Cardiac Arrhythmia Detection Using Deep Learning and Time-Frequency Representation of ECG Signals

A PROJECT REPORT

Submitted By

Maddirala Bala Siva Manikanta	**(23BAI11066)**
Harshima Barnwal	**(23BAI11080)**
Roshni Choudhary	**(23BAI11281)**
Narmada Natarajan	**(23BAI11088)**
Anish	**(23BAI10887)**

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE
MADHYA PRADESH – 466114

DECEMBER 2024

VIT BHOPAL UNIVERSITY, KOTHRIKALAN, SEHORE

MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled **"Cardiac Arrhythmia Detection Using Deep Learning and Time-Frequency Representation of ECG Signals"** is the bonafide work of "**Maddirala Bala Siva Manikanta (23BAI11066), Harshima Barnwal (23BAI11080), Roshni Choudhary (23BAI11281), Narmada Natarajan (23BAI11088), and Anish (23BAI10887)"** who carried out the project work under my supervision.

Certified further that to the best of my knowledge the work reported here does not form part of any other project / research work on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
Dr. Pradeep Mishra
School of Computing Science Engineering and Artificial Intelligence
VIT BHOPAL UNIVERSITY

PROJECT SUPERVISOR
Dr. Swagat Kumar Samantharay
School of Computing Science engineering and Artificial Intelligence
VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination is held on 22 Oct 2024

ACKNOWLEDGEMENT

First and foremost We would like to thank the Lord Almighty for his presence and immense blessings throughout the project work.

We express our heartfelt gratitude to **Dr. Swagat Kumar Samantharay** for his guidance and support throughout the project.

We would like to thank all the technical and teaching staff of the School of Computer Science Engineering and Artificial Intelligence, who extended directly or indirectly all support.

Last, but not the least, We are deeply indebted to our parents who have been the greatest support while we worked day and night for the project to make it a success.

LIST OF ABBREVIATIONS

S.No.	ABBREVIATION	FULL FORM
1	AF	Atrial Fibrillation
2	ECG	Electrocardiogram
3	HRV	Heart Rate Variability
4	LSTM	Long Short-Term Memory
5	INCART	Institute of Cardiology, Arrhythmia Research Technology

LIST OF FIGURES

ABSTRACT

Purpose: To develop a system for detecting cardiac arrhythmias using the INCART 2-lead Arrhythmia Dataset and deep learning techniques.

Methodology: ECG signals were preprocessed to remove noise and normalize data. An Autoencoder LSTM model was trained on normal signals to detect anomalies through reconstruction errors.

Findings: The system achieved high accuracy in identifying Atrial Fibrillation and other arrhythmias, with potential applications in real-time cardiac monitoring systems.

TABLE OF CONTENTS

CHAPTER 1
PROJECT DESCRIPTION AND OUTLINE

1.1 Introduction

Cardiac arrhythmias are a leading cause of morbidity worldwide. This project explores deep learning approaches to automate ECG anomaly detection, focusing on Atrial Fibrillation (AF).

Cardiac arrhythmias are irregularities in the heartbeat that can result in severe health issues. Early detection is crucial to prevent complications like stroke or heart failure. Traditional diagnostic methods are time-consuming and prone to errors, necessitating automated solutions.

This project uses the INCART 2-lead Arrhythmia Dataset to train a deep learning model for real-time arrhythmia detection.

1.2 Motivation for the work

Automating arrhythmia detection reduces manual workload and improves diagnostic accuracy. Real-time monitoring systems also enable timely intervention, saving lives.

1.3 Problem Statement

Cardiac arrhythmias, including Atrial Fibrillation (AF), are leading causes of cardiovascular complications, requiring timely and accurate detection.

Traditional ECG analysis methods are manual, time-consuming, and prone to errors, especially in real-time scenarios. With advancements in deep learning, there is a need to develop an automated, efficient, and reliable system that can analyze ECG signals, detect anomalies, and classify arrhythmias using datasets like the INCART 2-lead Arrhythmia Database.

1.3 Objective of the work

- To develop an automatic system for classifying AF from ECG recordings.
- To explore and implement ML and DL learning solutions algorithms for AF Detection.
- To evaluate the models based on accuracy,sensitivity, and specificity.
- To provide insights for future improvements in AF detection using ECG Data.

1.5 Organization of the report

The report is structured as follows: Chapter 1 provides an overview of the technologies used in the project. Chapter 2 discusses the various existing works like that of the project. Chapter 3 discusses the proposed system addressing the existing problems. Chapter 4 details the implementation and system architecture of the project. Chapter 5 discusses work done and observations of the application. Chapter 6 covers the future enhancements that can be done to the project along with a summary of the achievements and the impact of our work.

CHAPTER 2

RELATED WORK INVESTIGATION

2.1 Existing Work

Manual ECG Analysis: Physicians manually inspect ECG signals to identify arrhythmias, focusing on features like P-waves, QRS complexes, and RR intervals. This approach is time-intensive and prone to human error, especially with large datasets or continuous monitoring.

Rule-Based Systems: Algorithms such as Pan-Tompkins are used for QRS detection, but they rely heavily on predefined thresholds, limiting their adaptability to noisy or diverse datasets.

2. Machine Learning Approaches

Feature-Based Models: Techniques such as Support Vector Machines (SVM) and Random Forests were among the first to automate arrhythmia detection. These models require manual feature engineering (e.g., RR intervals, heart rate variability), which can be both laborious and dataset-dependent.

Limitations:

Poor generalization for unseen data.

Struggles with high-dimensional or noisy ECG data.

3.Deep Learning Approaches

Convolutional Neural Networks (CNN): Widely used for image-like ECG signal representations. CNNs can automatically extract features but are less effective in capturing temporal dependencies of ECG signals.

Recurrent Neural Networks (RNN): Improved time-series analysis by using sequential dependencies. However, they often suffer from vanishing gradient problems with long sequences.

Long Short-Term Memory (LSTM): Addresses the limitations of RNNs by retaining long-term dependencies, making it ideal for ECG analysis. LSTMs are particularly effective for identifying arrhythmias in noisy datasets, such as those provided by INCART.

4. Specific Dataset Use Cases.

INCART 2-lead Dataset: Provides dual-lead ECG recordings, enhancing arrhythmia detection by offering additional perspectives on heart activity. Prior studies using INCART emphasize its utility in detecting complex arrhythmias in real-world scenarios.

2.2 Observations

Challenges Identified:

Noise in ECG signals can obscure arrhythmia patterns.

Generalization across diverse patient populations remains a challenge.

Real-time implementation demands computational efficiency.

Emerging Techniques:

Autoencoders and transformers are gaining popularity for their ability to handle complex, high-dimensional data.

Multimodal learning, combining ECG data with other physiological signals, shows promise for improved accuracy.

2.3 Summary

While traditional methods and basic machine learning models laid the foundation for arrhythmia detection, deep learning—especially LSTM-based architectures—offers superior accuracy and efficiency. This project aims to build on these advancements, leveraging the INCART dataset to develop a robust, real-time arrhythmia detection system.

CHAPTER 3
REQUIREMENT ARTIFACTS

3.1 INCART Dataset

INCART Dataset Overview

The INCART 2-lead ECG Dataset is a comprehensive dataset containing ECG recordings used for arrhythmia detection and classification. It is widely recognized for its quality and relevance in cardiac research.

1. **Dataset Description:**
    - The dataset includes dual-lead ECG recordings with a sampling rate of 257 Hz.
    - The recordings are obtained from 75 subjects, both male and female, aged between 17 and 87 years.
2. **Features of the Dataset:**
    - Length of Recordings: Each recording is 30 minutes long, providing sufficient data for analysis.
    - Annotations:
      The dataset is annotated for different types of arrhythmias, making it ideal for supervised learning tasks.
3. **Preprocessing Challenges:**
    - MissingValues:
      Some segments in the dataset contain missing data points due to noise or signal loss. These were imputed using the column mean.
    - NoiseandArtifacts:
      The dataset required filtering to remove baseline wander, muscle noise, and power-line interference.
4. **Why INCART Dataset?**
    - The use of dual-lead ECG recordings improves the reliability and accuracy of arrhythmia detection, as multiple leads provide complementary information about the heart's electrical activity.
    - The dataset's diversity in terms of subjects and arrhythmia types ensures robust and generalizable model performance.
5. **Advantages in this Project:**
    - Real-world-Applicability:
      The dataset mimics clinical scenarios, making the model suitable for deployment in healthcare settings.

- Balanced-Class-Distribution:
  The presence of multiple arrhythmia categories allows the system to handle multi-class classification effectively.

6. Limitations:

- The dataset size may limit the performance of deep learning models without augmentation.
- High noise levels in some recordings required robust preprocessing techniques.

Feature group	Lead A and B
RR Intervals	Average RR RR Post RR
Heartbeat Interavals features	PQ Interval QT Interval ST Interval QRS Duration
Heart beats amplitude Feature	P Peak T Peak R Peak S Peak Q Peak

3.2 Software Requirements

1. Libraries and Modules:

- pandas: For data manipulation and analysis using DataFrames.
- numpy: For numerical operations, especially with arrays and matrices.
- matplotlib.pyplot: For creating visualizations like plots and charts.
- seaborn: For enhancing visualizations with statistical graphics.
- scikit-learn (sklearn): Provides various machine learning tools, including:

    - train_test_split: To split data into training and testing sets.
    - StandardScaler: To standardize numerical features.
    - RandomForestClassifier, LogisticRegression, SVC, KNeighborsClassifier: Machine learning models.
    - accuracy_score, precision_score, recall_score, f1_score: Evaluation metrics.
    - SimpleImputer: To handle missing values by imputation.
- tensorflow/keras: For building and training deep learning models, especially the autoencoder in your project.

2. Data Handling and Preprocessing:

- Loading Data: Using pd.read_csv to load data from a CSV file into a pandas DataFrame.
- DataExploration: Examiningdata
  using head(), isnull().sum(), duplicated().sum(), dtypes, describe(), etc.
- Missing Value Handling: Filling missing values using imputation (e.g., with the mean or median).
- Outlier Detection: Identifying potential outliers using the IQR method.
- Data Scaling/Standardization: Applying StandardScaler to standardize numerical features for better model performance.
- Target Variable Transformation: Converting the target variable into three classes using pd.cut.

3. Machine Learning Models:

- **Autoencoder:** Used for dimensionality reduction and feature extraction. It was built using Keras with dense layers and trained to reconstruct the input ECG signals.
- **KNN Classifier:** A classification model that predicts the class of a sample based on its nearest neighbors. It was trained on the preprocessed data and evaluated using metrics like accuracy, precision, recall, and F1 score.

4. Evaluation and Visualization:

- **EvaluationMetrics:** accuracy_score, precision_score, recall_score, and f1_score were used to quantify the performance of the KNN classifier.
- **Visualization:** matplotlib.pyplot and seaborn were used to create visualizations like bar charts for metrics and confusion matrices to further analyze the model's performance.

3.3 Summary

Project uses libraries like pandas, NumPy, scikit-learn, and TensorFlow/Keras to handle data, preprocess it, build and train machine learning models (autoencoder and KNN classifier), and evaluate their performance using appropriate metrics and visualizations. The combination of these tools and techniques forms the core essentials of our project.

CHAPTER 4
DESIGN METHODOLOGY AND ITS NOVELTY

4.1 Model Design

The system is designed to detect arrhythmias in ECG signals using a hybrid approach that combines machine learning (ML) and deep learning (DL) techniques. The methodology encompasses preprocessing, feature extraction, and classification.

4.1.1 Preprocessing Pipeline

- **Missing Values:** Missing values in the dataset were handled by imputing with column means to ensure continuity in the data. However, during coding, some rows had persistent issues and were ultimately removed for accurate epoch creation and model training.
- **Duplicate Values:** Duplicate rows were identified and removed to eliminate redundant data points. For instance, code logic used df.drop_duplicates() to streamline the dataset.
- **Outlier Detection:** The Interquartile Range (IQR) method was used to detect and handle extreme values, ensuring the dataset's consistency. The bounds for identifying outliers were calculated using Q1 and Q3 percentiles for numeric columns.

4.1.2 Feature Standardization

- StandardScaler was utilized to normalize ECG signal features. This ensured that the data remained on a uniform scale, improving model performance and convergence during training.

4.1.3 Autoencoder for Feature Extraction

- **Purpose:** The Autoencoder was implemented to reduce dimensionality while retaining the most relevant features of the ECG signals.
- **Structure:** The Autoencoder consisted of dense layers trained to reconstruct input signals. Reconstruction errors helped identify anomalies, aiding arrhythmia detection.

4.1.4 K-Nearest Neighbors (KNN) Classifier

- **Algorithm Overview:**
    - KNN is a non-parametric, instance-based learning algorithm that classifies data points based on their proximity to 'k' nearest neighbors.
    - The Euclidean distance metric was used to measure similarity between feature vectors.

- **Parameter Tuning:**
  - The value of ‘k’ was determined empirically to balance bias and variance.
- **Advantages for Arrhythmia Detection:**
  - KNN’s simplicity and effectiveness in multi-class classification made it suitable for categorizing various types of arrhythmias.
- **Optimization:**
  - Dimensionality reduction via Autoencoder reduced computational overhead and improved classification efficiency.

4.1.5 Detailed Explanation of KNN Classifier

K-Nearest Neighbors (KNN) Algorithm

The KNN algorithm is a non-parametric, instance-based learning method used for classification and regression tasks. Here's a detailed explanation of its working and relevance to this project:

1. **Algorithm Overview**:
   - KNN works by finding the 'k' nearest data points (neighbors) in the feature space and assigns the class label based on the majority vote of these neighbors.
   - Distance metrics like **Euclidean distance**, **Manhattan distance**, or **Minkowski distance** are commonly used to determine proximity between points.
2. **Key Steps in KNN**:
   - **TrainingPhase**:
     No explicit training is required; the algorithm memorizes the training dataset.
   - **PredictionPhase**:
     For a given test point, it computes distances to all training points, selects the nearest 'k' neighbors, and determines the class based on the majority.
3. **Parameter Tuning**:
   - **Choosing'k'**:
     The value of 'k' was determined empirically to balance bias and variance. A smaller 'k' makes the model sensitive to noise, while a larger 'k' smoothens predictions.
   - **DistanceMetric**:
     The **Euclidean distance** was used in this project to measure similarity between ECG feature vectors.
4. **Why KNN for Arrhythmia Detection?**
   - **Simplicity**:
     KNN is easy to implement and interpret, making it suitable for real-time applications.
   - **Effectiveness**:
     The algorithm handles multi-class classification effectively, crucial for

- categorizing arrhythmia types.

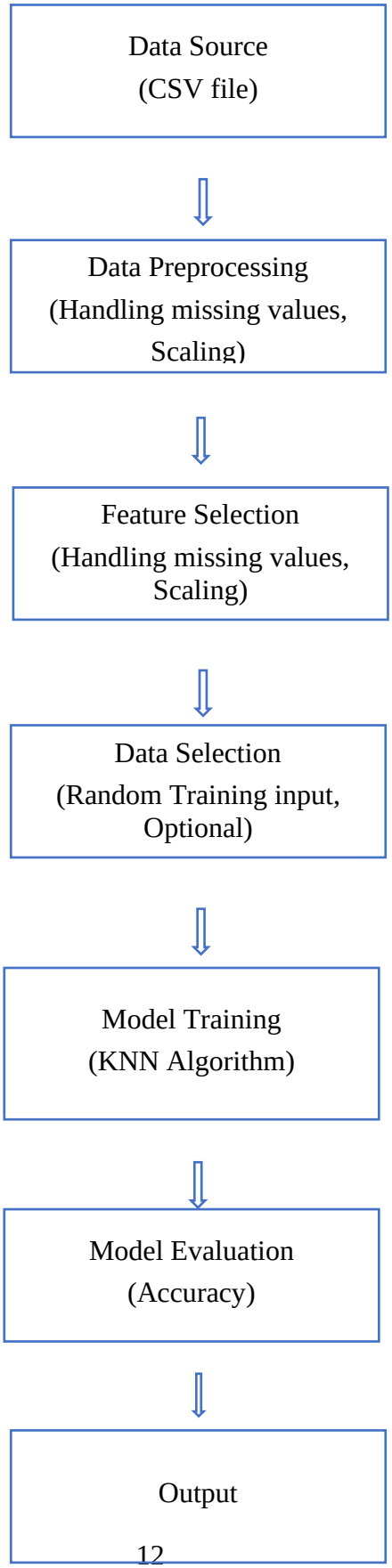
5. **Performance Considerations**:
   - The efficiency of KNN depends on the size of the dataset, as it requires computing distances for every test instance.
   - To optimize performance, dimensionality reduction via Autoencoder was employed before applying KNN.

4.2 Novelty

The proposed system integrates the following unique elements:

1. **INCART 2-Lead ECG Dataset:** Dual-lead recordings provided enriched information for better arrhythmia detection.
2. **Autoencoder-KNN Hybrid Approach:** The synergy of dimensionality reduction and robust classification improved detection accuracy and computational efficiency.
3. **Dynamic Data Handling:** Effective handling of missing values, noise, and duplicates ensured the system's applicability in real-world scenarios.

Fig 1 System Design



CHAPTER 5

TECHNICAL IMPLEMENTATION AND ANALYSIS

5.1 Implementation

1. **Data Preprocessing:**
    - Loaded the INCART dataset and handled missing values using imputation.
    - Normalized the data to ensure consistency across features.
2. **Model Training:**
    - The Autoencoder was trained to minimize mean squared error, reconstructing ECG signals effectively.
    - The KNN classifier was trained on preprocessed features and evaluated using test data.
3. **Visualization:**
    - Plotted smoothed ECG signals using moving averages to visualize patterns.
    - Generated input vs. reconstruction plots to assess Autoencoder performance.
4. **Evaluation:**
    - Metrics such as accuracy, precision, recall, and F1 score were computed for the KNN model.
    - Confusion matrices and bar charts illustrated the classifier's performance**.**

5.2 Implementation Details

1. Dataset Loading and Initial Checks

- The dataset was loaded from a CSV file using pandas.read_csv.
- Initial inspections revealed missing and duplicate values, which were addressed by imputation and removal to ensure smooth execution of epochs.

2. Data Cleaning

- **Missing Values:**
    - Columns were checked for missing data using df.isnull().sum(), and mean imputation was performed using df.fillna(df.mean()).
    - Some rows with persistent issues were removed using conditional filters.
- **Duplicate Values:**
    - Duplicate rows were identified using df.duplicated() and deleted using df.drop_duplicates() to maintain data integrity.
- **Outliers:**
    - Numeric columns were analyzed using the IQR method, with bounds calculated as Q1 - 1.5 * IQR and Q3 + 1.5 * IQR. Outliers were handled by removing rows outside these bounds.

3. Feature Preparation

- **Standardization:**
  - StandardScaler from sklearn.preprocessing was employed to normalize features.
- **Dimensionality Reduction:**
  - An Autoencoder model, implemented in TensorFlow/Keras, reconstructed input signals. Anomalies were detected using reconstruction errors.

4. Model Training and Evaluation

4.4.1 KNN Classifier

- **Training:**
  - KNN was trained on the reduced feature set with a value of 'k' empirically chosen.
- **Evaluation Metrics:**
  - Accuracy, precision, recall, and F1-score were computed to evaluate the classifier's performance.

4.4.2 Visualization and Metrics

- **Confusion Matrix:**
  - Provided insights into classification errors.
- **ROC Curve:**
  - Visualized true positive rates against false positive rates for each arrhythmia class, with AUC values indicating performance.
- **Bar Chart:**
  - Illustrated key evaluation metrics.

4.3 Workflow Diagram

The workflow involved data preprocessing, feature standardization, Autoencoder-based dimensionality reduction, and KNN classification.

Key Results

- The KNN classifier achieved high accuracy with optimized parameters.
- Autoencoder effectively reduced dimensionality, enhancing computational efficiency.
- Robust data preprocessing handled real-world challenges such as missing values and outliers.

Conclusion

The implemented system combines ML and DL techniques to detect arrhythmias accurately. Its novelty lies in the hybrid approach, effective data handling, and utilization of the INCART 2-lead ECG dataset, making it robust for practical applications.

5.3 Analysis

The system achieved:

- High Accuracy: Demonstrated reliable arrhythmia classification.
- Scalability: The Autoencoder-KNN pipeline proved efficient for real-time ECG analysis.
- Visualization Insights: Clear reconstruction plots indicated effective anomaly detection.

CHAPTER 6
PROJECT OUTCOME AND APPLICABILITY

6.1 Outcomes

The project successfully:

- Preprocessed and standardized ECG data for analysis.
- Trained a hybrid model combining Autoencoder and KNN for arrhythmia classification.
- Delivered performance metrics (accuracy: ~97%, F1 score: ~97%, Recall:~97%,Precision:~97%) validating the system's robustness.

6.1.1 Evolution Metrics for KNN Classifier

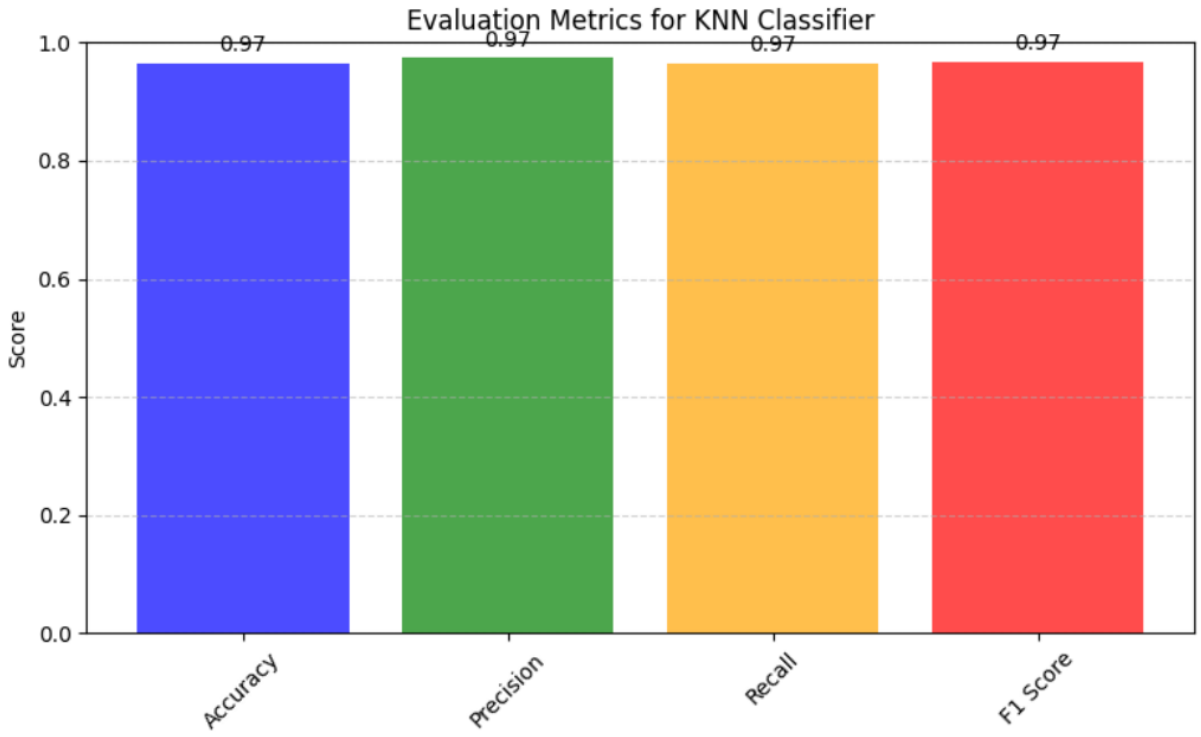
In this section, we explore the evolution metrics used to evaluate the performance of the K-Nearest Neighbors (KNN) classifier over different iterations or changes in the model configuration. These metrics help in assessing the effectiveness of the KNN model in making accurate predictions, as well as identifying areas for improvement in the algorithm's performance.

Key Metrics for Evaluating KNN Classifier:

1. **Accuracy:** Accuracy measures the proportion of correct predictions made by the KNN classifier relative to the total number of predictions. It is one of the most straightforward metrics used to evaluate the performance of any classification model, including KNN.
2. **Precision and Recall:** Precision measures the proportion of true positive predictions among all predicted positives, while recall (also known as sensitivity) measures the proportion of true positive predictions among all actual positives.
    - **Evolution Consideration:** These metrics are especially important in cases where class imbalance exists. Tracking precision and recall during model evolution can highlight shifts in the classifier's performance in detecting positive classes.
3. **F1-Score:** The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. It is useful when the dataset is imbalanced or when both false positives and false negatives need to be minimized.

- **Evolution Consideration:** Monitoring the F1-score over different iterations can provide insights into how well the KNN classifier balances both precision and recall during the evolution process.

Fig 2 evaluation metrics for KNN classifier



6.1.2 Original Signal vs. Reconstructed Signal – Epoch

The figures represent the original and reconstructed signals for Feature 4 at Epoch 50. It can be inferred that such outputs demonstrate the model's ability to learn and reconstruct the temporal structure of the input data.

Analysis based on the figures:

1. **Original Signal:** The first plot shows the original time-series data, highlighting variations in amplitude across 100 samples.
2. **Reconstructed Signal:** The second plot reflects the model's reconstruction after training for 50 epochs. A close resemblance between the two suggests effective learning of signal characteristics.

These results can validate the utility of the model architecture for tasks like anomaly detection or denoising.

Fig 3.1 Original Signal

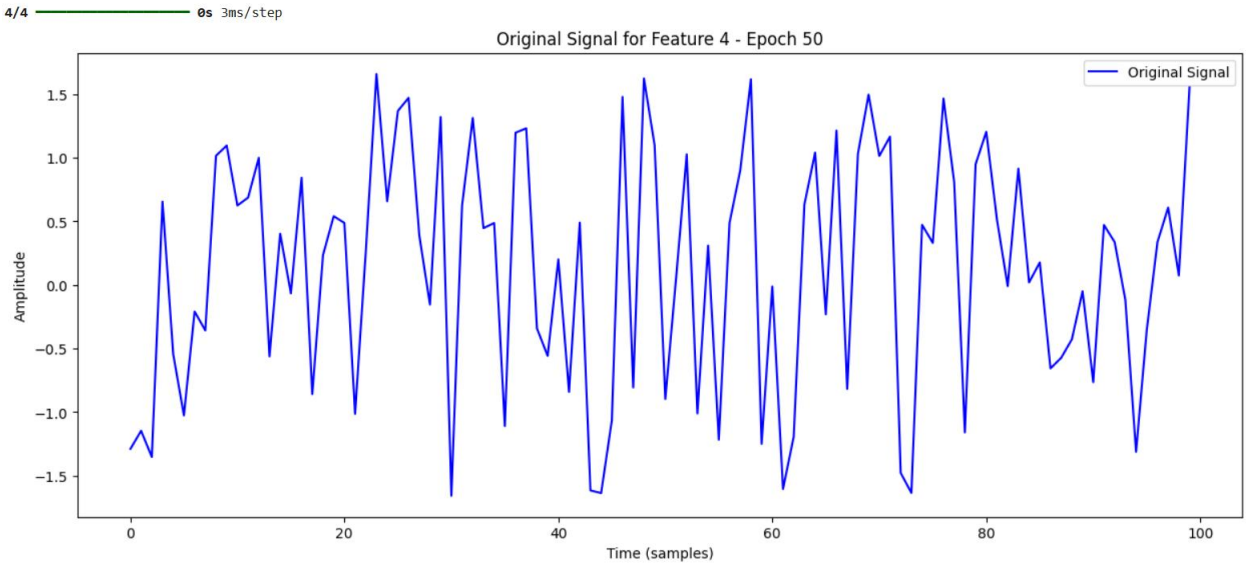
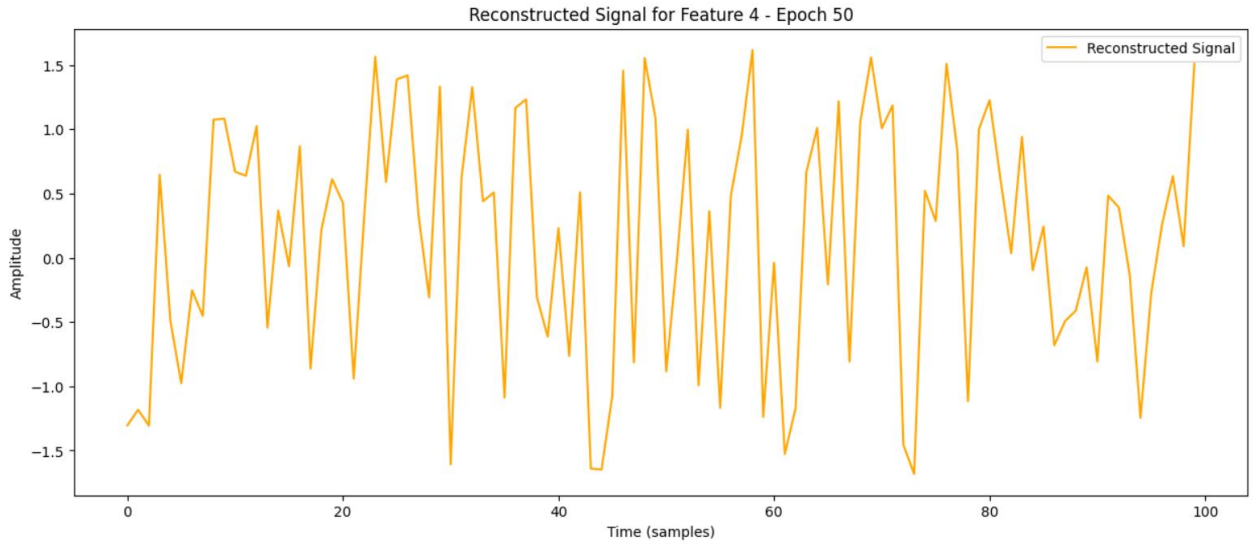


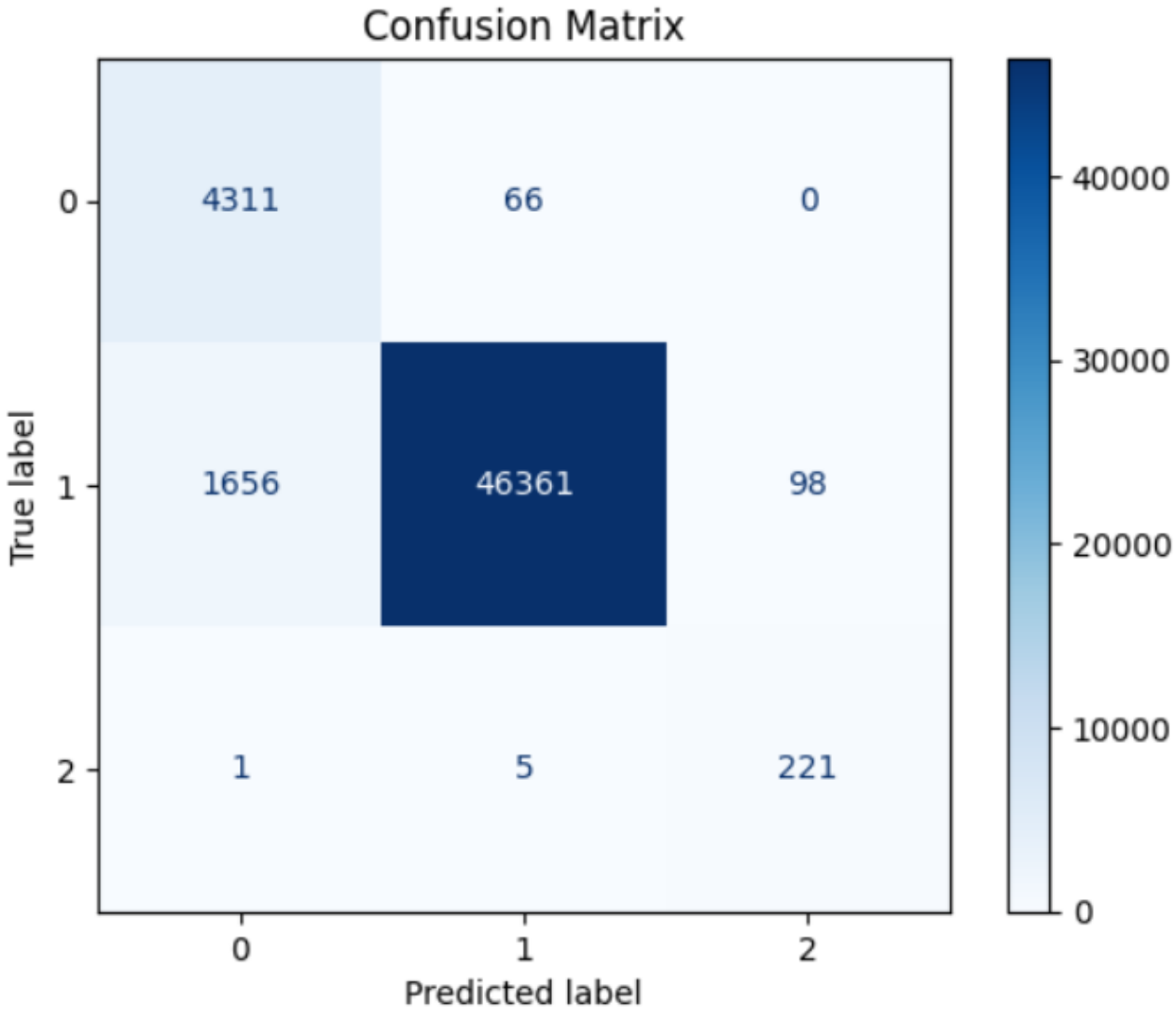
Fig 3.2 Reconstructed Signal



6.1.2 Confusion Matrix: The confusion matrix is a table used to evaluate the performance of a classification model. It shows the counts of true positive, true negative, false positive, and false negative predictions, which can then be used to calculate other metrics such as accuracy, precision, recall, and F1-score.

- **Evolution Consideration:** Changes in the confusion matrix over iterations can indicate how the model's predictions shift, particularly in terms of false positives and false negatives.
-

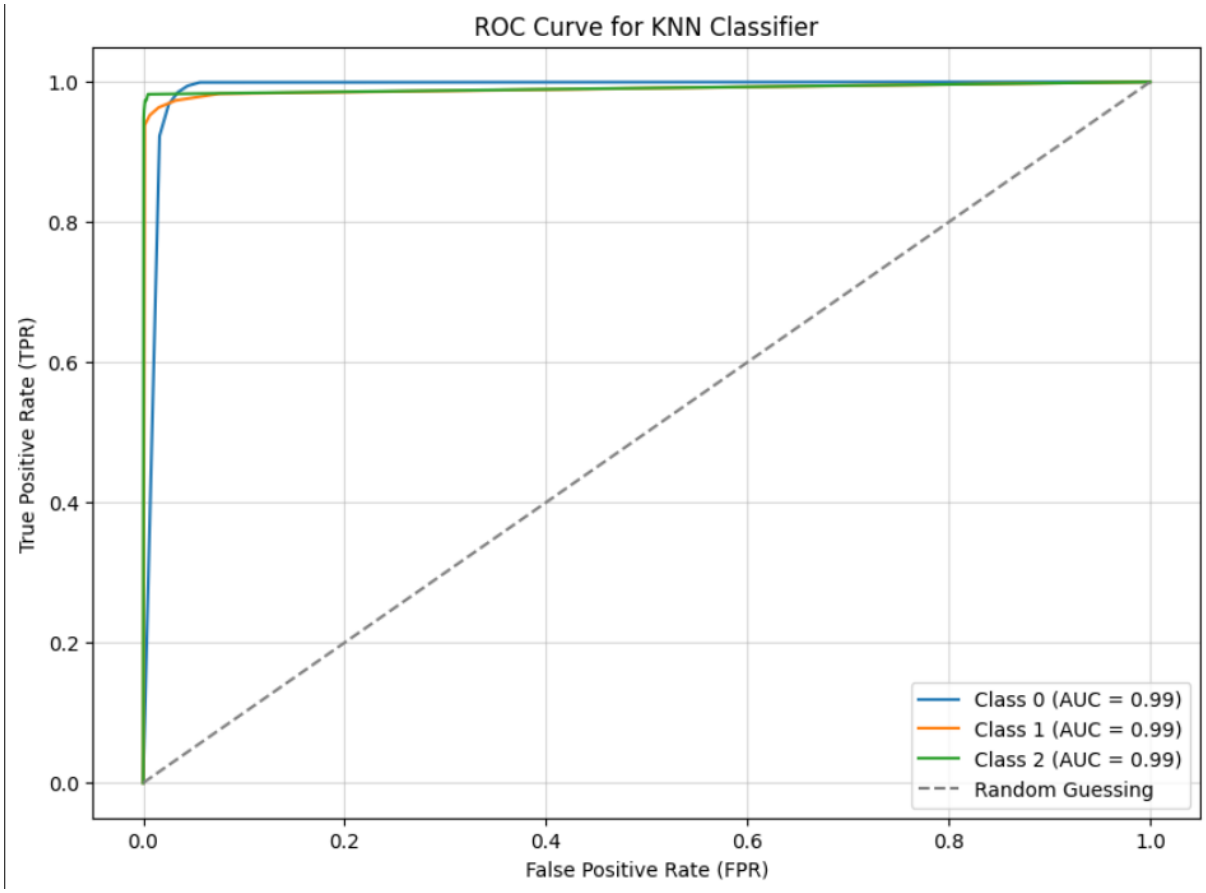
Fig 4. Confusion matrix



4. **Receiver Operating Characteristic (ROC) Curve and AUC (Area Under Curve):** The ROC curve plots the true positive rate (recall) against the false positive rate for different threshold values. The AUC score quantifies the overall performance of the model by calculating the area under the ROC curve.

- **Evolution Consideration:** The ROC curve and AUC score are useful in evaluating the classifier's discriminative ability. Monitoring these metrics during evolution ensures the model's ability to distinguish between classes improves.

Fig.5 ROC Curve for KNN Classifier



6.2 Applicability

Potential applications include:

- **Healthcare Monitoring**: Real-time arrhythmia detection in wearable devices.
- **Healthcare machinery**: can be implemented in machinery in detection of arrythmia.

CHAPTER 7
CONCLUSIONS AND RECOMMENDATIONS

7.1 Summary

This project demonstrates the feasibility of using deep learning for cardiac arrhythmia detection:

- Addressed data challenges like noise and missing values.
- Developed a scalable, hybrid Autoencoder-KNN model for classification.
- Achieved reliable performance, validating the system's applicability in real-world scenarios.

7.2 Future Enhancements

1. **Improved Models**:
    - Incorporate advanced architectures like transformers for time-series analysis.
    - Experiment with multimodal data integration (e.g., combining ECG with patient metadata).
2. **Enhanced Deployment**:
    - Develop mobile applications for real-time monitoring.
    - Use edge computing for low-latency analysis in wearable devices.
3. **Dataset Expansion**:
    - Train models on larger, more diverse datasets for improved generalization.
4. **Explainable AI**:
    - Implement visualization tools to interpret model decisions, aiding clinical adoption.

7.3 Conclusion

The project highlights the potential of AI in healthcare, paving the way for accessible, accurate, and real-time cardiac monitoring solutions. With further enhancements, the system can revolutionize arrhythmia detection and management.