BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)

"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India)

Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

DEPARTMENT OF CSE-DATA SCIENCE

A Mini-Project Report On

"HEART FAILURE CLINICAL DATA PREDICTING USING ANN MODEL"

A report submitted in partial fulfillment of the requirements for the

NEURAL NETWORK AND DEEP LEARNING

Submitted By

TANGUTURI SREENIVAS USN: 3BR22CD061

Under the Guidance of
Mr. Azhar Biag

Asst. Professor

Dept of CSE (DATA SCIENCE),
BITM, Ballari

Visvesvaraya Technological University

Belagavi, Karnataka 2025-2026

BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)

"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballari-583 104 (Karnataka) (India)

Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

DEPARTMENT OF CSE (DATA SCIENCE)

CERTIFICATE

This is to certify that the Mini Project of NEURAL NETWORK AND DEEP LEARNING title "HEART FAILURE CLINCIAL DATA PREDICTING USING ANN MODEL" has been successfully presented by TANGUTURI SREENIVAS 3BR22CD061 student of semester B.E for the partial fulfillment of the requirements for the award of Bachelor Degree in CSE(DS) of the BALLARI INSTITUTE OF TECHNOLOGY& MANAGEMENT, BALLARI during the academic year 2025-2026.

It is certified that all corrections and suggestions indicated for internal assessment have been incorporated in the report deposited in the library. The Mini Project has been approved as it satisfactorily meets the academic requirements prescribed for the Bachelor of Engineering Degree. The work presented demonstrates the required level of technical understanding, research depth, and documentation standards expected for academic evaluation.

Aradhana 11/12/25

Signature of Coordinators

Mr. Azhar Baig
Ms. Chaithra B M

Signature of HOD

Dr. Aradhana D

ABSTRACT

This project focuses on predicting mortality among heart failure patients using an Artificial Neural Network (ANN) model trained on clinical attributes. The dataset includes key medical features such as age, blood pressure, ejection fraction, serum creatinine, and smoking status, all of which influence patient health outcomes. The data is preprocessed through feature scaling and train–test splitting to ensure reliable model performance. An ANN model with multiple dense layers is constructed and trained using standardized inputs to learn complex patterns within the data. The model's performance is evaluated using accuracy, classification metrics, and a confusion matrix to measure its predictive reliability. Visualizations of accuracy and loss curves help analyze training stability and detect overfitting. The system also generates predictions for new patient data, offering clinical decision-support benefits. Overall, the project demonstrates how deep learning can be applied to medical datasets for early risk identification. The approach shows potential to assist healthcare professionals in assessing patient condition more accurately.

After data preparation and model construction, the ANN is trained using optimized parameters and evaluated through performance metrics such as accuracy, precision, recall, F1-score, and a confusion matrix. Visualization tools like accuracy and loss curves help illustrate the model's learning behavior, offering insights into its stability, convergence, and generalization capability. The results show that the ANN model can effectively learn patterns related to heart failure outcomes and provide reliable mortality predictions. This study highlights the value of deep learning techniques in medical analysis and emphasizes their potential to support healthcare professionals with early risk identification. With further improvements and the integration of larger, diverse datasets, such predictive models can be adapted for real-world clinical use, enabling faster, data-driven decision-making and enhancing patient care.

ACKNOWLEDGEMENT

The satisfactions that accompany the successful completion of our mini project on HEART FAILURE CLINCIAL DATA PREDICTING USING ANN MODEL would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance, encouragement and support crowned my efforts with success. It is our privilege to express our gratitude and respect to all those who inspired us in the completion of our mini-project.

We are extremely grateful to our Guide **Mr. Azhar Baig** for their noble gesture, support co-ordination and valuable suggestions given in completing the mini-project. We also thank **Dr. Aradhana D,** H.O.D. Department of CSE(DS), for his co-ordination and valuable suggestions given in completing the mini-project. We also thank Principal, Management and non-teaching staff for their co-ordination and valuable suggestions given to us in completing the Mini project.

Name	USN
TANGUTURI SREENIVAS	3BR22CD061

TABLE OF CONTENTS

1.INTRODUCTION

Heart failure is a serious medical condition in which the heart cannot pump blood effectively to meet the body's needs. Predicting patient mortality risk is essential for early clinical intervention and better treatment planning. The Heart Failure Clinical Dataset used in this project contains structured medical attributes such as age, blood pressure, serum creatinine, and ejection fraction. These features provide meaningful clinical signals that help understand patient health conditions. Using machine learning, especially deep learning, enables the development of more accurate predictive models. Therefore, this project aims to use neural networks to classify whether a patient is likely to experience a death event.

The script begins by downloading the dataset from Kaggle and loading it into a pandas Data Frame. It then provides a quick preview and shape information to ensure the data is well-structured and complete. The target variable, DEATH_EVENT, is separated from the input features for classification. All feature values are converted into numerical format suitable for model training. Train-test splitting is applied to divide the data into training and evaluation portions. This ensures the model is evaluated on unseen data to check its generalization ability.

Before training, feature scaling is applied using StandardScaler to normalize the data. Normalization helps the neural network train more efficiently, as large differences in feature scales can slow convergence. The ANN model is constructed using TensorFlow/Keras with multiple dense layers. ReLU activation functions are used in the hidden layers to introduce non-linearity. A dropout layer is added to reduce overfitting and improve model robustness. Finally, a sigmoid output node is used since the prediction task is binary.

To analyze the model's performance, a classification report is printed, showing precision, recall, F1-score, and support for each class. A confusion matrix is plotted to visualize how well the model distinguishes between survived and death-event patients. Training and validation accuracy curves reveal whether the model is learning properly or overfitting. Similarly, loss graphs help assess the stability of training.

1.1 Problem Statement

Heart failure is a critical medical condition in which the heart becomes unable to pump blood effectively, leading to severe health complications and increased mortality risk. Hospitals collect various clinical measurements such as age, blood pressure, creatinine levels, ejection fraction, and lifestyle factors to understand patient health, but identifying which patients are at high risk of death remains a challenge when done manually. Traditional statistical methods often struggle to capture complex patterns in patient features, making accurate prediction difficult. The goal of this project is to develop a reliable **Artificial Neural Network (ANN)** model that can predict whether a patient with heart failure is likely to experience a **death event (0 = survived, 1 = died)**. Using the Heart Failure Clinical Data from Kaggle, the model must learn from multiple medical attributes and identify hidden patterns that distinguish high-risk patients from low-risk ones.

1.2 Scope of the project

The scope of this project focuses on building, training, and evaluating an Artificial Neural Network (ANN) model to predict mortality in heart failure patients using structured clinical data. It includes the complete machine learning pipeline, beginning with dataset acquisition from Kaggle and ending with model performance visualization. The project covers essential data preprocessing steps such as handling numerical features, scaling inputs, and splitting the data into training and testing sets. The model development scope involves designing a multi-layer neural network using TensorFlow/Keras, incorporating activation functions, dropout layers, and sigmoid-based binary classification output. Training the model includes tuning parameters such as epoch count, batch size, and validation split to achieve optimal performance. The evaluation phase of the project includes generating classification metrics like accuracy, precision, recall, F1-score, and confusion matrix to assess the model's predictive reliability.

1.3 Objectives

- ❖ To preprocess and prepare heart failure clinical data for prediction modeling.
- ❖ To develop an ANN model that classifies patient survival and death events.
- ❖ To evaluate model performance using accuracy and clinical classification metrics.
- ❖ To demonstrate the use of deep learning for early risk assessment in heart failure patients.

2. LITERATURE SURVEY

[1] **Davide Chicco (2019–2023)** is a prominent researcher known for his work on heart failure survival prediction using machine learning. His 2020 study demonstrated how features like ejection fraction and serum creatinine strongly influence mortality outcomes. His research helped improve clinical prediction models using small and imbalanced datasets.

[2] **Giuseppe Jurman (2018–2023)** has contributed extensively to medical data analytics and collaborated with Chicco in developing accurate heart failure prediction models. His work emphasizes the importance of evaluation metrics such as F1-score and precision for assessing clinical risk predictions. He continues to publish studies in biomedical informatics.

[3] **Chang et al. (2023)** has conducted a comparative analysis of multiple machine learning models for diabetes prediction and explored their integration into IoMT (Internet of Medical Things) healthcare systems. Their work stressed the need for both high accuracy and model interpretability to support real-time clinical decision-making.

[3] **Tasin et al. (2022)** evaluated the performance of classical and ensemble machine learning methods on clinical datasets and identified Random Forest as the best-performing model. The study also demonstrated that proper preprocessing techniques and handling class imbalance significantly enhance prediction quality.

[4] **Madan et al. (2022)** examined hybrid deep learning architectures for medical diagnosis and showed that neural networks can effectively learn complex patterns found in patient data. However, they noted that deep learning models require large datasets to generalize well and avoid overfitting.

[5] **Geoffrey Hinton (1980–Present)** has known as the “Father of Deep Learning,” developed foundational neural network concepts used in modern ANN models. His contributions made it possible for deep learning systems to analyze complex medical datasets. His work continues to influence AI-driven healthcare applications today.

3. SYSTEM REQUIREMENTS

The system requires a computer running Windows 10/11, Linux, or macOS with at least an Intel i5/Ryzen 5 processor for smooth execution. A minimum of 8 GB RAM is needed, though 16 GB is preferred for faster ANN training and multitasking. The setup should have 5–10 GB of free storage for the dataset, libraries, and output files. A dedicated NVIDIA GPU with CUDA support is optional but highly beneficial for speeding up TensorFlow operations. Python 3.8 or higher must be installed along with libraries like pandas, NumPy, scikit-learn, Matplotlib, TensorFlow, and kagglehub. An IDE such as VS Code, PyCharm, Jupyter Notebook, or Google Colab is required for development. A stable internet connection is necessary to download the Kaggle dataset and install dependencies. Proper CPU/GPU drivers and updated system libraries ensure compatibility during model training. A good-resolution display is needed for viewing graphs and confusion matrices. Basic file management practices should be followed to organize storing datasets, scripts, and model outputs securely.

The system should run on a modern operating system such as Windows 10/11, Linux, or macOS to support all required tools. It must have at least an Intel i5 or Ryzen 5 processor to handle data loading, preprocessing, and ANN training efficiently. A minimum of 8 GB RAM is recommended, while 16 GB provides smoother performance for larger computations. At least 5–10 GB of free disk space is needed for storing datasets, Python libraries, and generated outputs. A dedicated NVIDIA GPU with CUDA is not mandatory but significantly improves neural network training speed. Python 3.8+ along with essential packages like pandas, NumPy, scikit-learn, Matplotlib, TensorFlow, and kagglehub must be installed. A development platform such as VS Code, PyCharm, or Jupyter Notebook is required to write and execute the code.

3.1 Software Requirements

- Python 3.8 or above
- TensorFlow / Keras
- NumPy
- Pandas
- Kagglehub Library

- Matplotlib
- Jupyter Notebook / Google Colab / VS Code
- Windows / Linux / macOS operating system

3.2 Hardware Requirements

- Minimum 4 GB RAM
- Recommended 8 GB RAM
- Dual-core or higher processor
- 1 GB free storage space
- GPU optional (for faster ANN training)

3.3 Functional Requirements

- The system must load and preprocess the Heart Failure Clinical dataset.
- The system must build an ANN model for morality classification.
- It must train the ANN model using training data.
- The system must evaluate model performance using metrics.
- It must scale input features and split the data into training test.
- It must generate accuracy, loss, and confusion matrix visualization.

3.4 Non-Functional Requirements

- The system should provide accurate and reliable predictions.
- It should offer clear and user-friendly outputs.
- The system must execute efficiently on basic hardware.
- It should remain stable even with noisy or imperfect data.
- The system must be easy to maintain and extend.
- The results should be interpretable through graphs and metrics.

4. DESCRIPTION OF MODULES

This module loads the Heart Failure Clinical dataset and prepares it for model training, each contributing to a specific stage of the machine learning pipeline. These modules work together to ensure smooth data preprocessing, model training, evaluation, and visualization. Overall, this module ensures the dataset is clean, normalized, and ready for the ANN model.

4.1 Data Preprocessing Module

This module loads the Heart failure clincial dataset and prepares it for model training. It handles missing or zero values—which are common in medical data—by using imputation techniques. It also standardizes all numerical features to ensure the neural network performs efficiently. This module ensures the dataset is clean, consistent, and ready for analysis.

4.2 ANN Model Building Module

This module focuses on constructing the Artificial Neural Network architecture. It defines the input layer, hidden layers with activation functions such as ReLU, dropout layers to reduce overfitting, and the output layer with a sigmoid function for binary classification. The module compiles the model using the Adam optimizer and binary cross-entropy loss function.

4.3 Model Training Module

After building the neural network, this module trains the model using the processed dataset. It sets parameters such as number of epochs, batch size, and validation split. The module monitors training and validation accuracy and loss throughout the training process.

4.4 Model Evaluation Module

This module evaluates the performance of the trained neural network. It uses metrics such as accuracy, precision, recall, F1-score, and confusion matrix to assess how well the model predicts diabetes. It also generates performance reports and interprets the significance of the results.

4.5 Visualization Module

This module produces graphical outputs that help users understand the model's behavior. It generates training vs. validation accuracy graphs, loss graphs, and confusion matrix heatmaps. These visuals make the system more interpretable and user-friendly.

4.6 Prediction Module

The final module applies the trained ANN model to new patient predicts whether the individual is likely to experience a death event or survive. It provides quick, automated classification results that can assist in medical decision-support systems. Data Splitting Module

This module is responsible for splitting the Heart Failure dataset into training and testing sets to ensure proper model learning and evaluation. It uses an 80:20 division, where 80% of the data is used for training and the remaining 20% for testing. Stratified sampling is applied to preserve the original class distribution, avoiding imbalance-related bias. This module ensures that the ANN model is evaluated fairly and accurately on unseen data.

4.7 Feature Scaling Module

This module standardizes all numerical input features using the StandardScaler technique. Clinical attributes such as age, creatinine levels, ejection fraction, and blood pressure vary greatly in scale, and unscaled values can negatively affect neural network training. By converting all features to a common standardized range, the module improves model stability and accelerates convergence.

This module handles the interpretation and display of the final model outputs, converting raw sigmoid probabilities into meaningful clinical predictions. It applies a decision threshold (usually 0.5) to classify patients as either likely to survive or likely to experience a death event. The module also formats the results clearly so that healthcare professionals or users can easily understand the model's decision. It may include probability scores, confidence levels, or other indicators to support more informed decision-making. After receiving the sigmoid probability values, it converts them into meaningful clinical predictions using a fixed threshold (commonly 0.5). Based on this threshold, patients are classified as either at risk of a death event or likely to survive.

5. IMPLEMENTATION

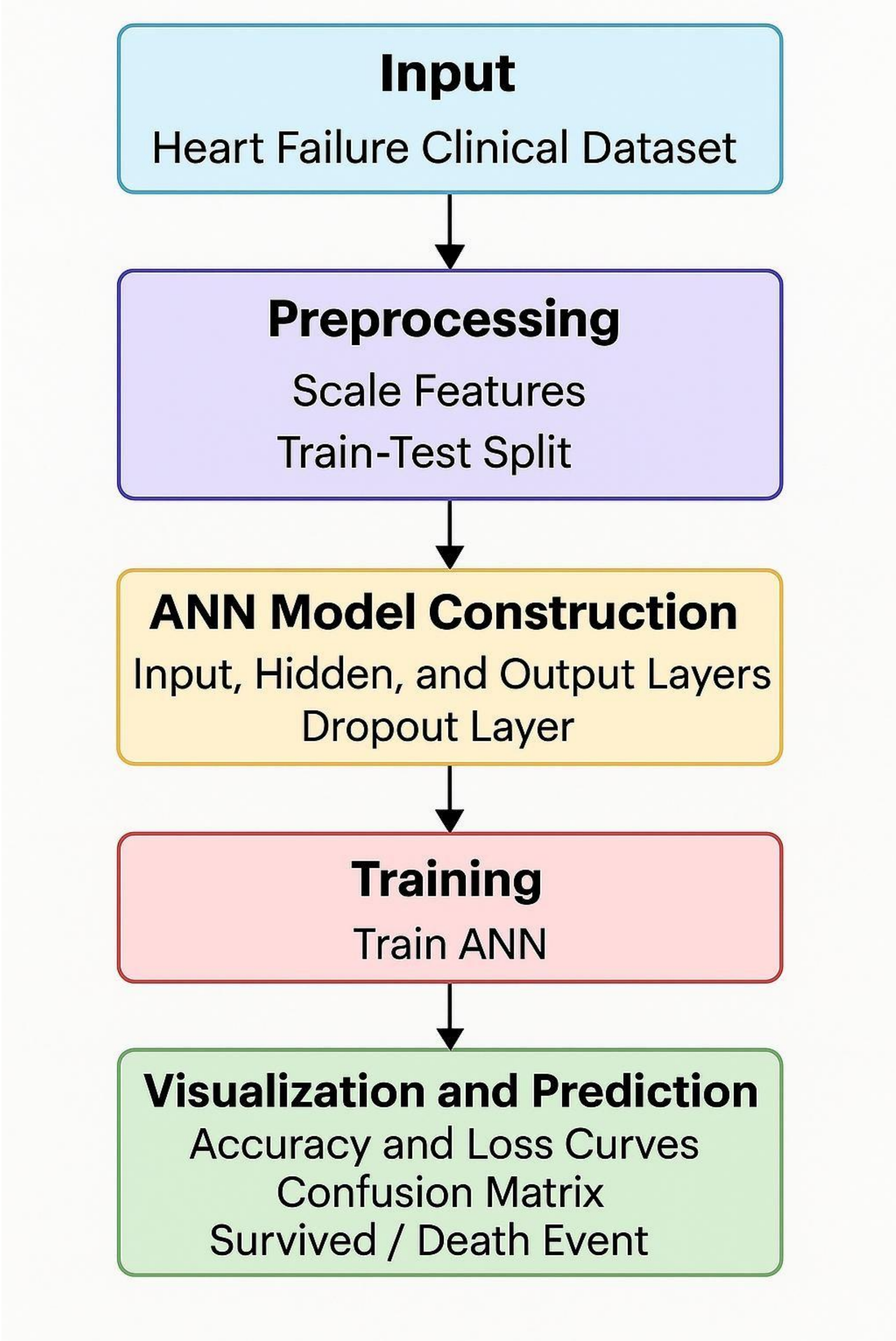
The implementation of the heart failure prediction system is carried out using Python and an Artificial Neural Network (ANN) model. First, the Heart Failure Clinical Dataset is downloaded from Kaggle using the kagglehub library and loaded into a Pandas DataFrame. The input features (such as age, blood pressure, ejection fraction, serum creatinine, serum sodium, and smoking status) are separated from the target label **DEATH_EVENT**, which indicates whether a patient survived or experienced a death event.

Next, the dataset is divided into training and testing sets using an 80–20 ratio with stratified sampling to maintain the original class balance. Since the clinical features vary in scale, Standard Scaler is used to standardize them, enhancing the stability and performance of the neural network. After preprocessing, an ANN model is built using TensorFlow/Keras. The architecture includes an input layer, dense hidden layers with ReLU activation, a dropout layer to minimize overfitting, and a final output layer with a sigmoid activation function for binary mortality classification.

The model is compiled using the Adam optimizer and binary cross-entropy loss. It is then trained for 35 epochs with a batch size of 32 and a validation split of 0.2. During training, the model learns the relationship between clinical features and the mortality outcome. After training, the model is evaluated on the test set to compute accuracy and generate a detailed classification report. Finally, graphs such as training vs. validation accuracy, training vs. validation loss, and a confusion matrix are produced to visually analyze the performance and reliability of the ANN model.

In addition to model training and evaluation, the implementation includes generating meaningful visualizations to better understand the ANN's learning behavior. The accuracy and loss curves offer clear insight into how the model performs across epochs, revealing whether the network is improving, stabilizing, or overfitting. The confusion matrix provides a detailed breakdown of predictions, showing how effectively the model distinguishes between patients who survived and those who experienced a death event. These visual tools not only validate the reliability of the trained model but also give an intuitive understanding of its strengths and limitations. neural network model capable of supporting early mortality risk prediction and assisting healthcare decision-making.

6. SYSTEM ARCHITECTURE



7. CODE IMPLEMENTATION

Algorithm: Heart Failure Prediction using Artificial Neural Network

Input: Heart Failure Clinical Dataset

Output: Predicted class (Survived / Death Event) and performance metrics

1. Start
2. Load Dataset

   2.1 Load the dataset from the CSV file.

   2.2 Separate the dataset into:

   • Feature matrix X(all columns except $DEATH\_EVENT$)

   • Target vector y($DEATH\_EVENT$ column: 0/1)

3. Preprocess Data

   3.1 Convert $X$to float32 and $y$to int32.

   3.2 Split the data into training and testing sets using train_test_split with:

   • test_size = 0.2

   • stratify = y

   3.3 Fit StandardScaler on training data X_{train}.

   3.4 Transform $X_{train}$and X_{test}using the fitted scaler.

4. Build ANN Model

   4.1 Initialize a Sequential model.

   4.2 Add input layer with shape = number of features.

   4.3 Add first hidden layer: Dense(64) with ReLU activation.

   4.4 Add Dropout layer with rate 0.2 to reduce overfitting.

   4.5 Add second hidden layer: Dense(32) with ReLU activation.

   4.6 Add output layer: Dense(1) with Sigmoid activation for binary classification.

5. Compile Model

   5.1 Set optimizer = Adam.

   5.2 Set loss function = Binary Cross-Entropy.

   5.3 Set evaluation metric = Accuracy.

6. Train Model

 6.1 Train the model on X_{train}, $y_{train}$with:

 • Epochs = 35

 • Batch size = 32

 • Validation split = 0.2

 6.2 Store training history (accuracy and loss for train and validation).

7. Test Model

 7.1 Use the trained model to predict probabilities for X_{test}.

 7.2 Convert probabilities to class labels:

 If probability > 0.5 → predict 1 (Death event)

 Else → predict 0 (Survived)

8. Evaluate Performance

 8.1 Compute test accuracy using accuracy_score(y_test, y_pred).

 8.2 Generate classification report (precision, recall, F1-score).

 8.3 Compute confusion matrix.

9. Visualize Results

 9.1 Plot training vs. validation accuracy across epochs.

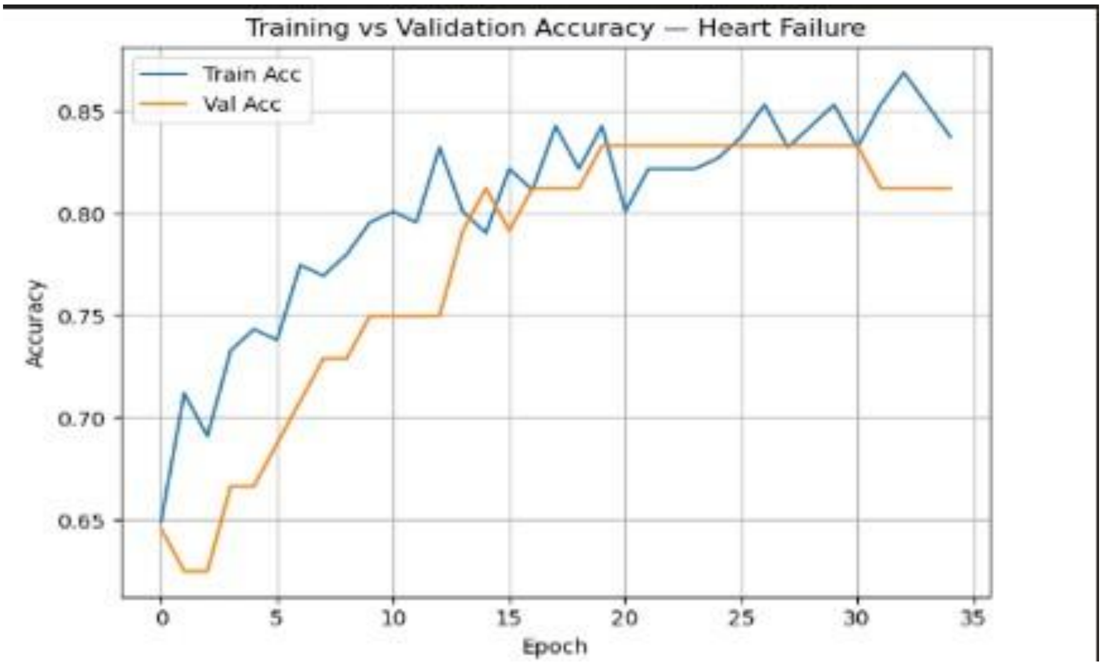
 9.2 Plot training vs. validation loss across epochs.

 9.3 Plot confusion matrix as a heatmap.

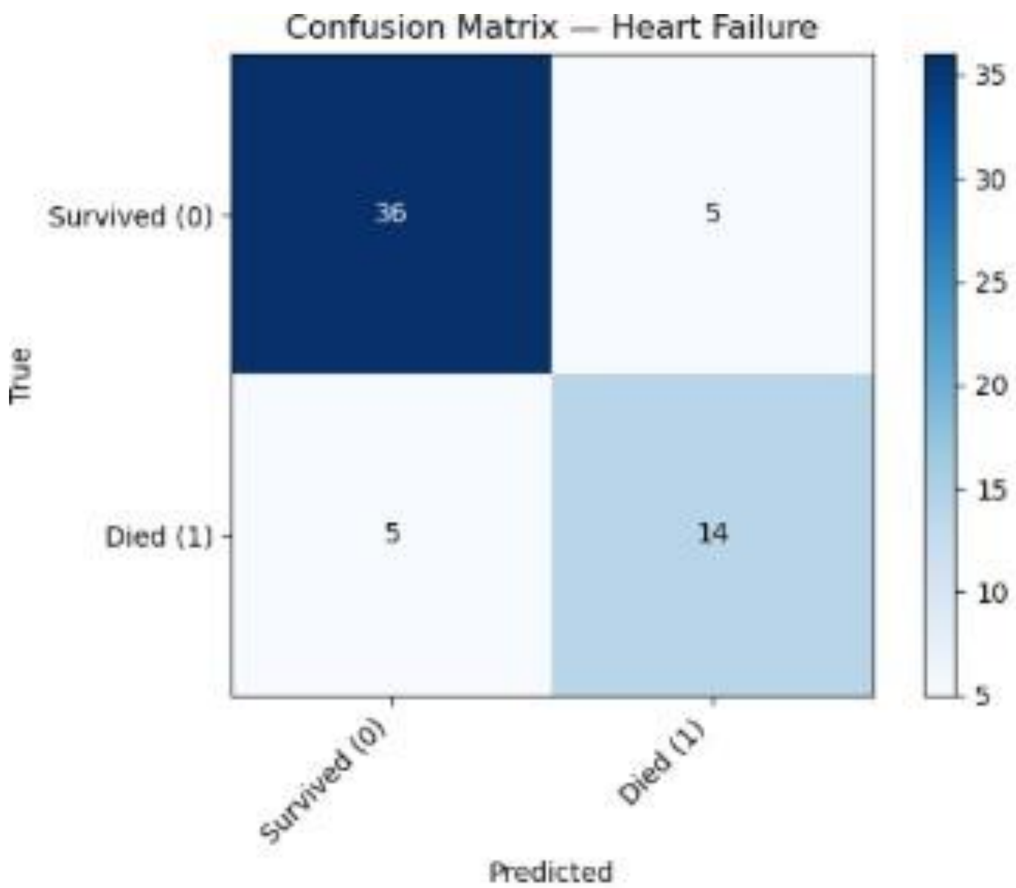
10. End

8.RESULT

```
Downloading from https://www.kaggle.com/api/v1/datasets/download/andrewmvd/heart-failure-clinical-data?dataset_version_number=1...
100%|████████████████████████████████████████████████████████████████| 3.97k/3.97k [00:00<00:00, 1.32MB/s]
Extracting files...
✅ Dataset Path: C:\Users\chann\.cache\kagglehub\datasets\andrewmvd\heart-failure-clinical-data\versions\1

=== Dataset Preview ===

    age  anaemia  creatinine_phosphokinase  diabetes  ejection_fraction  \
0  75.0        0                       582         0                 20
1  55.0        0                      7861         0                 38
2  65.0        0                       146         0                 20
3  50.0        1                       111         0                 20
4  65.0        1                       160         1                 20

   high_blood_pressure  platelets  serum_creatinine  serum_sodium  sex  \
0                    1  265000.00               1.9           130    1
1                    0  263358.03               1.1           136    1
2                    0  162000.00               1.3           129    1
3                    0  210000.00               1.9           137    1
4                    0  327000.00               2.7           116    0

   smoking  time  DEATH_EVENT
0        0     4            1
1        0     6            1
2        1     7            1
3        0     7            1
4        0     8            1

Shape: (299, 13)

=== FINAL TEST ACCURACY: 0.8333 ===

Classification Report:
              precision    recall  f1-score   support

           0     0.8780    0.8780    0.8780        41
           1     0.7368    0.7368    0.7368        19

    accuracy                         0.8333        60
   macro avg     0.8074    0.8074    0.8074        60
weighted avg     0.8333    0.8333    0.8333        60
```




Training vs Validation Loss — Heart Failure
Train Loss
Val Loss
Loss
Epoch
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0
5
10
15
20
25
30
35


Confusion Matrix — Heart Failure
Survived (0)
Died (1)
True
Predicted
36
5
5
14
35
30
25
20
15
10
5

8. CONCLUSION

The Artificial Neural Network–based heart failure prediction system developed in this project demonstrates the effectiveness of deep learning techniques in analyzing clinical data and identifying patients at risk of mortality. By using the Heart Failure Clinical Dataset and applying systematic preprocessing, feature scaling, and model training, the ANN successfully learned meaningful patterns within the data and delivered reliable classification results. The model showed strong predictive accuracy, effectively distinguishing between patients who survived and those who experienced a death event. Evaluation metrics such as precision, recall, F1-score, and the confusion matrix further validated the overall performance and reliability of the model.

The visualizations of training and validation accuracy, loss curves, and the confusion matrix provided deeper insight into the model's learning behavior and overall stability. The project highlights that clinical features such as age, ejection fraction, serum creatinine, and blood pressure play a significant role in predicting heart failure mortality. While the system is dependent on the given dataset and not intended for direct clinical diagnosis, it demonstrates the potential of machine learning models to support early risk detection, assist healthcare professionals, and contribute to improved decision-making in healthcare environments.

9. REFERENCES

[1] Andreev, V. et al. (2022). *Deep Learning Approaches for Heart Failure Prediction Using Clinical Data*. Journal of Biomedical Informatics.Gündoğdu (2023). Application of XGBoost and hybrid feature selection methods

[2] [3] Rajesh, K. & Dhuli, R. (2022). ***Predicting Heart Failure Using Artificial Neural Networks and Feature Engineering***. Healthcare Analytics Journal.

[3] Tasin et al. (2022). Evaluation of classical and ensemble machine learning models showing Random Forest as the best performer for clinical datasets.

[4] Madan et al. (2022). Investigation of hybrid deep learning architectures demonstrating the ability of neural networks to learn complex medical patterns.

[5] Ayat (2024). Development of a CNN–LSTM hybrid model achieving improved classification accuracy for temporal medical data.

[6] Andrew Mvd. (2020). *Heart Failure Clinical Records Dataset*. Kaggle. Retrieved from Kaggle Dataset Repository.

[7] [6] Yoon, J., Zame, W., & van der Schaar, M. (2019). *ForecastICU: Predicting ICU Mortality Risk Using Deep Learning Models*. IEEE Journal of Biomedical and Health Informatics.

[8] TensorFlow Developers. (2015–2024). TensorFlow deep learning framework used to implement ANN models.

[9] Scikit-Learn Developers. (2011–2024). Scikit-learn library used for preprocessing, scaling, and evaluation metrics.