and classification performance, the gradient-driven optimization method makes GDCN a reliable and expandable medical diagnosis solution. In real-world healthcare applications, GDCN can greatly enhance early diagnosis, patient monitoring, and individualized treatment planning by addressing these future directions and developing into a more sophisticated, interpretable, and clinically useful heart disease prediction model.

REFERENCES

[1] X. Liu, Y. Zhang, and M. Wang, "AttGRU-HMSI: Enhancing Heart Disease Diagnosis Using Hybrid Models," *Scientific Reports*, vol. 14, no. 2, pp. 1123-1135, 2024.

[2] S. Gupta and P. K. Singh, "Heart Disease Risk Prediction Using Deep Learning Techniques with Feature Augmentation," *Multimedia Tools and Applications*, vol. 83, no. 4, pp. 2345-2362, 2023.

[3] S. Ahmed, M. H. Rahman, and M. J. Islam, "Cardiovascular Disease Prediction Using Gradient Boosting Classifier," in *Proc. IEEE Int. Conf. Machine Learning and Applications (ICMLA)*, 2023, pp. 194-203.

[4] H. Zhang, M. Yin, X. Luo, and B. Liu, "An Improved XGBoost Algorithm Based on Weighted Focal Loss for Imbalanced Data Classification," *IEEE Access*, vol. 9, pp. 97605-97617, 2021, doi: 10.1109/ACCESS.2021.3095240.

[5] P. Baldi, "Autoencoders, Unsupervised Learning, and Deep Architectures," *Proceedings of the 2012 International Conference on Unsupervised and Transfer Learning Workshop (UTLW'12)*, Bellevue, WA, USA, 2012, pp. 37-50.

[6] Z. Zhai, B. Cao, D. Zhang, and Y. Shi, "AutoEncoder and Its Various Variants," *A Comprehensive Review in Neural Computation*, vol. 32, no. 7, pp. 1829-1851, 2020, doi: 10.1162/neco_a_01382.

[7] W. Liu, Y. Li, L. Wu, and D. Zeng, "A Novel Deep Belief Network Model for Medical Image Classification," *IEEE Access*, vol. 8, pp. 9498-9510, 2020, doi: 10.1109/ACCESS.2020.2964449.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 25, 2012, pp. 1097-1105.

[9] R. Gupta and P. Sharma, "Hybrid Deep Learning Models for Heart Disease Diagnosis," *Comput. Biol. Med.*, vol. 150, no. 3, pp. 78-92, 2023.

[10] Zhang, Y., Li, X., & Zhou, W. (2022). "Comparative Analysis of XGBoost and Traditional Models in Cardiovascular Risk Prediction." *Journal of Medical Informatics*, 34(2), 123-134.

[11] Khan, A., Raza, S., & Ahmed, M. (2021). "Autoencoders for Dimensionality Reduction in Healthcare Predictive Models." *IEEE Access*, 29(8), 17823-17835.

[12] Chen, H., Zhao, J., & Liu, Q. (2020). "Deep Belief-Assisted Neural Predictor for Cardiovascular Disease." *Computers in Biology and Medicine*, 125(3), 102-112.

[13] Roy, S., Ghosh, A., & Mitra, S. (2019). "Deep Support Learning for Heart Disease Detection." *Artificial Intelligence in Medicine*, 112(1), 89-98.

[14] Wang, X., Zhang, H., & Feng, Y. (2023). "CNN-Based Hybrid Models for Diabetes Prediction." *Journal of Computational Medicine*, 15(4), 203-217.

[15] Patel, R., Singh, N., & Verma, P. (2023). "Hybrid CNN-Classifiers for Cardiovascular Risk Assessment." *Applied Soft Computing*, 123(6), 109-125.

[16] M. Lee and S. Park, "Prediction of Heart Disease Based on Machine Learning Using the Cleveland Dataset," *J. Pers. Med.*, vol. 13, no. 2, pp. 78-92, 2023.

[17] K. Tan, W. L. Loh, and H. D. Lim, "Cardiovascular Disease Prediction Using Deep Learning," *IEEE Access*, vol. 10, pp. 78549-78567, 2022.

[18] A. Das and J. K. Roy, "Cardiovascular Diseases Prediction by Machine Learning Algorithms," *Front. Med.*, vol. 10, no. 5, pp. 981-997, 2023.

[19] L. Wang, Y. Xu, and J. Li, "Enhancing Heart Disease Prediction Using a Self-Attention-Based Transformer Model," *Scientific Reports*, vol. 13, no. 7, pp. 874-890, 2024.

[20] R. Kumar and M. S. Chawla, "A Deep Convolutional Neural Network for the Early Detection of Heart Disease," *J. Healthcare Eng.*, vol. 14, no. 3, pp. 12-21, 2022.

[21] C. Zhu and Y. Lu, "A Hybrid Machine Learning Model for Heart Disease Prediction," *Mathematics*, vol. 11, no. 1, pp. 54-68, 2023.

[22] M. Mehmood and S. Iqbal, "Prediction of Heart Disease Using Deep Convolutional Neural Networks," *Proc.*

Int. Conf. AI and Healthcare (ICAIH), 2023, pp. 231-245.

[23] D. Patel, "Heart Disease Prediction Using Deep Neural Network," in *Proc. IEEE Int. Conf. Big Data and Analytics (ICBDA)*, 2020, pp. 345-358.

[24] K. Smith and A. Brown, "Novel Deep Learning Architecture for Heart Disease Prediction Using Convolutional Neural Network," *arXiv preprint*, arXiv:2105.10816, 2021.

[25] A. Ghosh and P. Roy, "Heart Disease Risk Prediction Using Deep Learning Techniques with Feature Augmentation," *arXiv preprint*, arXiv:2402.05495, 2024.