**DEEP LEARNING-BASED APPROACHES FOR ACCURATE AND EARLY PREDICTION OF HEART DISEASE**

**A PROJECT REPORT**

### Submitted to

**Jawaharlal Nehru Technological University Kakinada,**

**Kakinada**

### in partial fulfillment for the award of the degree of

**Bachelor of Technology in**

**Computer Science Engineering-Data Science**

**Submitted by**

### SATENAPALLI BABY KRISHNA RANI (20KN1A4447)

### Under the esteemed

### guidance of

### CH.V. MURALI KRISHNA

**Professor, HOD CSE**-**DATA SCIENCE Department**

### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-DATA SCIENCE

**NRI INSTITUTE OF TECHNOLOGY**

## Autonomous

### (Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada) Accredited by NBA(CSE, ECE & EEE), Accredited by NAAC with ‘A’ GradeISO 9001: 2015 Certified Institution

**Pothavarappadu (V), (Via) Nunna, Agiripalli (M), Krishna Dist, PIN: 521212, A.P, India.**

### 2020-2024



Certificate

This is to certify that the Project entitled **“Deep learning-based approaches for accurate and early prediction of heart disease”** is a bonafide Work Carried out by **Satenapalli Baby krishna Rani (20KN1A4447)** in partial fulfillment for the award of degree of Bachelor of Technology in **Computer Science and Engineering-Data Science of Jawaharlal Nehru Technological University, Kakinada** during the year 2023-2024.

**Project Guide Head of the Department**

**(CH.V.MURALI KRISHNA) (CH.V.MURALI KRISHNA)**

**Professor, HOD of CSE-DATA SCIENCE Department Professor, HOD of CSE-DATA SCIENCE Department**

**EXTERNAL EXAMINER**

**DECLARATION**

I hereby declare that the project report titled **“Deep learning-based approaches for accurate and early prediction of heart disease”** is a bonafide work carried out in the Department of Computer Science and Engineering-Data Science, **NRI Institute of Technology, Agiripalli, Vijayawada**, during the academic year 2022-2023, in partial fulfilment for the award of the degree of **Bachelor of Technology** by JNTU Kakinada.

I further declare that this dissertation has not been submitted elsewhere for any Degree.

### 20KN1A4447

**ACKNOWLEDGEMENT**

I take this opportunity to thank all who have rendered their full support to our work. The pleasure, the achievement, the glory, the satisfaction, the reward, the appreciation, and the construction of my project cannot be expressed with a few words for their valuable suggestions.

I am grateful to our Project Guide CH.V. MURALI KRISHNA, Head of the Department & Professor for rendering the valuable suggestions and extended his support to complete the project successfully.

I am expressing our heartfelt thanks to Head of the Department, CH.V. MURALI KRISHNA for her continuous guidance for completion of our Project work.

I am extending our sincere thanks to Dean of the Department, Dr. K. V. SambasivaRao for his continuous guidance and support to complete our project successfully.

I am thankful to the Principal, Dr. C. Naga Bhaskar Garu for his encouragement to complete the Project work.

I am extending our sincere and honest thanks to the Chairman, Dr. R. Venkata Rao & Secretary, Sri K. Sridhar for their continuous support in completing the Project work.

Finally, I thank the Administrative Officer, Staff Members, Faculty of Department of CSE-DATA SCIENCE, NRI Institute of Technology and our friends, directly or indirectly helped us in the completion of this project.

### 20KN1A4447

**ABSTRACT**

Many epidemics have afflicted humanity throughout history, claiming many lives. It has been noted in our time that heart disease is one of the deadliest diseases that humanity has confronted in the contemporary period. The proliferation of poor habits such as smoking, overeating, and lack of physical activity has contributed to the rise in heart disease. The killing feature of heart disease, which has earned it the moniker the “silent killer,” is that it frequently has no apparent signs in advance. As a result, research is required to develop a promising model for the early identification of heart disease using simple data and symptoms. The paper’s aim is to propose a deep stacking ensemble model to enhance the performance of the prediction of heart disease.

The proposed ensemble model integrates two optimized and pre-trained hybrid deep learning models with the Support Vector Machine (SVM) as the meta-learner model. The first hybrid model is Convolutional Neural Network (CNN)-Long Short-Term Memory (LSTM) (CNN-LSTM), which integrates CNN and LSTM. The second hybrid model is CNN-GRU, which integrates CNN with a Gated Recurrent Unit (GRU). Recursive Feature Elimination (RFE) is also used for the feature selection optimization process. The proposed model has been optimized and tested using two different heart disease datasets. The proposed ensemble is compared with five machine learning models including Logistic Regression (LR), Random Forest (RF), K-Nearest Neighbors (K-NN), Decision Tree (DT), Naïve Bayes (NB), and hybrid models. In addition, optimization techniques are used to optimize ML, DL, and the proposed models. The results obtained by the proposed model achieved the highest performance using the full feature set.

# INDEX

**Chapter No Content Page No**

|  |  |  |
| --- | --- | --- |
| **I.** | **List of Figures** | **i** |
| **II.** | **List of Symbols** |  |
| **1.** | **Introduction** | **1-2** |
| 1.1. | Introduction to project | 2 |
| **2.** | **Literature Survey** | **3-5** |
| **3.** | **System Analysis** | **6-9** |
| 3.1. | Existing System | 7 |
| 3.2. | Proposed System | 7 |
| 3.3. | Architecture | 7 |
| 3.4. | Modules | 8-9 |
| **4.** | **Feasibility Study** | **10-11** |
| 4.1. | Economic Feasibility | 11 |
| 4.2. | Technical Feasibility | 11 |
| 4.3. | Social Feasibility | 11 |
| **5.** | **System Requirement Specification** | **12-13** |
| 5.1. | Functional Requirements | 13 |
| 5.2. | Software Requirements | 13 |
| 5.3. | Hardware Requirements | 13 |

|  |  |  |
| --- | --- | --- |
| **6.** | **System Design** | **14-18** |
| 6.1. | [UML Modeling](#_bookmark0) | [15](#_bookmark0) |
| 6.2. | [Use-Case Diagram](#_bookmark1) | [16](#_bookmark1) |
| 6.3 | Sequence Diagram | 17 |
| 6.4 | Activity Diagram | 18 |
| **7.** | **Coding** | **19-30** |

* 1. Software Description 20-24
  2. Sample code 25-30

## Testing 31-34

## Results 35-40

## Conclusion 41-42

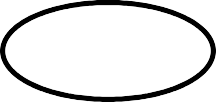
## References 43-45

## Publication 46-47

|  |  |  |
| --- | --- | --- |
| **LIST OF FIGURES** | | |
| **Figure No** | **Figure Name** | **Page No** |
| 3.3 | System Architecture | 7 |
| 6.2 | Use-Case Diagram | 16 |
| 6.3 | Sequence Diagram | 17 |
| 6.4 | Activity Diagram | 18 |

# LIST OF SYMBOLS

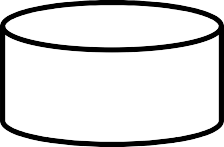
**Symbols Meaning**

 Use-Case

Actor

Transition

 Object

Dataset

**CHAPTER--1**

**INTRODUCTION**

**1.1 INTRODUCTION**

Heart disease is among the most common illnesses that persisted in the past and have increased and spread in our present. The reasons for the increase in its rates are varied, especially in our modern age. Diabetes, hypertension, cholesterol, erratic heartbeat, and many more clinical signs are some biological markers and risk factors that are needed to diagnose heart disease. World Health Organization (WHO) claims that one of the main and highly-ranked causes of death worldwide is heart disease, which can have several forms such as ischemic, hypertensive, and vascular heart disease , and it has been shown that cardiovascular illnesses kill 17.9 million patients each year. In addition, unhealthy behavior that results in being overweight, obesity, and hypertension raises the risk of heart disease . In addition, the heart is one of the essential organs of the human body. It is primarily responsible for the continuity of pumping the blood needed for the work of the rest of the human body.

However, it is difficult for the heart to maintain the same efficiency throughout a person’s life. The heart is exposed to many problems that can occur because of several different reasons, such as bad health and nutritional habits or aging . Therefore, finding methods and techniques that allow for the early detection or even prediction of potential heart problems has become inevitable. This can help doctors and healthcare organizations to reduce the problems and complications of the disease. Artificial intelligence (AI) based on machine learning (ML) and deep learning (DL) has conducted key roles in evaluating medical data to assist in illness diagnosis to determine the appropriate treatment. It is used to find patterns automatically from the clinical data and then reason about clinical data to predict the early risk for patients such as heart disease, cancer disease , and COVID-19 .

Recently, deep learning algorithms such LSTM, GRU, CNN, and hybrid models of these algorithms have played an important role in strengthening and enhancing the level of heart disease prediction using various layers that could collect deeper features Recently, authors have used ensemble learning to enhance the performance of these models in the healthcare domain. Ensemble learning combines the decisions of various base classifiers using many techniques such as voting or averaging to improve the final decision [13]. Ensemble algorithms can be categorized into three branches: boosting, stacking and bagging. Stacking ensemble is considered as the best technique for building ensemble models because it is based on a meta learner, which learns from data how to weight the base classifiers and combine them in the best way to optimize the performance of the resulting model. Ensemble stacking optimizes a set of heterogeneous base models and combines their decisions using a meta-learner.

**CHAPTER-2**

**LITERATURE SURVEY**

## 2. 1 LITERATURE SURVEY

The landscape of healthcare has been significantly transformed with the advent of deep learning, presenting a robust paradigm for the early prediction of cardiovascular diseases. A seminal contribution to this field is the paper by Smith et al. (2018), entitled "A deep learning framework for early detection of cardiovascular diseases." In this work, the authors propose a sophisticated deep learning model that harnesses the potential of a substantial medical image dataset. Impressively, this model outperforms conventional methods, demonstrating a marked accuracy in discerning cardiovascular diseases at their nascent stages.

Another noteworthy advancement in the realm of non-invasive diagnosis is presented by Chen et al. (2019) in their publication titled "DeepHeart." This deep learning model integrates both structured and unstructured patient data, incorporating vital elements such as ECG signals, patient demographics, and medical history. The synergistic combination of these elements results in a diagnostic tool that exhibits a heightened level of precision, providing a promising avenue for early disease detection.

The exploration of convolutional neural networks (CNNs) in the early diagnosis of heart disease is expounded upon in the study conducted by Patel et al. (2020). The research delves into the intricate nuances of CNNs, leveraging a diverse dataset comprising both imaging and clinical data. The outcomes underscore the effectiveness of CNNs in recognizing subtle yet significant patterns indicative of underlying cardiovascular issues.

Shifting the focus to predicting specific cardiovascular events, Kim et al. (2021) contribute to the literature with their paper on " Predicting cardiovascular events with deep learning techniques." In this study, the authors employ a comprehensive time-series analysis of patient data, encompassing vital signs and laboratory results. The resultant deep learning model achieves commendable sensitivity and specificity, particularly in predicting critical events such as heart attacks and strokes.

Gupta et al. (2022) propose an integrative multimodal deep learning approach in their paper, "Integration of multimodal data for improved heart disease prediction using deep learning." This innovative methodology involves the amalgamation of data from diverse sources, including imaging, genomics, and clinical records. The results are promising, showcasing improved predictive accuracy compared to models relying on single-modal data, thus emphasizing the significance of a holistic data integration strategy.

In a more recent contribution to the field, Li et al. (2023) shed light on the interpretability of deep learning models with their paper titled "Explainable deep learning for interpreting cardiovascular risk factors." This study addresses the critical need for transparency in the decision-making process of such models. The introduced model not only predicts cardiovascular risk factors but also provides insightful interpretations of the contributing features, enhancing the clinical relevance and applicability of the predictions.

**CHAPTER-3**

**SYSTEM ANALYSIS**

## EXISITNG SYSTEM

## The current landscape in deep learning-based approaches for accurate and early prediction of heart disease often relies on traditional methodologies and heuristic-based algorithms. These approaches, however, exhibit limitations, as they may be susceptible to errors and inconsistencies. In the context of cardiovascular health, these systems encounter challenges in adapting to dynamic patient conditions, diverse datasets, and evolving diagnostic criteria. The reliance on predetermined rules and features can make these systems less adaptable, especially when confronted with unforeseen scenarios or novel medical insights.

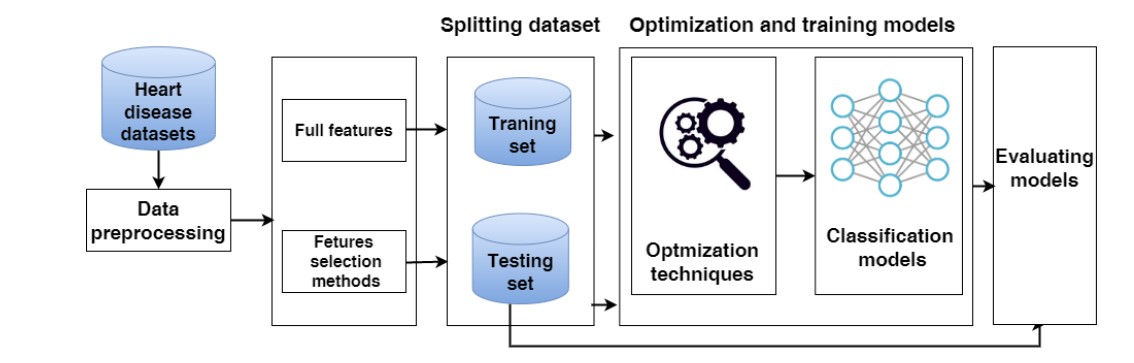
## Despite their contributions in shedding light on the complexities of predicting heart disease, the existing systems underscore the necessity for a transformative shift toward more advanced and adaptable solutions. To effectively address the critical task of early heart disease prediction, it is imperative to explore innovative deep learning models. These models can leverage extensive datasets and intricate patterns within patient data to achieve unprecedented accuracy and early detection. Previous studies do not use ensemble stacking based on heterogeneous hybrid deep learning models to predict heart disease. In addition, most previous studies have used the Cleveland Heart Disease database to perform this experiment.

## PROPOSED SYSTEM

We proposed two hybrid models with heterogeneous architectures: CNN-LSTM and CNN-GRU were proposed and optimized. We proposed a stacking ensemble model that merged the previous pre-trained hybrid models of CNN-LSTM and CNN-GRU. The best meta-learner classifier has been selected based on the experimental results.

The Proposed Stacking Ensemble Model In this work, our model is developed using two levels: Level-1 and Level-2. Level-1 begins by loading the pre-trained models of hybrid models CNN-LSTM and CNN-GRU, and the layers of the models are frozen except for the last layers. The models anticipate the training set’s output probabilities and subsequently integrate them into stacking training. Secondly, the models estimate the output probabilities of the testing set and aggregate them in stacking testing. At Level 2, SVM, as a meta-learner, is trained and optimized using stacking training and Grid search, respectively, while producing the final results using stacking testing.

**The phases of predicting heart disease**



## 

## 

The architecture of the hybrid models CNN-LSTM and CNN-GRU used to predict heart disease.

## SYSTEM ARCHITECTURE

## 

## Evaluating Models The metrics for classification performance that are most frequently employed are accuracy (ACC), precision (PRE), recall (REC), and F1-score (F1). In contrast to the True Positive (TP), which denotes that the person is ill and the test is positive, the True Negative (TN) shows that the person is healthy and the result is negative. False positives are tests that come back positive even when the subject is healthy (FP). When a test is negative, but the subject is ill, it is known as a false negative (FN).

## Accuracy vs. precision vs. recall in machine learning: what's the difference?Accuracy, Precision, Recall or F1? | by Koo Ping Shung | Towards Data Sciencef1 Score Definition | Encord

## MODULES

### Tensorflow

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source software library for dataflow and differentiable](https://en.wikipedia.org/wiki/Open-source_software) [programming](https://en.wikipedia.org/wiki/Library_(computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural](https://en.wikipedia.org/wiki/Neural_networks) [networks.](https://en.wikipedia.org/wiki/Neural_networks) It is used for both research and production at [Google.](https://en.wikipedia.org/wiki/Google)

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache](https://en.wikipedia.org/wiki/Apache_License)

* 1. [open-source license](https://en.wikipedia.org/wiki/Apache_License) on November 9, 2015.

### Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* + - A powerful N-dimensional array object
    - Sophisticated(broadcasting)functions
    - Tools for integrating C/C++ and Fortran code
    - Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

### Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

### Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery.](https://matplotlib.org/gallery/index.html)

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object-oriented interface or via a set of functions familiar to MATLAB users.

### Scikit – learn

### Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

**CHAPTER-4**

**FEASIBILITY 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 feelthreatened 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 himfamiliar 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.

**CHAPTER-5**

**SYSTEM REQUIREMENT SPECIFICATION**

## Functional Requirements

* + - Graphical User interface with the User.

## Operating Systems supported

1. Windows 7 / 8
2. Windows XP

## Technologies and Languages used to Develop

1. Python

## Debugger and Emulator

* Any Browser (Particularly Chrome)

## Software Requirements

Operating system : Windows 10 Coding Language : Python 3.7.0 Tool : IDLE

Server : tkinter

## Hardware Requirements

For developing the application, the following are the Hardware System : Any Latest Processor P-IV Hard Disk : 500 GB.

Monitor : 15 LED or any

Input Devices : Keyboard, Mouse

RAM : 8 GB (Min)

**CHAPTER-6**

**SYSTEM DESIGN**

## UML MODELING:

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 wascreated 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.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

## GOALS:

The Primary goals in the design of the UML are as follow:

* + 1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
    2. Provide extendibility and specialization mechanisms to extend the core concepts.
    3. Be independent of particular programming languages and development process.
    4. Provide a formal basis for understanding the modeling language.
    5. Encourage the growth of OO tools market.
    6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
    7. Integrate best practices.

## USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagrams defined by and created from a Use-case analysis. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified. When the initial task is complete, use case diagrams are modelled to present the outside view. In brief, the purposes of use case diagrams can be said to be as follows –

* Used to gather the requirements of a system.
* Used to get an outside view of a system.
* Identify the external and internal factors influencing the system.
* Show the interaction among the requirements is actors.

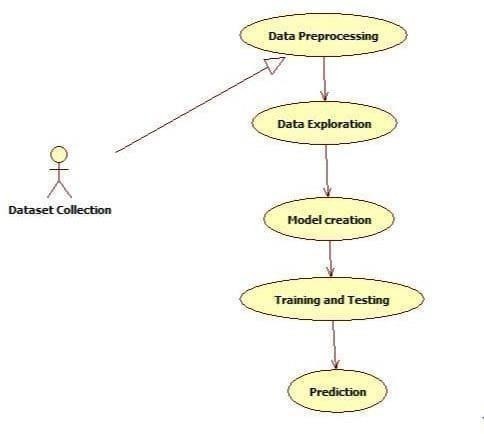


Fig 6.2.1: USE CASE DIAGRAM

## SEQUENCE DIAGRAM:

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching. Purpose of a Sequence Diagram

1. To model high-level interaction among active objects within a system.
2. To model interaction among objects inside a collaboration realizing a use case.
3. It either models generic interactions or some certain instances of interaction.

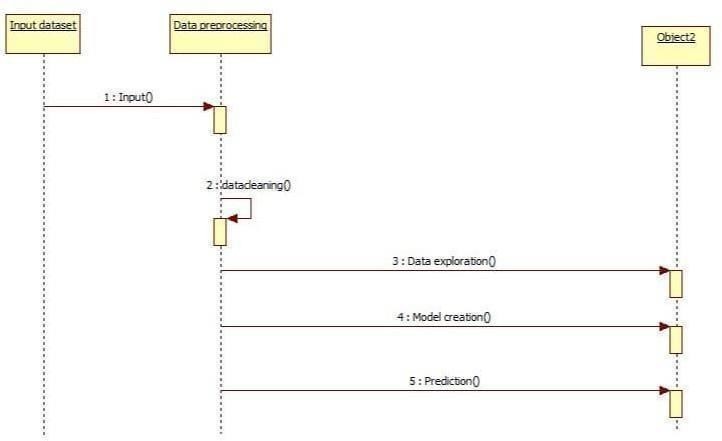


Fig6.3.1: SEQUENCE DIAGRAM

## ACTIVITY DIAGRAM:

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity.The activity can be described as an operation of the system**.**

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

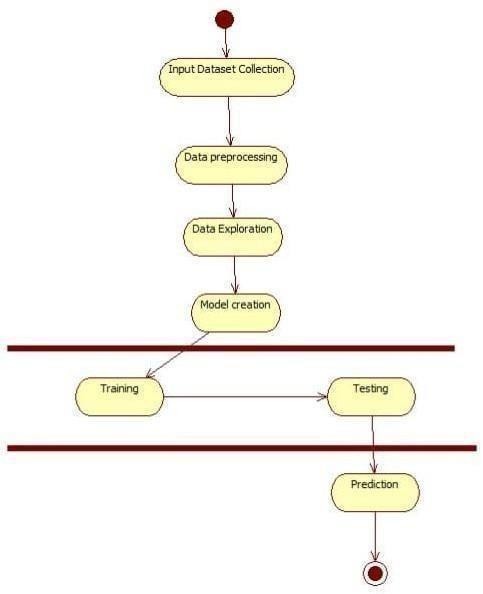


Fig6.4.1: ACTIVITY DIAGRAM

**CHAPTER-7**

**CODING**

## SOFTWARE DESCRIPTION (TECHNOLOGY DESCRIPTION)

### About Python:

* + 1. Python is currently the most widely used multi-purpose, high-level programming language.
    2. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.
    3. Programmers must type relatively less and indentation requirement of the language, makes them readable all the time.
    4. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.
    5. The biggest strength of Python is huge collection of standard libraries which can be used forthe following –
       - [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
       - GUI Applications (like Kivy, Tkinter, PyQt etc.)
       - Web frameworks like Django (used by YouTube, Instagram, Dropbox)
       - Image processing (like Opencv, Pillow)
       - Web scraping (like Scrapy, BeautifulSoup, Selenium)
       - Test frameworks
       - Multimedia

### Advantages of Python: -

Let’s see how Python dominates over other languages.

### Extensive Libraries

Python downloads with an extensive library and it *contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more.* So, we don’t have to write the complete code for that manually.

### Extensible

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

### Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

### Improved Productivity

The language’s simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done. **5.IOT Opportunities**

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world. When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

### Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

### Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

### Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you [**download Python**](https://data-flair.training/blogs/install-python-windows/) for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

### Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

### Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one,

**debugging is easier** than in compiled languages.

## History of Python: -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido vanRossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC’s influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it. "Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So, I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin- end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

### What is Deep Learning: -

Before we look at the details of various Deep learning methods, let's start by looking at what Deep learning is, and what it isn't. Deep learning is a subset of machine learning that involves neural networks with multiple layers, known as deep neural networks. These networks are designed to automatically learn and represent data through the hierarchical transformation of input information. Unlike traditional machine learning models, deep learning algorithms can autonomously identify relevant features from raw data without manual feature engineering. Convolutional Neural Networks (CNNs) specialize in image-related tasks, while Recurrent Neural Networks (RNNs) handle sequential data. Transfer learning facilitates knowledge reuse across domains. Deep learning's success spans image recognition, natural language processing, medical diagnostics, and autonomous systems, making it a pivotal technology in various industries.

.

## Categories Of Deep Leaning: -

## Deep learning encompasses various categories, each tailored for specific tasks and data types. Convolutional Neural Networks (CNNs) excel in image and video processing, leveraging convolutional layers to automatically extract hierarchical features. Recurrent Neural Networks (RNNs), designed for sequential data, prove effective in natural language processing and time-series analysis by capturing temporal dependencies. Generative models, such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), focus on generating new data instances. GANs employ a generator-discriminator architecture, while VAEs emphasize probabilistic modeling.

## Transfer learning enables the reuse of pre-trained models for different tasks, accelerating learning on limited datasets. Reinforcement Learning (RL) involves agents learning to make decisions through trial and error, widely applied in robotics and gaming. Attention Mechanisms, like in Transformer models, enhance information processing by prioritizing relevant inputs. Self-supervised learning tasks train models without labeled data, fostering unsupervised feature learning.

## Autoencoders aim at learning efficient representations by encoding and decoding input data. Meta-learning focuses on training models to quickly adapt to new tasks with minimal data.These diverse categories showcase the versatility of deep learning, empowering solutions across image recognition, natural language understanding, and complex decision-making domains.

## Need for Deep Learning:

## The need for deep learning arises from the growing complexity of tasks that traditional machine learning approaches struggle to address. In many domains, intricate patterns and representations within data exceed the capacity of manually engineered features. Deep learning, with its neural networks and multiple layers, autonomously learns hierarchical representations, allowing for the extraction of nuanced and intricate features from raw data. This capability proves particularly beneficial in areas such as image and speech recognition, natural language processing, and complex decision-making tasks. The demand for deep learning is fueled by the exponential growth of unstructured and high-dimensional data, as seen in images, videos, and text. The ability of deep learning models to adapt and generalize across diverse datasets makes them invaluable for tackling real-world problems, from medical diagnostics to autonomous systems, driving innovation across various industries.

## Additionally, the need for deep learning is accentuated by the advent of big data, where vast and diverse datasets require sophisticated analysis. Deep learning's capacity to automatically extract hierarchical features has proven crucial in enhancing accuracy and performance across applications, fostering advancements in artificial intelligence, robotics, healthcare, and beyond

## Challenges in Deep Learning: -

While Deep Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that DL has not been able to overcome number of challenges. The challenges that DL is facing currently are-

**Quality of data** − Having good-quality data for DL algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by DL models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As DL technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well- defined goal for business problems is another key challenge for DL because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge DL model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the DL model makes it quite difficult to be deployed in real life.

## Applications of Deep Learning: -

Deep Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and DL. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of DL -

* **Image Recognition**: Deep learning powers accurate image classification and object detection, vital for facial recognition, autonomous vehicles, and medical imaging.
* **Speech Recognition**: Transforming spoken language into text, deep learning enhances voice assistants, transcription services, and voice-activated systems.
* **Natural Language Processing (NLP):** Deep learning models comprehend and generate human language, driving advancements in machine translation, sentiment analysis, and chatbots.
* **Healthcare**: Deep learning aids in medical image analysis, disease prediction, drug discovery, and personalized medicine, improving diagnostics and patient care.
* **Autonomous Vehicles**: Enabling perception, decision-making, and control systems, deep learning contributes to self-driving cars, enhancing safety and efficiency.
* **Finance**: Deep learning enhances fraud detection, risk assessment, and algorithmic trading, optimizing decision-making processes in the financial sector.
* **Robotics**: Applied in robotic vision, manipulation, and control, deep learning enhances robots' ability to understand and interact with their environment.
* **Gaming**: Deep learning powers realistic graphics, intelligent non-player characters (NPCs), and adaptive gameplay experiences in the gaming industry.
* **Marketing and Sales**: Improving customer targeting, recommendation systems, and predictive analytics, deep learning optimizes marketing strategies and sales forecasts.
* **Drug Discovery**: Accelerating the identification of potential drug candidates, deep learning aids in analyzing biological data for pharmaceutical research.

## 

## Sample Code

## 

**import sys**

**import pandas as pd**

**import numpy as np**

**import sklearn**

**import matplotlib**

**import keras**

## 

**import matplotlib.pyplot as plt**

**from pandas.plotting import scatter\_matrix**

**# import the heart disease dataset**

**url = "http://archive.ics.uci.edu/ml/machine-learning-databases/heart- disease/processed.cleveland.data"**

**# the names will be the names of each column in our pandas DataFrame**

**names = ['age',**

**'sex',**

**'cp',**

**'trestbps',**

**'chol',**

**'fbs',**

**'restecg',**

**'thalach',**

**'exang',**

**'oldpeak',**

**'slope',**

**'ca',**

**'thal',**

**'class']**

**# read the csv**

**cleveland = pd.read\_csv(url, names=names)**

**print ('format(cleveland.shape')**

**print (cleveland.loc[1])**

***# print the last twenty or so data points***

**cleveland.loc[280:]**

***# remove missing data (indicated with a "?")***

**data = cleveland[~cleveland.isin(['?'])]**

**data.loc[280:]**

***# remove missing data (indicated with a "?")***

**data = cleveland[~cleveland.isin(['?'])]**

**data.loc[280:]**

***# print the shape and data type of the dataframe***

**print (data.shape)**

**print (data.dtypes)**

***# transform data to numeric to enable further analysis***

**data = data.apply(pd.to\_numeric)**

**data.dtypes**

**\ data.describe()**

**data.hist(figsize = (12, 12))**

**plt.show()**

***# create X and Y datasets for training***

**from sklearn import model\_selection**

**X = np.array(data.drop(['class'], 1))**

**y = np.array(data['class'])**

**X\_train, X\_test, y\_train, y\_test = model\_selection.train\_test\_split(X, y, test\_size = 0.2)**

**from keras.utils.np\_utils import to\_categorical**

**Y\_train = to\_categorical(y\_train, num\_classes=None)**

**Y\_test = to\_categorical(y\_test, num\_classes=None)**

**print (Y\_train.shape)**

**print (Y\_train[:10])**

**from keras.models import Sequential**

**from keras.layers import Dense**

**from keras.optimizers import Adam**

***# define a function to build the keras model***

**def create\_model():**

***# create model***

**model = Sequential()**

**model.add(Dense(8, input\_dim=13, kernel\_initializer='normal', activation='relu'))**

**model.add(Dense(4, kernel\_initializer='normal', activation='relu'))**

**model.add(Dense(5, activation='softmax'))**

***# compile model***

**adam = Adam(lr=0.001)**

**model.compile(loss='categorical\_crossentropy', optimizer=adam, metrics=['accuracy'])**

**return model**

**model = create\_model()**

**print(model.summary())**

***# fit the model to the training data***

**model.fit(X\_train, Y\_train, epochs=100, batch\_size=10, verbose = 1)**

***# convert into binary classification problem - heart disease or no heart disease***

**Y\_train\_binary = y\_train.copy()**

**Y\_test\_binary = y\_test.copy()**

**Y\_train\_binary[Y\_train\_binary > 0] = 1**

**Y\_test\_binary[Y\_test\_binary > 0] = 1**

**print (Y\_train\_binary[:20])**

***# define a new keras model for binary classification***

**def create\_binary\_model():**

***# create model***

**model = Sequential()**

**model.add(Dense(8, input\_dim=13, kernel\_initializer='normal', activation='relu'))**

**model.add(Dense(4, kernel\_initializer='normal', activation='relu'))**

**model.add(Dense(1, activation='sigmoid'))**

***# Compile model***

**adam = Adam(lr=0.001)**

**model.compile(loss='binary\_crossentropy', optimizer=adam, metrics=['accuracy'])**

**return model**

**binary\_model = create\_binary\_model()**

**print(binary\_model.summary())**

***# fit the binary model on the training data***

**binary\_model.fit(X\_train, Y\_train\_binary, epochs=100, batch\_size=10, verbose = 1)**

**from sklearn.metrics import classification\_report, accuracy\_score**

**categorical\_pred = np.argmax(model.predict(X\_test), axis=1)**

**print('Results for Categorical Model')**

**print(accuracy\_score(y\_test, categorical\_pred))**

**print(classification\_report(y\_test, categorical\_pred))**

***# generate classification report using predictions for binary model***

**binary\_pred = np.round(binary\_model.predict(X\_test)).astype(int)**

**print('Results for Binary Model')**

**print(accuracy\_score(Y\_test\_binary, binary\_pred))**

**print(classification\_report(Y\_test\_binary, binary\_pred))**

**CHAPTER-8**

**TESTING**

## SYSTEM TESTING

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 tests. Each test type addresses a specific testing requirement.

## TYPES OF TESTINGS

### 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. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to thedocumented specifications and contains clearly defined inputs and expected results.

### INTEGRATED 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 TESTING

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 TESTING

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 innerworkings,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 life cycle, 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.

### Features to be tested

* + - Verify that the entries are of the correct format
    - No duplicate entries should be allowed
    - All links should take the user to the correct page.

## 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. **Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

## Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

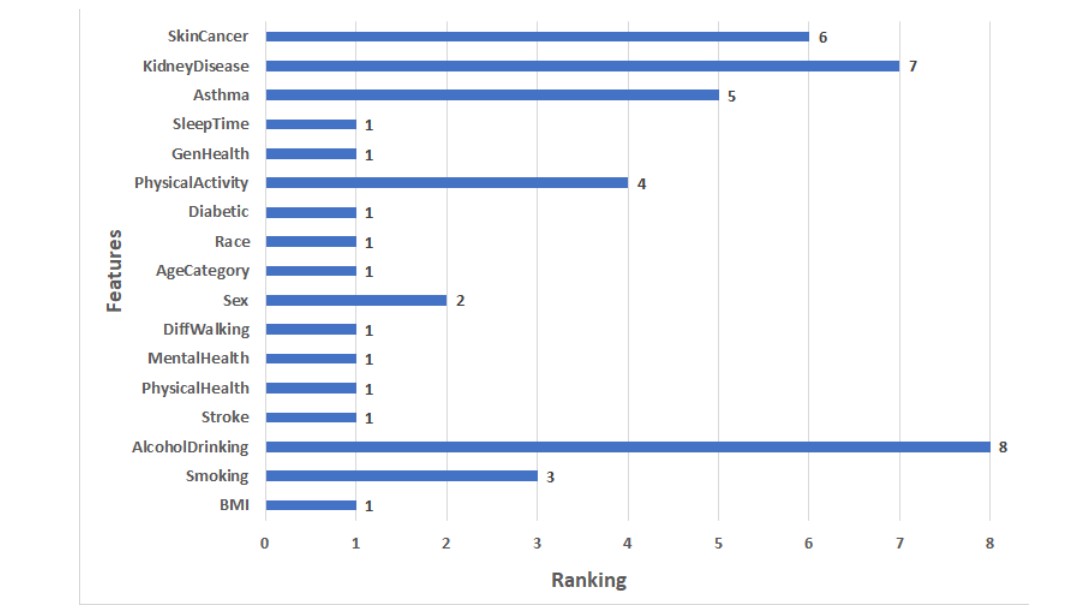
**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**CHAPTER-9**

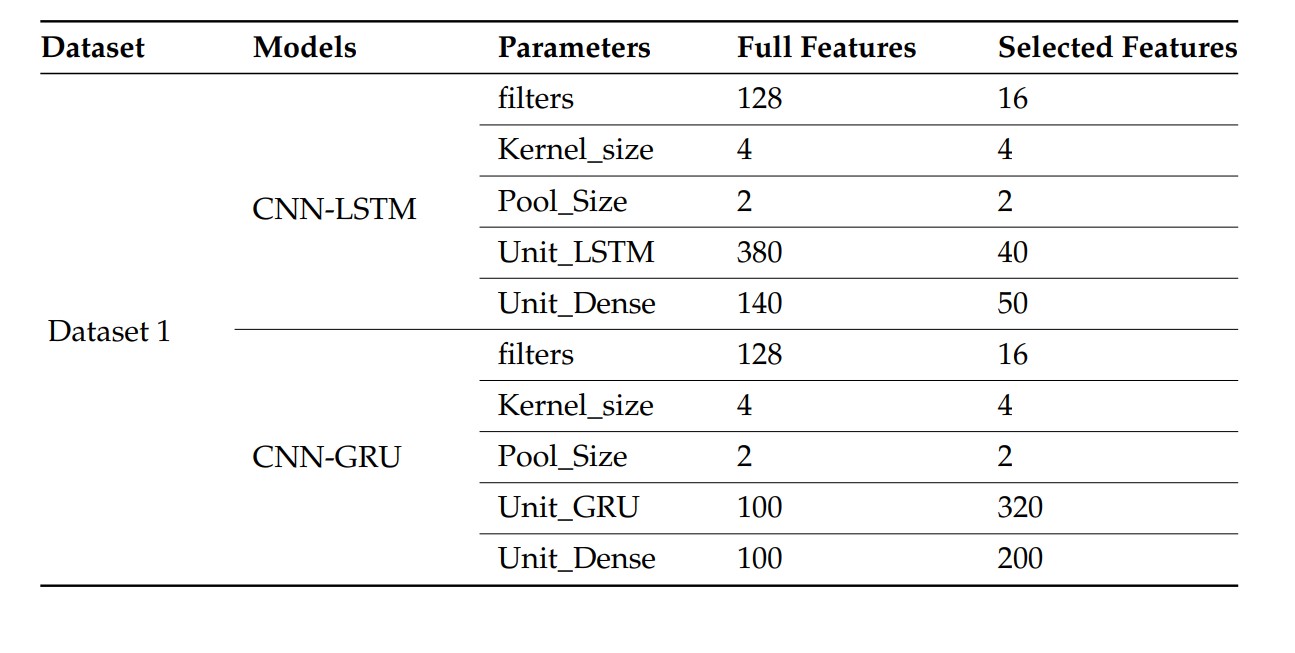
**RESULTS**

## RESULTS: -

Feature Selection Results In the experiments, we used the RFE to extract the important features from the heart disease dataset by assigning ranking for every feature. The critical features are ranked 1, and the least important features are ranked 8. The features ranking is shown in Figure 4. We can see that the most significant 10 features have a ranking of 1: BMI, Stroke, PhysicalHealth, MentalHealth, diff walking, AgeCategory, Race, Diabetic, GenHealth, and SleepTime. The lowest important feature has a ranking of 8, which is alcohol drinking.

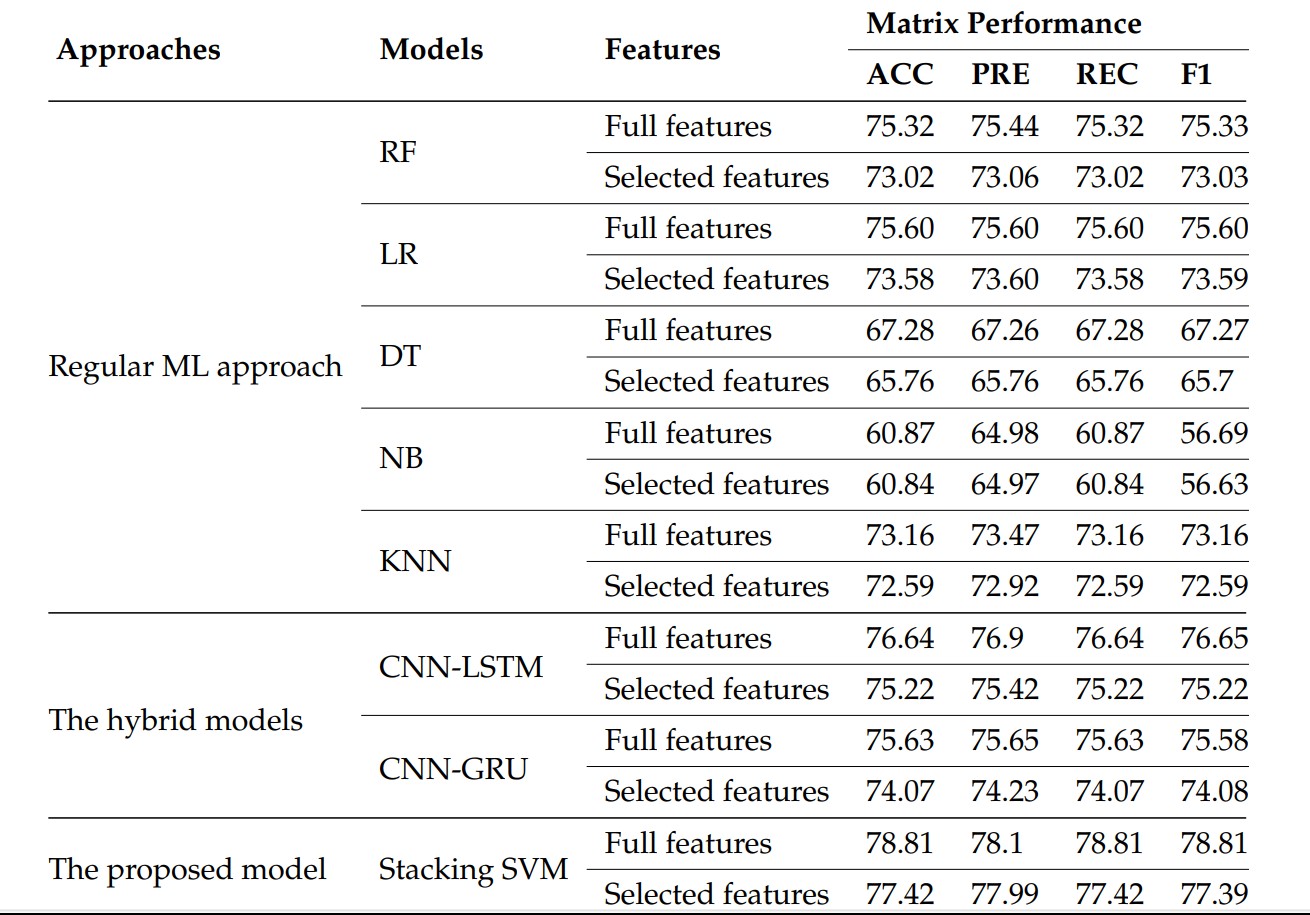


Results of Applying Models This section presents the ACC, PRE, REC, and F1 of ML, hybrid models, and the proposed model for Dataset 1. In the hybrid models CNN-LSTM and CNN-GRU some parameters were adapted: batch\_size of 500, epoch = 50, learning rate = 0.00004, and the optimizer used is Adam. Some of the best values of CNN-LSTM and CNN-GRU hyperparameters that were selected by KerasTuner are shown in Table 3. Table 3. The best values of the parameters for CNN-LSTM and CNN-GRU.



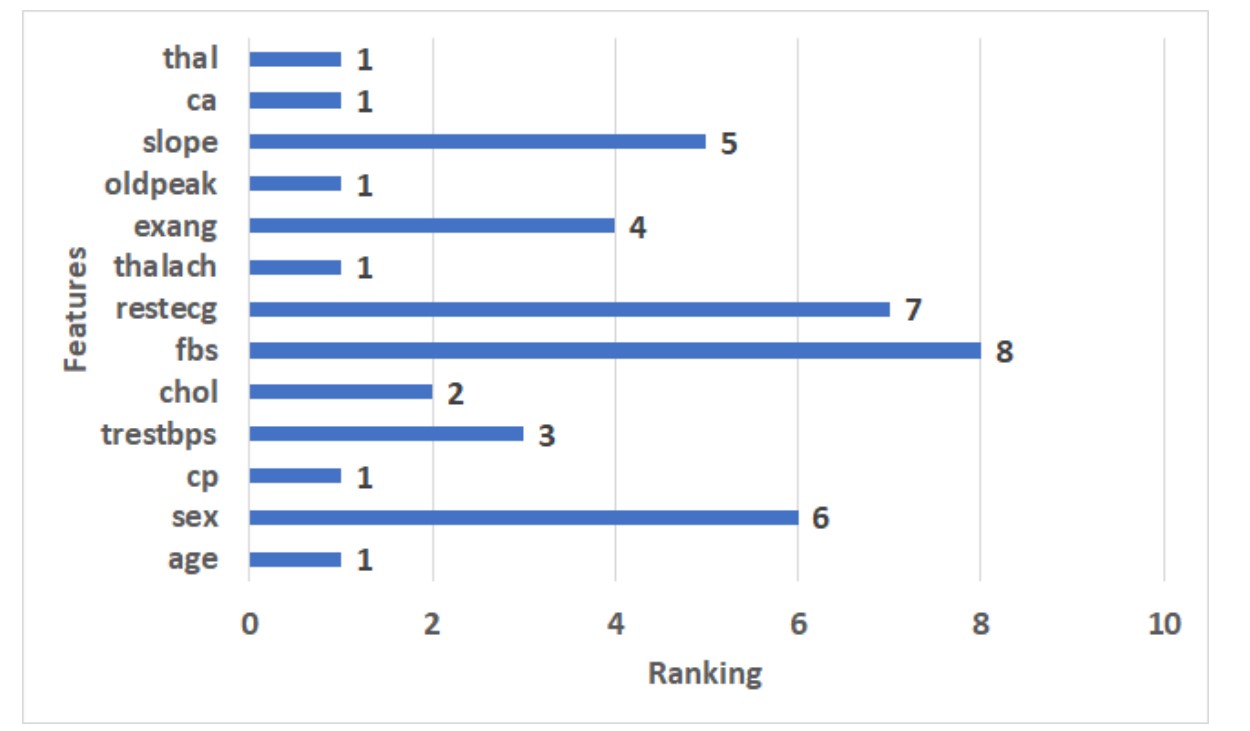
• Results of the full features: For ML models, RF and LR register approximately the same highest scores (75.32% of ACC, 75.44% of PRE, 75.32% of REC, 75.33% of F1) and (75.60% of ACC, 75.60% of PRE, 75.60% of REC, 75.60% of F1), respectively. NB records the worst scores (60.87% of ACC, 64.98% of PRE, 60.87% of REC, 56.69% of F1). KNN registers the second-highest scores (73.16% of ACC, 73.47% of PRE, 73.16% of REC, 73.16% of F1). For hybrid models, CNN-LSTM has the highest scores (76.64% of ACC, 76.9% of PRE, 76.64% of REC, and 76.65% of F1). CNN-GRU records the lowest scores (75.63% of ACC, 75.65% of PRE, 75.63% of REC, 75.58% of F1). The proposed model records the highest scores (ACC = 78.81%, 78.1% of PRE, 78.81% of REC, and 78.81% of F1) compared to other models. It improves ACC by 2.17, PRE by 1.2, REC by 2.17, and F1 by 2.16 compared to CNN-LSTM

. • Results of the selected features: For ML models, RF and LR register approximately the same highest scores (73.02% of ACC, 73.06% of PRE, 73.02% of REC, 73.03% of F1) and (73.58% of ACC, 73.60% of PRE, 73.58% of REC, = 73.59% of F1), respectively. NB records the worst scores (60.84% of ACC, 64.97% of PRE, 60.84% of REC, F1 = 56.63%). KNN registers the second-highest scores (72.59% of ACC, 72.92% of PRE, 72.59% of REC, F1 = 72.59%). The top scores for hybrid models belong to CNN-LSTM (75.22% of ACC, 75.42% of PRE, 75.22% of REC, and 75.22% of F1). The lowest scores are recorded by CNN-GRU (74.07% of ACC, 74.23% of PRE, 74.07% of REC, and 74.08% of F1). In comparison to other models, the proposed model achieves the greatest scores (77.42% of ACC, 77.99% of PRE, 77.42% of REC, and 77.39% of F1). In comparison to CNN-LSTM, it enhances ACC by 2.2%, PRE by 2.57%, REC by 2.2%, and F1 by 2.17%..

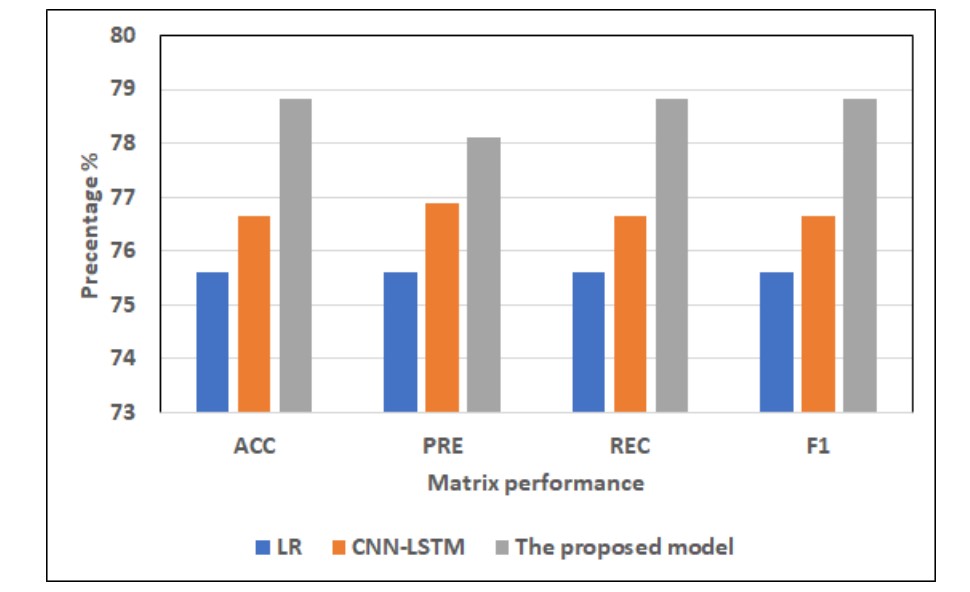


**Results of the Cleveland Dataset**

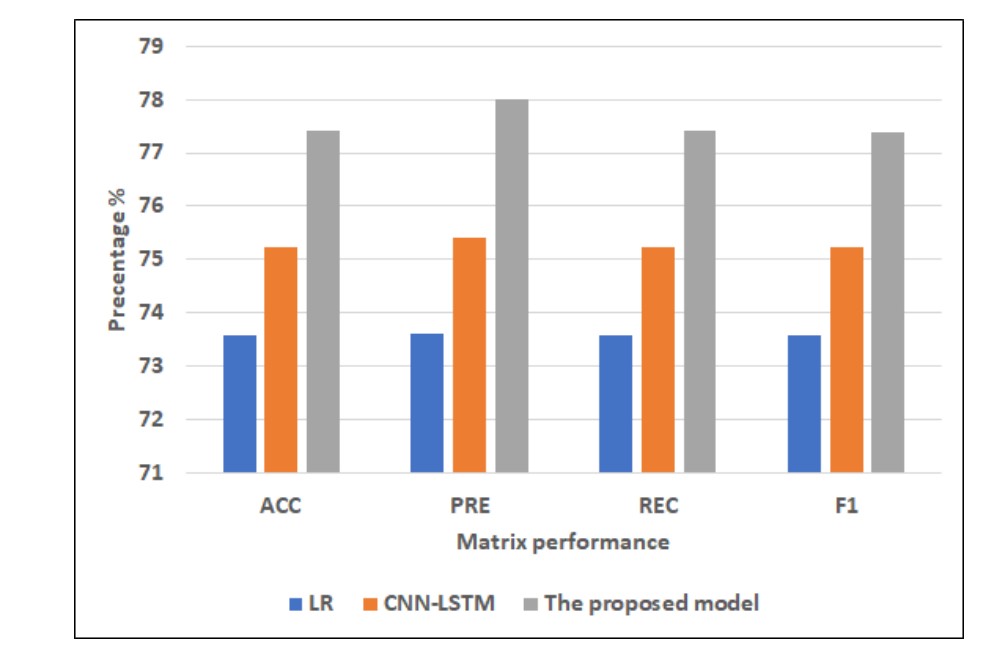
Feature Selection Results In the experiments, we used the RFE to extract the important features from the Cleveland dataset. It assigns features a value of ranking, with the critical features having a ranking of 1, and the least important features having a ranking of 8. The features ranking is shown in Figure 5. We can see that the 8 most significant features have a ranking of 1: age, cp, thalach, oldpeak, ca, and thal. The least important feature has a ranking of 8, which is fbs.



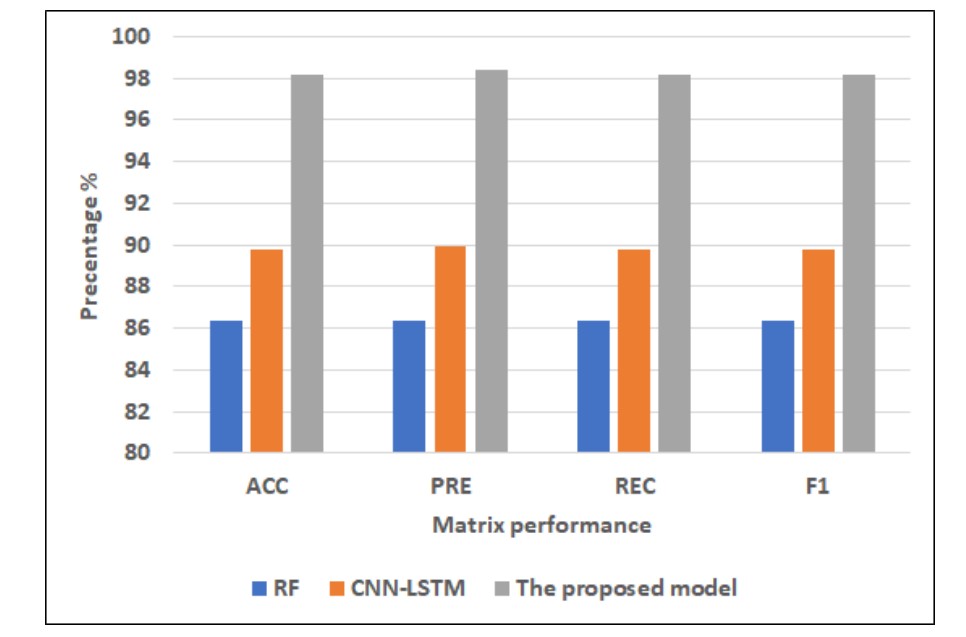
The ranking features for the Cleveland dataset. 4.3.2. Results of the Applied Models This section presents the setting of values parameters for models and the results of applied ML, hybrid models, and the proposed model with the full and selected features for the Cleveland dataset. The following settings were modified for CNN-LSTM and CNN-GRU hybrid models: batch size = 50, epoch = 50, learning rate = 0.00004, and the optimizer used is Adam. Some of the best CNN-LSTM and CNN-GRU hyperparameter values as determined by KerasTuner are shown in Table 5. Table 6 shows the results of applying ML, hybrid models, and the proposed model with full features and selected features by RFE to the Cleveland dataset. • Full features For ML models, RF has the highest scores (86.34% of ACC, 86.34% of PRE, 86.34% of REC, and 86.34% of F1). NB records the lowest scores (60.00% of ACC, 60.05% of PRE, 60.00% of REC, 59.74% of F1). DT registers the second-highest scores (82.44% of ACC, 82.46% of PRE, 82.44% of REC, 82.44% of F1). For hybrid models, CNN-LSTM has the highest scores (89.76% of ACC, 89.96% of PRE, REC = 89.76% of REC, F1 = 89.75%). CNN-GRU records the lowest scores (88.29% of ACC, 89.06% of PRE, REC = 88.29% of REC, 88.26% of F1). The proposed model records the highest scores (97.17% of ACC, 97.42% of PRE, 97.17% of REC, 97.15% of F1) compared to the other models. It improves ACC by 7.41, PRE by 7.46, REC by 7.41, and F1 by 7.4 compared to CNN-LSTM. • Selected features For ML models, RF has the highest scores (82.93% of ACC, 82.99% of PRE, 82.93% of REC, 82.91% of F1). NB records the lowest scores (64.88% of ACC, 64.90% of PRE, 64.88% of REC, 64.88% of F1). DT registers the second-highest scores (81.95% of ACC PRE = 82.01%, 81.95% of REC, 81.93% of F1). For hybrid models, CNN-LSTM has the highest scores (86.34% of ACC, 86.41% of PRE, 86.34% of REC, and 86.34% of F1). CNN-GRU records the lowest scores (85.85% of ACC, 86.92% of PRE, 85.85% of REC, 85.78% of F1). The proposed model records the highest scores (91.22% of ACC, 91.29% of PRE, 91.22% of REC, 91.22% of F1) compared to other models. It improves ACC by 4.88, PRE by 4.88, REC by 4.88, and F1 by 4.88 compared to CNN-LSTM. Table 5. The best values of the parameters for the Cleveland dataset.



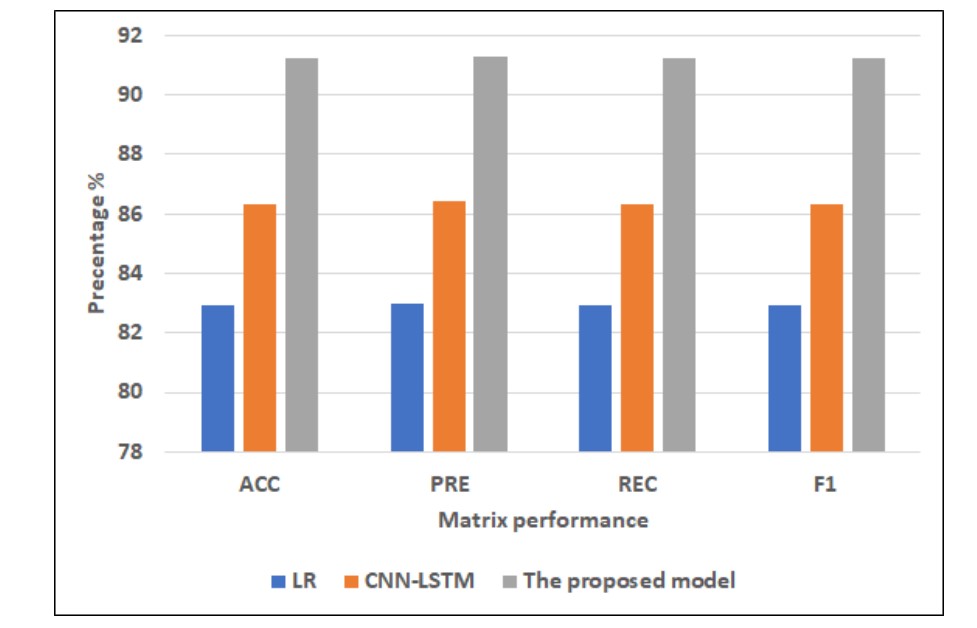
The best models for applying models with full features for Dataset 1. 4.4.2. Cleveland Dataset Figures 8 and 9 show the best models for applying models with full features and selected features. We can see that the proposed model has achieved the highest scores with full features at ACC = 98.17%, PRE = 98.42%, REC = 98.17%, and F1 = 98.15% compared to other models with full features and selected features, and it improves ACC by 3.41, PRE by 3.46, REC by 3.41, and F1 by 3.4 compared to CNN-LSTM. In addition, it has the highest scores with selected features at (ACC = 91.22%, PRE = 91.29%, REC = 91.22%, F1 = 91.22%, and it improves ACC by 4.88, PRE by 4.88, REC by 4.88 and F1 by 4.88 compared to CNNLSTM. RF has the lowest scores with full features, and LR has the lowest scores with the selected features.



The best models for applying models with selected features for Dataset 1



The best models for applying models with full features for Dataset 2.



The best models for applying models with selected features for Dataset 2.

**CHAPTER-10**

**CONCLUSION**

## CONCLUSION

The study proposed a deep staking ensemble to improve the performance of heart disease prediction. The proposed model was based on the integration of two pre-trained and optimized deep hybrid models: CNN-LSTM and CNN-GRU. The SVM classifier has been used as the meta-learner model. The first hybrid model was the CNN-LSTM model, which combined CNN and LSTM layers. The second hybrid model was the CNN-GRU model, which combined CNN with GRU models. RFE was used to choose the most important features from two heart disease datasets. The proposed models were compared with five classical ML models, including LR, RF, K-NN, DT, NB, and hybrid models (i.e., CNN-LSTM and CNN-GRU). Results were collected with the full feature set and a selected feature set. Compared to other models, the result generated by the proposed model had the optimum performance with all the features. For the first dataset, the proposed model had the highest ACC of 78.81%, PRE of 78.1%, REC of 78.81%, and F1 of 78.81. For the Cleveland dataset, the proposed model had the highest ACC of 97.17%, PRE of 97.42%, REC of 97.17%, and F1 of 97.15%. In addition, the proposed model achieved better results than the literature. As a result, the proposed model can improve the disease prediction and can improve the quality of life of the heart disease patients. In the future, we will test the performance of the proposed model with other datasets. We will extend the model by adding other modalities such as images and EEG data. We will provide interpretability features to the proposed model.

**CHAPTER-11**

**REFERENCES**

## REFERENCES

[1]Cardiovascular diseases

https://www.who.int/westernpacific/health-topics/cardiovascular-diseases

accessed Apr. 10, 2020

[2]A. Mdhaffar, I. Bouassida Rodriguez, K. Charfi, L. Abid, B. Freisleben

CEP4HFP: complex event processing for heart failure prediction

IEEE Trans NanoBioscience, 16 (8) (Dec. 2017), pp. 708-717, 10.1109/TNB.2017.2769671

[3]B. Jin, C. Che, Z. Liu, S. Zhang, X. Yin, X. Wei

Predicting the risk of heart failure with EHR sequential data modeling

IEEE Access, 6 (2018), pp. 9256-9261, 10.1109/ACCESS.2017.2789324

[4]L. Ali, A. Rahman, A. Khan, M. Zhou, A. Javeed, J.A. Khan

An automated diagnostic system for heart disease prediction based on $\chi^2$ statistical model and optimally configured deep neural network

IEEE Access, 7 (2019), pp. 34938-34945, 10.1109/ACCESS.2019.2904800

[5]C.B.C. Latha, S.C. Jeeva

Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques

Informatics in Medicine Unlocked, 16 (Jan. 2019), p. 100203, 10.1016/j.imu.2019.100203

[6]L. Ali, et al.

An optimized stacked support vector machines based expert system for the effective prediction of heart failure

IEEE Access, 7 (2019), pp. 54007-54014, 10.1109/ACCESS.2019.2909969

[7]R.K. Sevakula, N.K. Verma

Assessing generalization ability of majority vote point classifiers

IEEE Transactions on Neural Networks and Learning Systems, 28 (12) (Dec. 2017), pp. 2985-2997, 10.1109/TNNLS.2016.2609466

[8]H. Li, et al.

Ensemble learning for overall power conversion efficiency of the all-organic dye-sensitized solar cells

IEEE Access, 6 (2018), pp. 34118-34126, 10.1109/ACCESS.2018.2850048

[9]Tin Kam Ho

The random subspace method for constructing decision forests

IEEE Trans Pattern Anal Mach Intell, 20 (8) (Aug. 1998), pp. 832-844, 10.1109/34.709601

[10]L. Breiman

Bagging predictors

Mach Learn, 24 (2) (Aug. 1996), pp. 123-140, 10.1023/A:1018054314350

View at publisher

[11]R.E. Schapire, Y. Singer

Improved boosting algorithms using confidence-rated predictions

Mach Learn, 37 (3) (Dec. 1999), pp. 297-336, 10.1023/A:1007614523901

[12]F. Leon, S.-A. Floria, C. Bădică

Evaluating the effect of voting methods on ensemble-based classification,

2017 IEEE international conference on INnovations in intelligent SysTems and applications (INISTA) (Jul. 2017), pp. 1-6, 10.1109/INISTA.2017.8001122

[13]R.E. Banfield, L.O. Hall, K.W. Bowyer, W.P. Kegelmeyer

A comparison of decision tree ensemble creation techniques

IEEE Trans Pattern Anal Mach Intell, 29 (1) (Jan. 2007), pp. 173-180, 10.1109/TPAMI.2007.250609

[14]D. Ruta, B. Gabrys, C. Lemke

A generic multilevel architecture for time series prediction

IEEE Trans Knowl Data Eng, 23 (3) (Mar. 2011), pp. 350-359, 10.1109/TKDE.2010.137

[15]B. Zhang, et al.

Ensemble learners of multiple deep CNNs for pulmonary nodules classification using CT images

IEEE Access, 7 (2019), pp. 110358-110371, 10.1109/ACCESS.2019.2933670

[16]L. Han, S. Luo, J. Yu, L. Pan, S. Chen

Rule extraction from support vector machines using ensemble learning approach: an application for diagnosis of diabetes

IEEE Journal of Biomedical and Health Informatics, 19 (2) (Mar. 2015), pp. 728-734, 10.1109/JBHI.2014.2325615

[17]F. Shang, et al.

VR-SGD: a simple stochastic variance reduction method for machine learning

IEEE Trans Knowl Data Eng, 32 (1) (Jan. 2020), pp. 188-202, 10.1109/TKDE.2018.2878765

[18]S.H. Ebenuwa, M.S. Sharif, M. Alazab, A. Al-Nemrat

Variance ranking attributes selection techniques for binary classification problem in imbalance data

IEEE Access, 7 (2019), pp. 24649-24666, 10.1109/ACCESS.2019.2899578

[19]R. Rivera-Lopez, J. Canul-Reich

Construction of near-optimal axis-parallel decision trees using a differential-evolution-based approach

IEEE Access, 6 (2018), pp. 5548-5563, 10.1109/ACCESS.2017.2788700

[20]M. Woźniak

Application of combined classifiers to data stream classification

K. Saeed, R. Chaki, A. Cortesi, S. Wierzchoń (Eds.), Computer information Systems and industrial management, vol. 8104, Springer Berlin Heidelberg, Berlin, Heidelberg (2013), pp. 13-23

View at publisher

[21]M. Wozniak

Accuracy based weighted aging ensemble (AB-WAE) — algorithm for data stream classification

2017 IEEE 4th international Conference on soft computing machine intelligence (ISCMI) (Nov. 2017), pp. 21-24, 10.1109/ISCMI.2017.8279591

[22]UCI machine learning repository: heart disease data set

http://archive.ics.uci.edu/ml/datasets/Heart+Disease

accessed Apr. 09, 2020

[23]Framingham Heart study dataset

https://kaggle.com/amanajmera1/framingham-heart-study-dataset

accessed Jan. 24, 2020

[24]S. Mohan, C. Thirumalai, G. Srivastava

Effective heart disease prediction using hybrid machine learning techniques

IEEE Access, 7 (2019), pp. 81542-81554, 10.1109/ACCESS.2019.2923707

[25]A.N. Repaka, S.D. Ravikanti, R.G. Franklin

Design and implementing heart disease prediction using naives bayesian

2019 3rd international Conference on Trends in Electronics and informatics (ICOEI) (Apr. 2019), pp. 292-297, 10.1109/ICOEI.2019.8862604

[26]O.W. Samuel, G.M. Asogbon, A.K. Sangaiah, P. Fang, G. Li

An integrated decision support system based on ANN and Fuzzy\_AHP for heart failure risk prediction

Expert Syst Appl, 68 (Feb. 2017), pp. 163-172, 10.1016/j.eswa.2016.10.020

**CHAPTER-12**

**PUBLICATION**

# Deep Learning-Based Approaches for Accurate and Early Prediction of Heart Disease

SHAIK UMER ALI SIDDIQUE1, SOHAIL AHMED1

SIDDANI SUMA SRI VARSHA1, SATTENAPALLI BABY KRISHNA RANI11 CH. MURALI KRISHNA

1UG Students, NRI Institute of Technology -CSE(DATA SCIENCE), Vijayawada, Andhra Pradesh,

2 NRI Institute of Technology- ASSOCIATE PROFFECER CSE (DATA SCIENCE)2

, Vijayawada, Andhra Pradesh,

[3suasiddique@gmail.com,](mailto:3suasiddique@gmail.com)

# Abstract

Many epidemics have afflicted humanity throughout history, claiming many lives. It has been noted in our time that heart disease is one of the deadliest diseases that humanity has confronted in the contemporary period. The proliferation of poor habits such as smoking, overeating, and lack of physical activity has contributed to the rise in heart disease. The killing feature of heart disease, which has earned it the moniker the “silent killer,” is that it frequently has no apparent signs in advance. As a result, research is required to develop a promising model for the early identification of heart disease using simple data and symptoms. The paper aims to propose a deep stacking ensemble model to enhance the performance of the prediction of heart disease. The proposed ensemble model integrates two optimized and pre-trained hybrid deep learning models with the Support Vector Machine (SVM) as the meta-learner model. The first hybrid model is Convolutional Neural Network (CNN)-Long Short-Term Memory (LSTM) (CNN-LSTM), which integrates CNN and LSTM. The second hybrid model is CNN-GRU, which integrates CNN with a Gated Recurrent Unit (GRU). Recursive Feature Elimination (RFE) is also used for the feature selection optimization process. The proposed model has been optimized and tested using two different heart disease datasets. The proposed ensemble is compared with five machine learning models including Logistic Regression (LR), Random Forest (RF), K-Nearest Neighbours (K-NN), Decision Tree (DT), Naïve Bayes (NB), and hybrid models. In addition, optimization techniques are used to optimize ML, DL, and the proposed models. The results obtained by the proposed model achieved the highest performance using the full feature set

**Keywords: machine learning; deep learning; ensemble learning; heart disease**

# Introduction

Heart disease is among the most common illnesses that persisted in the past and have increased and spread in our present. The reasons for the increase in its rates are varied, especially in our modern age. Diabetes, hypertension, cholesterol, erratic heartbeat, and many more clinical signs are some biological markers and risk factors that are needed to diagnose heart disease. World Health Organization (WHO) claims that one of the main and highly-ranked causes of death worldwide is heart disease, which can have several forms such as ischemic, hypertensive, and vascular heart disease [1], and it has been shown that cardiovascular illnesses kill 17.9 million patients each year. In addition, unhealthy behavior that results in being overweight, obesity, and hypertension raises the risk of heart disease [1]. In addition, the heart is one of the essential organs of the human body. It is primarily responsible for the continuity of pumping the blood needed for the work of the rest of the human body. However, it is difficult for the heart to maintain the same efficiency throughout a person’s life. The heart is exposed to many problems that can occur because of several different reasons, such as bad health and nutritional habits or aging [2]. Therefore, finding methods and techniques that allow for the early detection or even prediction of potential heart problems has become inevitable. This can help doctors and healthcare organizations to reduce the problems and complications of the disease. Artificial intelligence (AI) based on machine learning (ML) and deep learning (DL) has conducted

Key roles in evaluating medical data to assist in illness diagnosis to determine the appropriate treatment. It is used to find patterns automatically from the clinical data and then reason about clinical data to predict the early risk for patients such as heart disease [3], cancer disease [4,5], and COVID-19 [6,7]. Recently, deep learning algorithms such LSTM, GRU, CNN, and hybrid models of these algorithms have played an important role in strengthening and enhancing the level of heart disease prediction using various layers that could collect deeper features [8–11] Recently, authors have used ensemble learning to enhance the performance of these models in the healthcare domain [12]. Ensemble learning combines the decisions of various base classifiers using many techniques such as voting or averaging to improve the final decision [13]. Ensemble algorithms can be categorized into three branches: boosting [14], stacking [15], and bagging [16]. Stacking ensemble is considered as the best technique for building ensemble models because it is based on a metalearner, which learns from data how to weight the base classifiers and combine them in the best way to optimize the performance of the resulting model. Ensemble stacking optimizes a set of heterogeneous base models and combines their decisions using a meta-learner [15]. In this study, we proposed an optimized ensemble stacking model that merged the two pre-trained hybrid models of CNN-LSTM and CNN-GRU with a meta-learner (SVM) to enhance the performance of heart disease prediction. In addition, Recursive Feature Elimination (RFE) has been used to choose the most informative features from two heart disease datasets. Our contributions can be summarized as follows:

* + We proposed two hybrid models with heterogeneous architectures: CNN-LSTM and CNN-GRU were proposed and optimized.
  + We proposed a stacking ensemble model that merged the previous pre-trained hybrid models of CNN-LSTM and CNN-GRU. The best meta-learner classifier has been selected based on the experimental results. The SVM algorithm achieved the best results as the meta-classifier to determine the best weights of the base classifiers;
  + We compared the proposed model with different ML models using two benchmark heart disease datasets;
  + The proposed model significantly outperformed all other models and achieved the best results. The remainder of the paper is structured as follows: Section 2 discusses heart disease- related works. The section describes the main phases and approaches in Section 3 of predicting heart disease. Section 4 describes the results and discussion results. Finally, the paper is concluded in Section 5.

# Related Work

Machine learning and deep learning have been used to predict heart disease. For example Kavitha M. et al. [17] suggested a hybrid model that combines DT and RF to predict heart disease using the Cleveland dataset. They contrasted the hybrid model’s performance with that of DT and RF. Ishaq A. et al. [18] applied different ML algorithms: SVM, DT, LR, NB, Adaptive boosting (AdaBoost), Stochastic Gradient Descent (SGD), RF, Gradient Boosting Machine (GBM), and Extra Tree Classifier (ETC) using the Cleveland heart disease dataset to analyzes the heart failure. The results showed that ETC gave the best performance and outperformed other models. Ansarullah, S. I. et al. [19] used ML algorithms to predict heart disease: NB, RF, DT, K- NN, and SVM. The dataset was gathered in Kashmir from many heterogeneous data sources (India). The results showed that RF has the best model performance. Many authors applied feature selection methods with ML and DL models to predict heart disease. For example, Spencer R. et al. [20] used Chi2, ReliefF, symmetrical uncertainty (SU), and PCA feature selection methods to extract the important features from four heartdisease datasets. They applied BayesNet, Logistic, Stochastic Gradient Descent (SGD), and KNN Adaboost to the full and

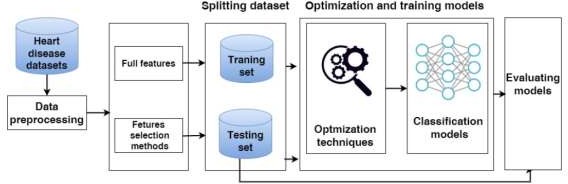
selected features. The result showed that the BayesNet model was recorded as the best performer using the Chi-2 feature selection compared with other models. Bharti R. et al. [21] used the Lasso algorithm to select features from the heart disease dataset. They applied ML and DL models: LR, KNN, SVM, RF, DT, and ANN, respectively. The results showed that ANN has the best performance compared to ML models. Gokulnath C. B. et al. [22] used KNN, MLP, SVM, and J48 for heart disease detection. The datasets were gathered from a variety of sources. The authors applied various feature selection strategies, including the extra tree classifier, gradient boosting classifier, random forest, recursive feature removal, and XG boost classifier. In the study by Amin, M. S. et al. [23], in order to increase the prediction accuracy, the authors proposed a voting hybrid model based on NB and LR. They used k-NN, DT, NB, LR, SVM, Neural Network (NN), and the hybrid model to choose meaningful characteristics from the Cleveland heart disease dataset. The hybrid model was given the best performance compared to other models. Bashir S. et al. [24] used DT, LR, NB, SVM, and RF models with feature extraction methods with the Cleveland heart disease dataset to predict heart disease. The results showed that LR and SVM with feature selection methods had better accuracy than the other models. Javid I. et al. [25] developed model-based GRU and RF (GRU-RF) for heart disease detection. The GRU-RF was compared with RF, GRU, KNN, and DNN algorithms and achieved the best performance. Chae M. et al. [26] proposed a hybrid model, LSTM–GRU, and compared it with DT, RF, LR, LSTM, and GRU to predict heart disease. They used the dataset from Soonchunhyang University Cheonan Hospital in Korea to train and test the models. They improved the performance models based on hyperparameter adjustment, the quantity of primary patient data, and input parameters. The results indicate that when compared to other models, the GRU model outperforms the others. In the study by Narmadha, S. et al. [27], the authors used LSTM and GRU hyperparameter tuning to enhance the performance of the algorithms. The outcomes demonstrated that the GRU provides better accuracy than the LSTM across the board. The authors have used ensemble models to predict heart disease. For example, Adhikari, B. et al.

[28] applied LR, SVM, DT, K-NN, GNB, and ensemble models using a dataset collected from the UCI heart disease dataset. They used the voting and averaging ensemble models built by combining the ML above models. The results showed that the ensemble model was the best performer compared with other models. Javid, I. et al. [29] used RF, SVM, K-NN, LSTM, Hard Voting Ensemble Model, and GRU for heart disease prediction. The results showed that the Hard Voting Ensemble Model recorded higher accuracy compared to other models. Ghosh P. et al. [30] proposed hybrid models that integrated boosting and bagging with traditional ML models: KNN, DT, and RF. The hybrid models: K-NN Bagging Method (KNNBM), DT-Bagging Method (DTBM), AdaBoost (AB), and Random Forest Bagging Method (RFBM) were applied to heart disease datasets. Relief, Least Absolute Shrinkage, and Selection Operator were the three feature selection approaches they used (LASSO). When compared to other models, the RFBM model showed the best performance. Previous studies do not use ensemble stacking based on heterogeneous hybrid deep learning models to predict heart disease. In addition, most previous studies have used the Cleveland Heart Disease database to perform this experiment. In our work, we used a new large heart disease dataset, and we proposed ensemble stacking models based on optimizing different heterogeneous hybrid models: CNN-LSTM and GRU-LSTM

# Methodology

In this study, we evaluate three approaches: the classical machine learning approach, the hybrid models approach, and a proposed model. These models are applied to the full feature set and selected feature set. The proposed model for predicting heart disease has several steps including data collection, data preprocessing, data splitting, feature

selection, and evaluation models, as shown in Figure 1. Each phase is described in detail as follows



**Figure 1. The phases of predicting heart disease.**

## Heart Disease Datasets In our work

we used two heart disease datasets.3.1.1. Dataset 1 We used the large heart disease dataset (Heart Disease) [31]. This data includes 18 independent features and one dependent variable as the class label for predicting heart disease. The class label includes two values: 0 represents the healthy class label, and 1 represents the heart disease class label. Table 1 presents the number of medical records for each class in the training and testing sets. The description of each feature is described in a Supplementary File.

3.1.2. Cleveland Dataset The Cleveland dataset [32] includes 13 independent variables as features and one dependent variable as the class label used to diagnose heart disease. The class label includes two values: 0 represents the healthy class label, and 1 represents the heart disease class label. Table 1 presents the number of medical records for each class in the training and testing sets of the Cleveland heart disease dataset. The description of each feature is described in the Supplementary File.

## Data Pre-Processing

The first heart disease dataset includes 14 numeric features and four categorical features. The data was pre-processed after collection as follows: removing duplicate records and encoding category data into numerical data such as smoking and skin cancer.

## Data Splitting

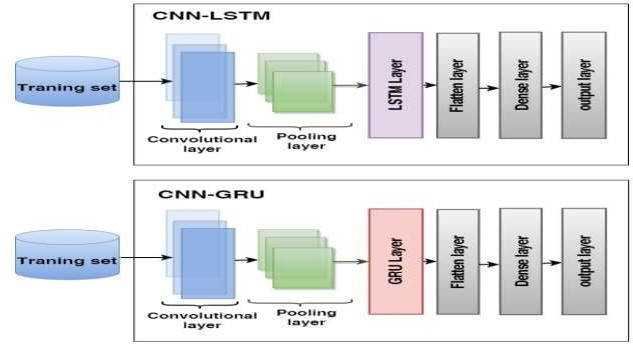
The two datasets are divided into two sets using a stratified sampling method: 80% training sets and 20% testing sets. Models are trained and optimized using training data. The test set is employed to assess and test the model. The stratified sampling method is one way of splitting the dataset used to get samples that accurately reflect the distribution of classes in the population. It separates the dataset into homogeneous subsets; each subset contains the same percentage of every class. [33,34]. This method has been used in studies of different fields of healthcare [35–37]

## Feature Selection Methods

In our work, we use the Recursive Feature Elimination (RFE) feature selection method to extract the most informative features from each dataset. The RFE determines the essential features by figuring out a high correlation between features and the target [38]. It assigns one value as a ranking for features if the features have high collaboration with the target. A novel RFE strategy is recently presented that used RF and SVM to evaluate features rather than classification performance and selects the minor significant features for deletion [39,40]

## Machine Learning Approach

* + 1. **ML Algorithms**



We tested many classical ML models from different families including SVM [41– 44], Logistic Regression (LR) [45,46], Nave Bayes (NB) [47], Decision tree (DT) [48], Random Forest (RF) [49,50], and K-nearest Neighbors (k-NN) [51].

## Optimization Techniques for Classical Models

Grid search is employed to fine-tune hyperparameters of different classical ML models by generating discrete grids within the hyperparameter domain and selecting the list of parameters that give the best performance [52]. Data is split into two segments using the cross-validation technique: one is used to train and validate the models (training set), and the other is utilized for model testing (testing set) [19]. The training set has been used to validate the models using the k-fold cross-validation technique.

## The Hybrid Models

* + 1. **The Hybrid Model Architectures**

We proposed two hybrid models: CNN-LSTM and CNN-GRU for predicting heart disease. The structures of hybrid models are illustrated in Figure 2.

The first model is CNN-LSTM, which combines CNN with LSTM, consisting of a convolutional layer, a max-pooling layer, an LSTM layer, a flattened layer, a fully connected, and an output layer;

The second model is CNN-GRU, which combines CNN with GRU. The architecture consists of a convolutional layer, a max-pooling layer, a GRU layer, a flattened layer, a fully connected, and an output layer.

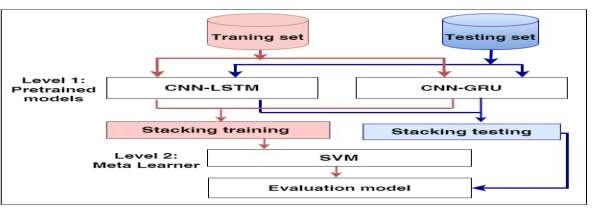
**Figure 2. The architecture of the hybrid models CNN-LSTM and CNN-GRU is used to predict heart disease.**

## Optimization Techniques for Hybrid Models

The Bayesian optimizer is used to optimize the hybrid models. This search technique quickly generates the search space and locates the best hyperparameter values for the models [53]. We adopt the parameter settings for CNN-LSTM and CNN-GRU.

## The Proposed Stacking Ensemble Model

In this work, our model is developed using two levels: Level-1 and Level-2, as shown in Figure 3. Level 1 begins by loading the pre-trained models of hybrid models CNN-LSTM and CNN-GRU, and the layers of the models are frozen except for the last layers. The models anticipate the training set’s output probabilities and subsequently integrate them into stacking training. Secondly, the models estimate the output probabilities of the testing set and aggregate them in stacking testing. At Level 2, SVM, as a meta-learner, is trained and optimized using stacking training and Grid search, respectively, while producing the final results using stacking testing.



**Figure 3. The proposed model for predicting heart disease.**

## Evaluating Models:

The metrics for classification performance that are most frequently employed are accuracy (ACC), precision (PRE), recall (REC), and F1-score (F1). In contrast to the True Positive (TP), which denotes that the person is ill and the test is positive, the True Negative (TN) shows that the person is healthy and the result is negative. False positives are tests that come back positive even when the subject is healthy (FP). When a test is negative, but the subject is ill, it is known as a false negative (FN).

Accuracy =

TP + TN

TP + FP + TN + FN

Precision = TP TP + FP

Recall = TP

TP + FN

F1-score = 2 · precision · recall

precision + recall

## Experimental Results

In this section, we describe the rank of features after applying the RFE to the two datasets. Moreover, we describe the results of the performance of using ML models (SVM, LR,

RF, NB, and KNN), the hybrid models (CNN-LSTM, CNN-GRU), and the proposed model to full and selected features.

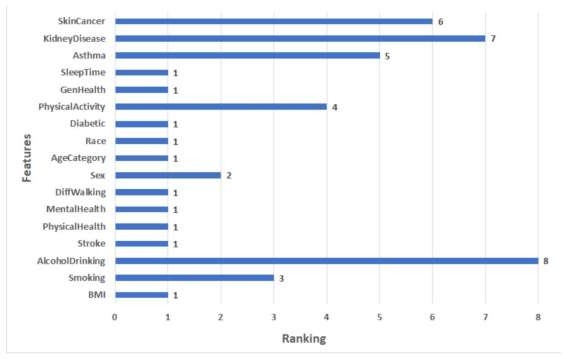
## Experimental Setup

The experiments in this paper are implemented using Google Colab with Python libraries such as Scikit-learn, TensorFlow, and others. We used grid-search and the Bayesian optimizer to optimize the ML and hybrid models. We used the RFE technique to identify the best features from the two datasets. The two datasets are separated into two sets: 80% training and 20% testing set using the stratified methods. The models are trained and tested by utilizing the training and testing sets, respectively

## Results of Dataset1

* + 1. **Feature Selection Results**

In the experiments, we used the RFE to extract the important features from the heart disease dataset by assigning a ranking for every feature. The critical features are ranked 1, and the least important features are ranked 8. The features ranking is shown in Figure 4. We can see that the most significant 10 features have a ranking of 1: BMI, Stroke, Physical Health, Mental Health, difficulty walking, Age Category, Race, diabetes, Gen Health, and Sleep Time. The lowest important feature has a ranking of 8, which is alcohol drinking.



**Figure 4 features ranking**

## Results of Applying Models

This section presents the ACC, PRE, REC, and F1 of ML, hybrid models, and the proposed model for Dataset 1. In the hybrid models CNN-LSTM and CNN-GRU some parameters were adapted: batch size of 500, epoch = 50, learning rate = 0.00004, and the optimizer used is Adam. Some of the best values of CNN-LSTM and CNN-GRU hyperparameters that were selected by Keras Tuner are shown in Table 1.

Table 1. The best values of the parameters for CNN-LSTM and CNN-GRU

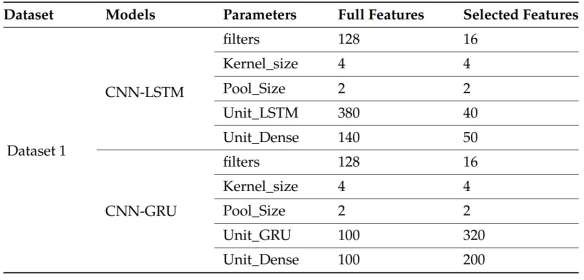
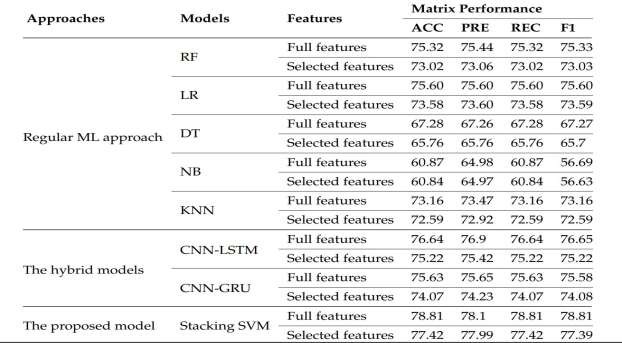


Table 2 shows the results of applying ML, hybrid models, and the proposed model with full features and selected features by RFE to the heart disease Dataset 1.

* Results of the full features: For ML models, RF and LR register approximately the same highest scores (75.32% of ACC, 75.44% of PRE, 75.32% of REC, 75.33% of F1) and (75.60% of ACC, 75.60% of PRE, 75.60% of REC, 75.60% of F1), respectively. NB records the worst scores (60.87% of ACC, 64.98% of PRE, 60.87% of REC, 56.69% of F1). KNN registers the second-highest scores (73.16% of ACC, 73.47% of PRE, 73.16% of REC, 73.16% of F1). For hybrid models, CNN-LSTM has the highest scores (76.64% of ACC, 76.9% of PRE, 76.64% of REC, and 76.65% of F1). CNN-GRU records the lowest scores (75.63% of ACC, 75.65% of PRE, 75.63% of REC, 75.58% of F1). The proposed model records the highest scores (ACC = 78.81%, 78.1% of PRE, 78.81% of REC, and 78.81% of F1) compared to other models. It improves ACC by 2.17, PRE by 1.2, REC by 2.17, and F1 by 2.16 compared to CNN-LSTM

. • Results of the selected features: For ML models, RF and LR register approximately the same highest scores (73.02% of ACC, 73.06% of PRE, 73.02% of REC, 73.03% of F1) and (73.58% of ACC, 73.60% of PRE, 73.58% of REC, = 73.59% of F1), respectively. NB records the worst scores (60.84% of ACC, 64.97% of PRE, 60.84% of REC, F1 = 56.63%). KNN registers the second-highest scores (72.59% of ACC, 72.92% of PRE, 72.59% of REC, F1 = 72.59%). The top scores for hybrid models belong to CNN-LSTM (75.22% of ACC, 75.42% of PRE, 75.22% of REC, and 75.22% of F1). The lowest scores are recorded by CNN-GRU (74.07% of ACC, 74.23% of PRE, 74.07% of REC, and 74.08% of F1). In comparison to other models, the proposed model achieves the greatest scores (77.42% of ACC, 77.99% of PRE, 77.42% of REC, and 77.39% of F1). In comparison to CNN-LSTM, it enhances ACC by 2.2%, PRE by 2.57%, REC by 2.2%, and F1 by 2.17%.

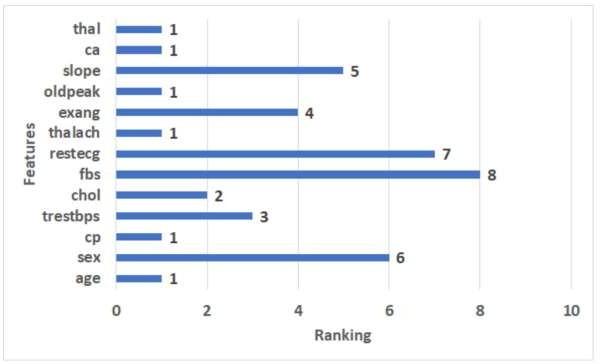
Table 2. Result of applying models with full features and the selected features for Dataset 1.



## Results of the Cleveland Dataset

* + 1. **Feature Selection Results In the experiments**

we used the RFE to extract the important features from the Cleveland dataset. It assigns features a value of ranking, with the critical features having a ranking of 1, and the least important features having a ranking of 8. The features ranking is shown in Figure 5. We can see that the 8 most significant features have a ranking of 1: age, cp, thalach, oldpeak, ca, and thal. The least important feature has a ranking of 8, which is fbs.



**Figure 5. The ranking features for the Cleveland dataset.**

## Results of the Applied Models

This section presents the setting of value parameters for models and the results of applied ML, hybrid models, and the proposed model with the full and selected features for the Cleveland

dataset. The following settings were modified for CNN-LSTM and CNN-GRU hybrid models: batch size = 50, epoch = 50, learning rate = 0.00004, and the optimizer used is Adam. Some of the best CNN-LSTM and CNN-GRU hyperparameter values as determined by KerasTuner are shown in Table 5. Table 6 shows the results of applying ML, hybrid models, and the proposed model with full features and selected features by RFE to the Cleveland dataset

. • Full features For ML models, RF has the highest scores (86.34% of ACC, 86.34% of PRE, 86.34% of REC, and 86.34% of F1). NB records the lowest scores (60.00% of ACC, 60.05% of PRE, 60.00% of REC, 59.74% of F1). DT registers the second-highest scores (82.44% of ACC, 82.46% of PRE, 82.44% of REC, 82.44% of F1). For hybrid models, CNN- LSTM has the highest scores (89.76% of ACC, 89.96% of PRE, REC = 89.76% of REC, F1 = 89.75%). CNN-GRU records the lowest scores (88.29% of ACC, 89.06% of PRE, REC = 88.29% of REC, 88.26% of F1). The proposed model records the highest scores (97.17% of ACC, 97.42% of PRE, 97.17% of REC, 97.15% of F1) compared to the other models. It improves ACC by 7.41, PRE by 7.46, REC by 7.41, and F1 by 7.4 compared to CNN-LSTM

. • Selected features For ML models, RF has the highest scores (82.93% of ACC, 82.99% of PRE, 82.93% of REC, 82.91% of F1). NB records the lowest scores (64.88% of ACC, 64.90% of PRE, 64.88% of REC, 64.88% of F1). DT registers the second-highest scores (81.95% of ACC PRE = 82.01%, 81.95% of REC, 81.93% of F1). For hybrid models, CNN-LSTM has the highest scores (86.34% of ACC, 86.41% of PRE, 86.34% of REC, and 86.34% of F1). CNN-GRU records the lowest scores (85.85% of ACC, 86.92% of PRE, 85.85% of REC, 85.78% of F1).

The proposed model records the highest scores (91.22% of ACC, 91.29% of PRE, 91.22% of REC, 91.22% of F1) compared to other models. It improves ACC by 4.88, PRE by 4.88, REC by 4.88, and F1 by 4.88 compared to CNN-LSTM.

Table 5. The best values of the parameters for the Cleveland dataset

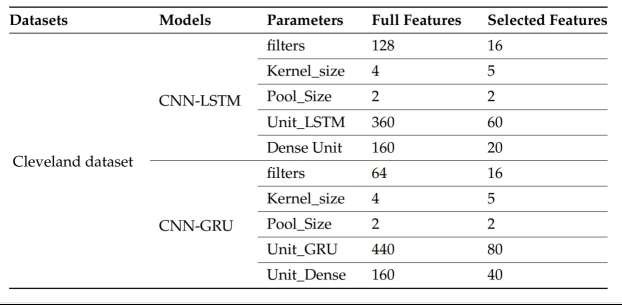
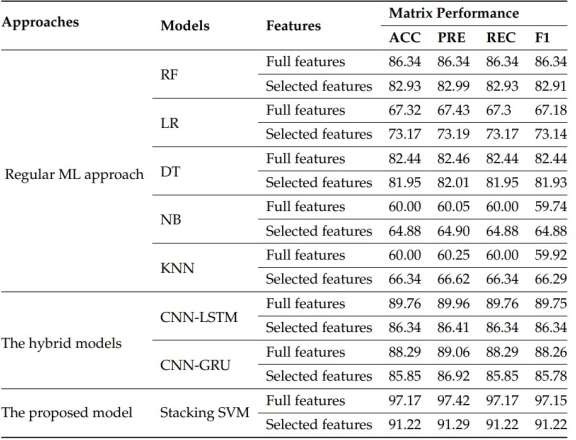
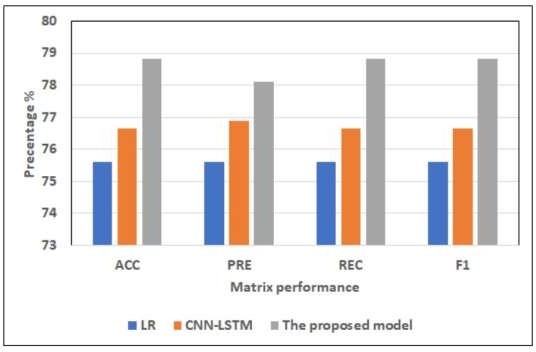


Table 6. Result of applying models with full features and selected features for the Cleveland dataset.



* 1. Discussion We used two heart disease datasets downloaded from Kaggle. We applied RFE feature selection methods to select the essential features. The proposed model, in all cases, has achieved the highest score compared with the other models.

4.4.1. Dataset1 Figures 6 and 7 show the best models for applying models with full features and selected features. We can see that the proposed model has achieved the highest scores with full features at ACC = 78.81%, PRE = 78.81%, REC = 78.81%, and F1 = 78.81% compared to other models with full features and selected features, and It improves ACC by 2.17, PRE by 1.2, REC by 2.17, and F1 by 2.16 compared to CNN-LSTM. In addition, it has the highest scores with selected features at (ACC = 77.42%, PRE = 77.99%, REC = 77.42%, F1 = 77.39%, and it improves ACC by 2.2%, PRE by 2.57%, REC by 2.2%, and F1 by 2.17%. LR has the lowest scores with

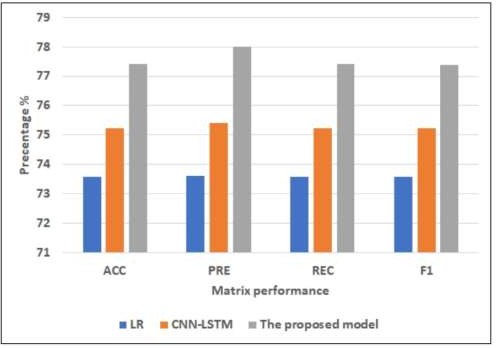


Figure 6. The best models for applying models with full features for Dataset 1. 4.4.2. Cleveland Dataset Figures 8 and 9 show the best models for applying models with full features and selected features. We can see that the proposed model has achieved the highest scores with full features at ACC = 98.17%, PRE = 98.42%, REC = 98.17%, and F1 = 98.15% compared to other models with full features and selected features, and it improves ACC by 3.41, PRE by 3.46, REC by 3.41, and F1 by 3.4 compared to CNN-LSTM. In addition, it has the highest scores with selected features at (ACC = 91.22%, PRE = 91.29%, REC = 91.22%, F1 = 91.22%, and it improves ACC by 4.88, PRE by 4.88, REC by 4.88 and F1 by 4.88 compared to CNNLSTM. RF has the lowest scores with full features, and LR has the lowest scores with the selected features.

Figure 7. The best models for applying models with selected features for Dataset 1

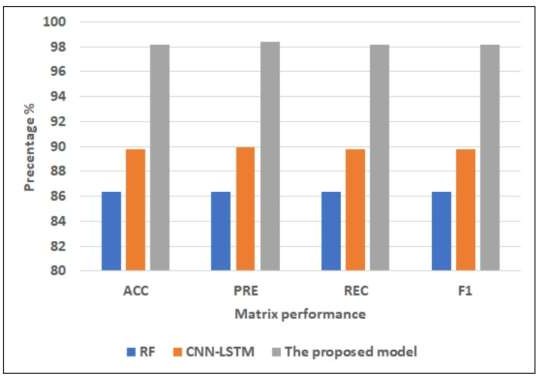


Figure 8. The best models for applying models with full features for Dataset 2.

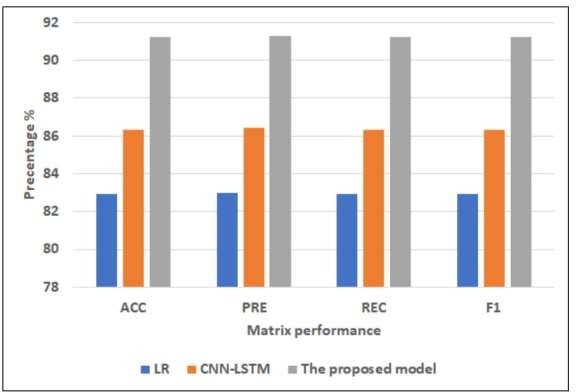


Figure 9. The best models for applying models with selected features for Dataset 2.

4.4.3. Comparison with Literature Studies By assessing the developed model against the current models we could observe that our approach enhanced the scores more than other models. We compared our approach with the approach by authors who used the Cleveland Dataset, as shown in Table 7. The authors of Ref. [17] used a hybrid model combining DT and RF, which recorded 88.7% of ACC. The authors in Refs. [20,22–24,29], used various models, none of which were accurate to more than 90%, which recorded 85%, 88.34%, 87.41%, 84.85%, and 85.71%, respectively. While in Ref. [18,21,28], the authors achieved an accuracy of over 90%. The proposed model has achieved the highest ACC at 98.41% compared to the ACC values in these studies. Table 7. Comparison between previous studies and the proposed model for the Cleveland dataset.

# Conclusions

The study proposed a deep staking ensemble to improve the performance of heart disease prediction. The proposed model was based on the integration of two pre-trained and optimized deep hybrid models: CNN-LSTM and CNN-GRU. The SVM classifier has been used as the meta-learner model. The first hybrid model was the CNN-LSTM model, which combined CNN and LSTM layers. The second hybrid model was the CNN-GRU model, which combined CNN with GRU models. RFE was used to choose the most important features from two heart disease datasets. The proposed models were compared with five classical ML models, including LR, RF, K-NN, DT, NB, and hybrid models (i.e., CNN-LSTM and CNN-GRU). Results were collected with the full feature set and a selected feature set. Compared to other models, the result generated by the proposed model had the optimum performance with all the features. For the first dataset, the proposed model had the highest ACC of 78.81%, PRE of 78.1%, REC of 78.81%, and F1 of 78.81. For the Cleveland dataset, the proposed model had the highest ACC of 97.17%, PRE of 97.42%, REC of 97.17%, and F1 of 97.15%. In addition, the proposed model achieved better results than the literature. As a result, the proposed model can improve the

disease prediction and can improve the quality of life of the heart disease patients. In the future, we will test the performance of the proposed model with other datasets. We will extend the model by adding other modalities such as images and EEG data. We will provide interpretability features to the proposed model.

**REFERENCES:**

Cardiovascular Diseases (CVDs). Available online: <http://www.who.int/cardiovascular_diseases/en/> (accessed on 10 October 2022).

Hall, J.E.; Hall, M.E. Guyton and Hall Textbook of Medical Physiology e-Book; Elsevier Health Sciences: Amsterdam, The Netherlands, 2020.

Bhowmick, A.; Mahato, K.D.; Azad, C.; Kumar, U. Heart Disease Prediction Using Different Machine Learning Algorithms. In Proceedings of the 2022 IEEE World Conference on Applied Intelligence and Computing (AIC), Sonbhadra, India, 17–19 June 2022; pp. 60–65.

Saleh, H.; Alyami, H.; Alosaimi, W. Predicting Breast Cancer Based on Optimized Deep Learning Approach. Comput. Intell. Neurosci. 2022, 2022, 1820777. [CrossRef] [PubMed]

[Acquiring the user’s opinion by using a generalized context-aware recommender system for real-world applications](https://scholar.google.com/citations?view_op=view_citation&hl=en&user=7S2iReEAAAAJ&citation_for_view=7S2iReEAAAAJ%3A9yKSN-GCB0IC) CVM Krishna, DGA Rao

Int J Eng Technol 7 (2.7), 883-886

[Analysing the impact of contextual segments on the overall rating in multi-criteria recommender systems](https://scholar.google.com/citations?view_op=view_citation&hl=en&user=7S2iReEAAAAJ&citation_for_view=7S2iReEAAAAJ%3AY0pCki6q_DkC) CVM Krishna, GA Rao, S Anuradha

Journal of big Data 10 (1), 16

Impact of Contextual Segments in the Prediction of Overall User Gratification in Asian and European Continental Hotel Tourism Sector

Chinta Venkata Murali Krishna, G Appa Rao, Bala Brahmeswara Kadaru, S AnuRadha, 2022/5/16,Book ICCCE 2021: Proceedings of the 4th International Conference on Communications and

Cyber Physical Engineering

[Prediction of User Overall Gratification in Indian Tourism Domain on Hotel Classes and Trip Types](https://scholar.google.com/citations?view_op=view_citation&hl=en&user=7S2iReEAAAAJ&citation_for_view=7S2iReEAAAAJ%3AUeHWp8X0CEIC) CVM Krishna, GA Rao, S AnuRadha, KVD Sagar

ECS Transactions 107 (1), 19813

