DEEP LEARING POWERED ARRHYTHMIA DIAGNOSIS AND TREATMENT WITH NOISE DETECTION

AUTHORS:

1. Palle Pranay Reddy 2. Gurram Meghana 3. Sanaka Mounika

B.Tech., 4th year students, Department of CSE – Data Science, Sphoorthy Engineering College.

pranayreddy9010@gmail.com, gurrammeghana78@gmail.com, mounikasanaka123@gmail.com

4. Ichangi Raghavendra (Assistant Professor)

rmichangi@gmail.com

ABSTRACT

This paper presents an advanced approach to cardiac arrhythmia detection using deep learning models, specifically convolutional neural networks (CNN) and long short-term memory (LSTM) algorithms, to analyze electrocardiogram (ECG) signals. Building upon existing research, our methodology focuses on enhancing accuracy, efficiency, and data preprocessing techniques to improve diagnostic outcomes. We begin by integrating CNN and LSTM architectures to automatically identify and classify arrhythmias while minimizing preprocessing requirements. Leveraging insights from the MIT-BIH arrhythmia database, our study adopts a rigorous five-fold cross-validation approach to comprehensively evaluate model performance.

Moreover, we improve traditional methods by combining CNN and LSTM networks, which work together to understand both the space and time aspects of ECG data. Compared to older approaches, this combination results in better performance metrics like accuracy, sensitivity, specificity, and F1 score. Our results emphasize the importance of using smart preprocessing techniques and incorporating specialized knowledge to make diagnostics better.

Keywords: Cardiac Arrhythmia Detection, Deep Learning Models, Convolutional Neural Networks, Long Short-Term Memory Algorithms, Electrocardiogram Signals, Accuracy, Efficiency, CNN-LSTM Architectures.

INTRODUCTION:

Deep learning, a subset of artificial intelligence (AI), involves training neural networks with multiple layers to recognize patterns in data. When applied to ECGs, deep learning models can learn to identify subtle patterns that may indicate various heart conditions. Electrocardiogram (ECG) is one of the most important and, as a result,

observed electrophysiological data,1where the variability of its cyclic activity may provide evidence of cognitive, behavioural, and cardiovascular changes and pathologies. This bio-signal is the gold standard for offering non-invasive diagnostics for cardiac disorders by analysing its characteristics. Thanks to the ambulatory properties of these instruments, which provide ECG tracking over long periods, wearables now allow for the diagnosis of unusual cardiac episodes. Early diagnosis of these episodes in our everyday lives improves people's chances of living a healthier lifestyle. Automatic identification using signal processing and pattern recognition techniques may be useful in delivering relevant data without relying on hospital services. These models can be trained on large datasets of ECG recordings to achieve high accuracy in tasks such as arrhythmia detection, gender identification from ECG signals, and even predicting mortality risks associated with certain ECG patterns. Recent studies have utilized vast datasets, sometimes comprising millions of ECGs, to develop models that provide cardiologist-level accuracy in interpreting ECGs. These models can also offer interpretability, which is crucial for clinicians to trust and understand the AI's decision-making process. For instance, a study published in February 2024 utilized over 2.3 million ECGs to create a deep-learning model that could interpret ECGs with remarkable granularity. This model was not only able to diagnose cardiac abnormalities but also stratify mortality risks and identify new clinically useful information from ECG data alone. Deep Learning (DL) has significantly advanced the field of ECG arrhythmia detection and classification. Researchers have been focusing on various DL architectures to improve the accuracy and efficiency of detecting cardiac irregularities. The progress from 2017 to 2023 has been substantial, with several key developments. A variety of DL models have been explored, including Convolutional Neural Networks (CNNs), Multilayer Perceptron's (MLPs), Deep learning-based analysis methods using CNN, CNN-LSTM.

LITERATURE SURVEY:

Automated detection of cardiac arrhythmia using deep learning techniques.

- In an open-access study published under the CC BY-NC-ND license, Swapna G*, Soman KP, and Vinayakumar R describe how they plan to use deep learning methods to diagnose cardiac arrhythmia from ECG signals with as little preprocessing as possible. For the purpose of classifying arrhythmias, they create and analyze various deep learning models, such as CNN, RNN, LSTM, GRU, and hybrid architectures, and assess their effectiveness using metrics like accuracy and F1 score. The research makes use of five-fold cross-validation and the MIT-BIH arrhythmia database. The study illustrates the effectiveness of several deep learning algorithms and training strategies in automating arrhythmia detection and classification, while also noting certain limits including the need for additional real-world datasets and computing expenses.

2) ECG Guided Automated Diagnostic Intervention of Cardiac Arrhythmias with Extra-Cardiac Noise Detection using Neural Network.

With this study, Binoy Sasmal and Sayan Roy—both connected to Chennai, India's SRM Institute of Science and Technology—advance the subject of cardiac health monitoring. Using deep neural networks to analyze regular electrocardiogram (ECG) rhythms and spot anomalies like noise and arrhythmias is the main goal of their study. They seek to increase the precision and effectiveness of ECG analysis while providing a cost-effective and non-invasive solution by putting forth two different topologies for noise and arrhythmia identification. They use convolutional neural networks (CNN) for noise detection and recurrent neural networks (RNN) with gated recurrent units (GRU) for arrhythmia identification to classify ECG data into groups using autoencoder models.

MIT-BIH NOISE STRESS DATABASE:

It was critical to collect clean signals with the same characteristics as those mentioned above for noise detection, as well as signals that clearly indicated particular forms of noise. As a result, signals from the Fantasia Database were used for both clean and noise-affected sections of regular ECG signals, which were created by adding different types of noise to said signals. The MIT-BIH Noise Stress Array, which involves wandering baseline, muscle artefact, and raw noise electrode motion, was used to generate noise-containing signals. By adding these noise-corrupted signals to the Fantasia Database's clean signals and retaining the same sampling frequency, this approach was used to mark in a controlled manner while still collecting a larger range of noise-corrupted signals. In the training process, the same volume of data of clean ECG signals with NSR was used, as well as equivalent quantities of each form of noise. As a result, each class's percentage in the total training results was equal.

MIT-BIH ARRHYTHMIA DATABASE

The proposed system integrates convolutional neural networks (CNN) and long short-term memory (LSTM) architectures for enhanced cardiac arrhythmia detection from electrocardiogram (ECG) signals. Following minimal preprocessing, ECG signals are fed into a hybrid CNN-LSTM model, which combines spatial feature extraction and temporal sequence modeling. This model effectively captures both spatial and temporal dependencies within the ECG data, enabling accurate detection of arrhythmic patterns. The output undergoes classification to identify one of seven predefined arrhythmia classes. Through rigorous evaluation, our system demonstrates superior performance metrics, paving the way for improved non-invasive and cost-effective arrhythmia detection in clinical and wearable device applications

ADVANTAGES OF PROPOSED SYSTEM:

- 1) High accuracy
- 2)High efficiency

ALGORITHMS

CNN(Convolutional Neural Network)

CNN stands for Convolutional Neural Network. It's a type of deep learning algorithm that's particularly effective for image recognition, classification, and segmentation tasks.

Convolutional Layer: This is the core building block of a CNN. It applies a set of learnable filters (also known as kernels) to small patches of the input image, typically moving left to right and top to bottom. Each filter detects specific features, such as edges, corners, or textures. The result of this operation is a set of feature maps, which capture the presence of these features at different spatial locations in the input image.

Activation Function: After the convolution operation, an activation function like ReLU (Rectified Linear Unit) is applied element-wise to introduce non-linearity into the network. This helps the network learn more complex patterns and relationships in the data.

Pooling Layer: Pooling layers are used to reduce the spatial dimensions of the feature maps while retaining important information. Max pooling is a common technique, where the maximum value within each region of the feature map is retained, effectively downsampling the feature maps.

LSTM(Long Short Term Memory)

LSTM is a type of recurrent neural network (RNN) architecture, designed to overcome the limitations of traditional RNNs in capturing long-term dependencies in sequential data. Traditional RNNs suffer from the vanishing gradient problem, which makes it difficult for them to learn and remember long-term dependencies in sequences.

LSTM addresses this issue by introducing a memory cell and several gating mechanisms that control the flow of information into and out of the cell. Here's a breakdown of the key components of an LSTM unit:

1. **Cell State (C_t)**: This is the "memory" of the LSTM. It runs straight down the entire chain, with only minor linear interactions. It can add or remove information to it, regulated by structures called gates.

2. **Hidden State (h_t)**: This is the output of the LSTM cell at a specific time step. It's a filtered version of the cell state and is used to carry information to the next time step and/or to the output.

3. **Forget Gate (f_t)**: This gate decides what information to discard from the cell state. It takes as input the previous hidden state h_{t-1} and the current input x_t and produces a number between 0 and 1 for each number in the cell state C_{t-1}. A 1 represents "keep this" while a 0 represents "get rid of this".

MACHINE LEARNING

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results. Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input, use an algorithm to formulate answers. A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation. Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

DEEP LEARNING

Deep learning is a subset of machine learning that employs artificial neural networks with multiple layers to model and extract high-level features from complex data representations. These networks, often referred to as deep neural networks, consist of an input layer, one or more hidden layers, and an output layer. Each layer contains numerous interconnected neurons that perform mathematical operations on the input data to progressively learn and extract abstract representations. Deep learning algorithms, such as convolutional neural networks (CNNs) for image processing and recurrent neural networks (RNNs) for sequential data, have demonstrated remarkable success across a wide range of domains, including computer vision, natural language processing, speech recognition, and reinforcement learning.

APPLICATION OF DEEP LEARNING

Deep learning has found applications across various domains due to its ability to learn from large volumes of data and automatically extract complex patterns and features. Here are some notable applications:

Computer Vision: Deep learning has revolutionized computer vision tasks such as object detection, image classification, segmentation, and image generation. Applications range from autonomous vehicles and facial recognition systems to medical image analysis and satellite image interpretation.

Natural Language Processing (NLP): Deep learning techniques like recurrent neural networks (RNNs), transformers, and deep belief networks have significantly advanced NLP tasks such as machine translation, sentiment analysis, text summarization, question answering, and language generation.

SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.
The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

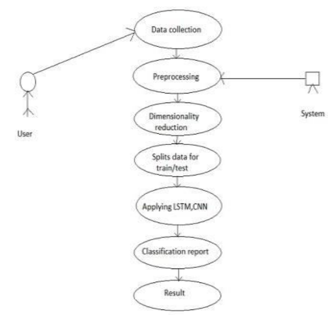

Fig-1: Use Case Diagram

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

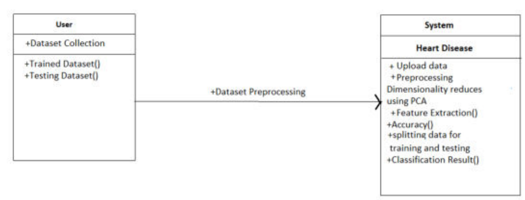

Fig-2: Class Diagram

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

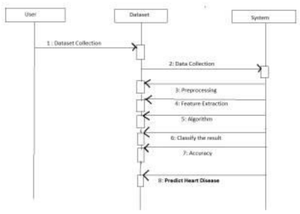

Fig-3: Sequence Diagram

SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive.

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

UNIT TESTING

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

SAMPLE SCREENS:

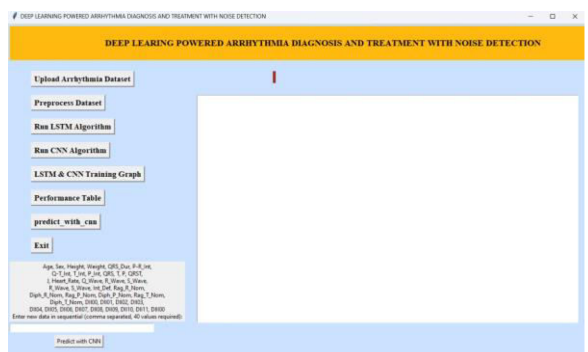

Fig 4: To run project double, click on 'run.bat' file to get below screen

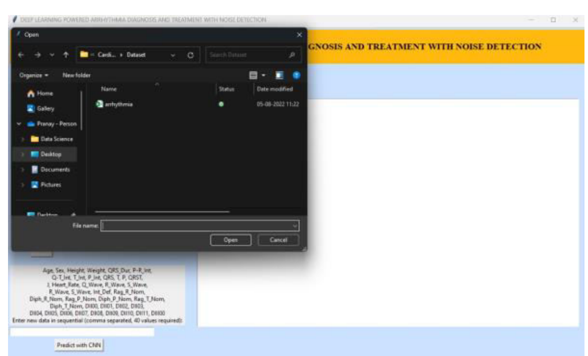

Fig5: In above screen click on 'Upload Arrhythmia Dataset' button to upload dataset and get below output .

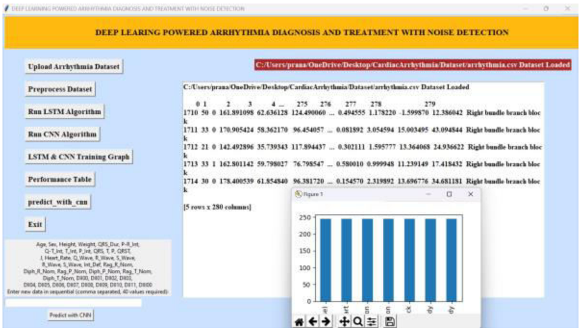

Fig6: In above screen selecting and uploading 'Arrhythmia' dataset and then click on 'Open' button to load dataset and get below output

Fig7: In above screen we can see dataset loaded and in graph x-axis represents 7 different disease stages and y-axis represents number of records found

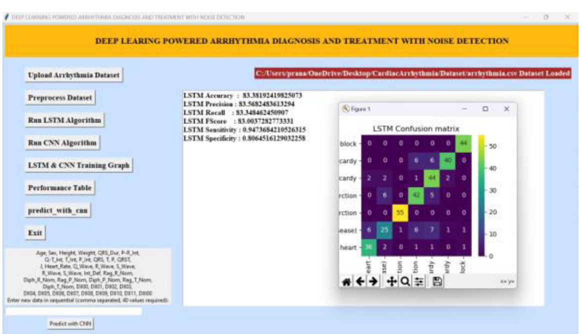

Fig8: In above screen with LSTM, we got 83% accuracy and in confusion matrix graph x-axis represents Predicted classes and y-axis represents TRUE

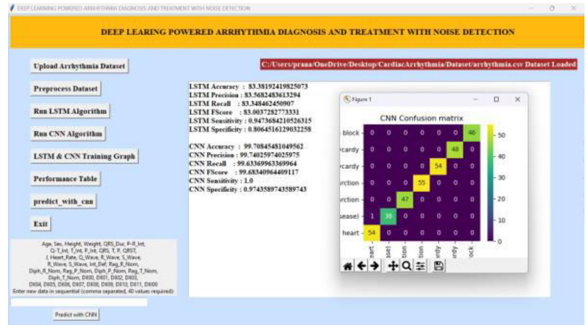

Fig9: In above screen with CNN, we got 99% accuracy and in confusion matrix graph x-axis represents Predicted classes and y-axis represents TRUE

Fig10: In the above graph x-axis represents the training epoch and the y-axis represents training accuracy and loss values and green colour line represents the LSTM accuracy

Dataset Name	Algorithm Name	Accuracy	Precision	Recall	FSCORE	Sensitivity	Specificity
MIT-BH Dataset	LSTM	83.38192419825073	83.5682483613294	83.348462450907	83.0037282773331	0.9473684210526315	0.8064516129032258
MIT-BH Dataset	CNN	99.70845481049562	99.74025974025975	99.63369963369964	99.68340964409117	1.0	0.9743589743589743

Fig11: In the above screen we can see the output metrics of both algorithms in tabular format.

Fig12: In the above screen enter the new data(40 values required) then click on the predict_with_cnn button to get the predicted label.

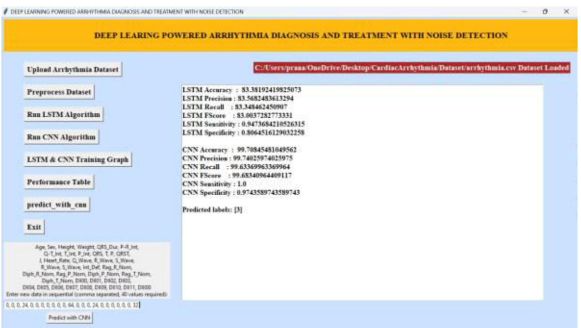

Fig13: In the above screen we can see the predicted label for new data.

CONCLUSION

In conclusion, the application of deep learning techniques for the automatic detection of cardiac arrhythmia holds significant promise in revolutionizing diagnostic processes and improving patient outcomes. Through the utilization of large datasets and advanced neural network architectures, deep learning models have demonstrated remarkable accuracy in identifying various types of arrhythmias from electrocardiogram (ECG) signals. These models offer advantages such as rapid analysis, scalability, and the potential for real-time monitoring, thereby facilitating timely interventions and personalized treatment plans. While challenges such as data quality, interpretability, and generalization to diverse populations remain, ongoing research and technological advancements continue to address these limitations. Overall, the integration of deep learning into clinical practice has the potential to enhance the efficiency and effectiveness of arrhythmia detection, ultimately contributing to better management of cardiac health and improved patient care.

REFERENCES

1. Rajpurkar, P., Hannun, A. Y., Haghpanahi, M., Bourn, C., Ng, A. Y. (2017). Cardiologist-level arrhythmia detection with convolutional neural networks. arXiv preprint arXiv:1707.01836.
2. Attia, Z. I., Kapa, S., Lopez-Jimenez, F., McKie, P. M., Ladewig, D. J., Satam, G., ... & Asirvatham, S. J. (2019). Screening for cardiac contractile dysfunction using an artificial intelligence–enabled electrocardiogram. Nature Medicine, 25(1), 70-74.
3. Minchole, A., Camps, J., Lyon, A., & Rodriguez, B. (2019). Machine learning in the electrocardiogram. Journal of Electrocardiology, 57, S61-S64.
4. Mousavi, S. M., Pourahmadi, M. R., & Sarrafzadegan, N. (2020). Application of machine learning techniques in detection of heart diseases: A systematic review. Journal of Research in Medical Sciences: The Official Journal of Isfahan University of Medical Sciences, 25.
5. Hong, S., Oh, S., Woo, S., Moon, S., & Kang, J. (2021). Deep learning in the diagnosis and prognosis of cardiac arrhythmias: A systematic review. Frontiers in Cardiovascular Medicine, 8, 639241.