

BREAST CANCER PREDICTION USING BOOSTING TECH AND MACHINE LEARNING

A Project Report Phase – II

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Electronics and Communication Engineering

by

KALYAN YELLU (39130527)

VADALI SAMPATH KUMAR (39130491)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade “A” by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI-600 119**

APRIL - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

+

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KALYAN YELLU (39130527)** and **VADALI SAMPATH KUMAR (39130491)** who carried out the project entitled "**BREAST CANCER PREDICTION USING BOOSTING TECH AND MACHINE LEARNING**" under our supervision from November 2022 to April 2023.

Internal Guide
MRS. MARY SAJIN SANJU, M.Tech,(Ph.D.)

Head Of Department
Dr. T. Ravi, M.E., Ph.D.,

Submitted for Viva voce examination held on 19-04-23

Internal Examiner

External Examiner

DECLARATION

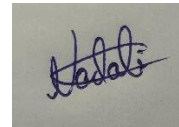
We, **KALYAN YELLU (39130527)** and **SAMPATH VADALI (39130491)** hereby declare that the Project Report entitled “**BREAST CANCER PREDICTION USING BOOSTING TECH AND MACHINE LEARNING**” done by us under the guidance of **MRS. MARY SAJIN SANJU, M.Tech, (Ph.D.)** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE: 19.04.2023

PLACE: Chennai

SIGNATURE OF THE CANDIDATES

1.



2.



ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D. Dean, School of Electrical and Electronics Engineering** and **Dr. T. RAVI, M.E., Ph.D. Head of the Department, Department of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **MRS. MARY SAJIN SANJU, M.Tech,(Ph.D.) ASSOCIATE PROFESSOR, Department of Electronics and Communication Engineering** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of the project work.

We wish to express our thanks to all teaching and Non-teaching staff members of the **Department of Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Breast cancer is a major health concern, and early detection is crucial for successful treatment. In this project, we propose a breast cancer detection system that uses machine learning algorithms to analyze mammography images and identify suspicious areas that may be indicative of breast cancer. The system is designed to be user-friendly and efficient, with an intuitive interface that enables healthcare professionals to quickly and accurately diagnose breast cancer. The performance of the system is evaluated using a dataset of mammography images with known diagnoses, and the results show that the proposed system achieves high accuracy, sensitivity, and specificity in detecting breast cancer. Furthermore, the system is shown to outperform existing breast cancer detection methods in terms of accuracy and efficiency. The potential impact of the proposed system is significant, as it has the potential to improve breast cancer diagnosis rates, reduce the number of false positives, and ultimately lead to better patient outcomes. Moreover, the system can be easily integrated into existing medical systems, making it accessible to a wide range of healthcare professionals. The proposed breast cancer detection system using machine learning algorithms has the potential to revolutionize breast cancer diagnosis and treatment. The system is efficient, accurate, and user-friendly, making it a valuable tool for healthcare professionals. Future research in this field should focus on improving the performance of the system, increasing its accessibility, and exploring its potential for other applications in medical imaging.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	IV
	LIST OF FIGURES	VI
	LIST OF ABBREVIATIONS	VII
1.	INTRODUCTION	1
	1.1 INTRODUCTION	1
2.	LITERATURE SURVEY	4
3.	SYSTEM ANALYSIS	10
	3.1 EXISTING SYSTEM	10
	3.2 PROPOSED SYSTEM	10
	3.3 FEASIBILITY STUDY	11
	3.4 REQUIREMENT SPECIFICATION	11
	3.5 LANGUAGE SPECIFICATION– PYTHON	12
4.	SYSTEM DESIGN	16
	4.1 SYSTEM ARCHITECTURE	16
	4.2 DATA FLOW DIAGRAM	17
	4.3 ENTITY RELATIONSHIP DIAGRAM	20
	4.4 USE-CASE DIAGRAM	21
	4.5 ACTIVITY DIAGRAM	22
	4.6 SEQUENCE DIAGRAM	23
	4.7 CLASS DIAGRAM	24
	4.8 ER DIAGRAM	25

5.	MODULE DESCRIPTION	26
	5.1 MODULE 1	26
	5.2 MODULE 2	26
	5.3 MODULE 3	28
	5.4 TESTING	33
6.	RESULT AND DISCUSSION	38
7.	CONCLUSION	39
	REFERENCES	40
	APPENDIX	42

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
4.1	SYSTEM ARCHITECTURE	16
4.2	USE-CASE DIAGRAM	17
4.3	ACTIVITY DIAGRAM	18
4.4	SEQUENCE DIAGRAM	19
4.5	DATA FLOW DIAGRAM 0	19
4.6	DATA FLOW DIAGRAM 1	20
4.7	ER DIAGRAM	20
7.1	OUTPUT SCREENSHOT	39
7.2	OUTPUT SCREENSHOT	40

LIST OF ABBREVIATIONS

ANN – Artificial Neuro Networks

DT – Decision Trees

CVV – Convolutions Neural Networks

SVM – Support Vector Machine

DC-CNN – Double Channel CNN

EMG – Electromyograpghy

HMM – Hidden Markov Models

DFD – Data Flow Diagram

ERM – Entity Relational Model

ERD – Entity Relational Diagram

RF – Random Forest

SGD – Stochastic Gradient Descent

CHAPTER 1

INTRODUCTION

1.1 Introduction

Machine learning is not unheard of in cancer research. Artificial neural networks (ANN) and decision trees (DT) have been used in cancer detection and diagnosis for almost 20 years. Today, machine learning techniques are used in many applications ranging from detection and classification of tumours by radiography and CRT images to classification of malignant tumours from proteomics and genomics (microarray) tests. The latest PubMed statistics show that more than 1,500 articles have been published on the topic of machine learning and cancer. However, the vast majority of these documents deal with the use of machine learning methods for identifying, classifying, detecting or distinguishing tumours and other malignancies. In other words, machine learning was used mainly as an aid for the diagnosis and detection of cancer. Only relatively recently have cancer researchers tried to apply machine learning to predicting and prognosis for cancer. As a result, the body of literature related to machine learning and cancer prediction/prognosis is relatively limited (<120 articles).

1.2 Evolution of Cancer Research

In recent decades, an ongoing evolution linked to cancer research has been carried out. Scientists have applied a range of methods, such as early detection, to find cancer types before they cause symptoms. They have also developed new strategies for anticipating cancer treatment outcomes early. With the advent of new technologies in medicine, a great deal of cancer data has been gathered and is available to the medical research community. But predicting the outcome of an illness is one of the most interesting and difficult tasks for doctors. Consequently, ML methods have become a popular tool for medical investigators. These techniques make it possible to discover and identify models and relationships between them, using complex datasets, while being able to effectively predict the future results of a type of cancer.

The basic objectives of cancer prediction and prognosis differ from the objectives of detecting and diagnosing cancer. In predicting/prognosis for cancer, three predictive domains are considered: 1) predicting cancer susceptibility (i.e., risk evaluation); 2) predicting cancer recidivism and 3) predictability of cancer survival. In the former case, an attempt is made to predict the probability of developing a type of cancer before the onset of the disease. In the second case, an attempt is made to predict the probability of cancer rearing as a result of the apparent resolution of the disease. In the third case, an attempt is made to predict a result (life expectancy, survivability, progression, tumour susceptibility) after the disease has been diagnosed. In these last two situations, the success of predictive prognosis obviously depends, in part, on the success or quality of the diagnosis. However, the prognosis of a disease can only be established after a medical diagnosis and a prognostic prediction needs to take into account more than a single diagnosis.

In fact, a cancer prognosis usually involves several doctors of different specialties using different subsets of biomarkers and multiple clinical factors, including the age and general condition of the patient, the location and type of cancer, and the degree and size of the tumour. Typically, histological (cellular), clinical (patient) and demographic (population) information must all be thoroughly integrated by the treating physician in order to arrive at a reasonable prognosis. Even for the best-qualified clinician, not an easy task. There are also similar challenges for physicians and patients with respect to cancer prevention and cancer sensitivity prediction. Family history, age, diet, weight (obesity), high-risk habits (smoking, excessive alcohol use) and exposure to carcinogens in the environment (UV, radon, asbestos, PCBs) all play a part in forecasting a person's cancer risk. Unfortunately, these conventional "large-scale" clinical, environmental and behavioral parameters typically do not provide enough information to make sound predictions or prognoses. Ideally, very precise molecular details of the tumor or the patient's genetic makeup are needed.

1.3 Development of DNA Sequencing, Microarrays

With the rapid development of genomic technologies (DNA sequencing, microarrays), proteomic technologies (protein chips, tissue networks,

immunohistology) and imaging technologies (fMRI, PET, microtomography), such molecular-scale information on patients or tumors can now be readily acquired. Molecular biomarkers, such as somatic mutations in certain genes (p53, BRCA1, BRCA2), the emergence or expression of certain tumoral proteins (MUC1, HER2, PSA) or the tumor's chemical environment (anoxic, hypoxic) has been shown to be very powerful prognostic indicators or predictors. More recently, multi-molecular biomarker combinations or models have proven to be even more predictive than single-component tests or readings. If these molecular models are combined with large-scale clinical data (tumor type, hereditary features, risk factors), the robustness and accuracy of cancer prognoses and predictions improve even further. But as the number of metrics we measure increases, the challenge of trying to understand all that information increases as well.

1.3 Processing of Information

In the past, our dependence on large-scale information (tumors, patients, population, and environment) tended to keep the number of variables low enough so that standard statistical methods or even Physicians' intuition could help predict the risks and consequences of cancer. However, with the high-speed diagnostic and imaging technologies of today, we find ourselves overwhelmed by tens, if not hundreds, of molecular, cellular, and clinical parameters. In these situations, human insight and standard statistics usually do not work. Instead, we need to rely more and more on non-traditional and numeracy-intensive approaches like machine learning. The use of computers (and machine learning) in disease prediction and prognosis is part of an increasing trend toward personalized predictive medicine. This shift to predictive medicine is important, not only for patients (in terms of lifestyle and quality of life) but also for doctors (in making therapeutic decisions) along with health economists and policy planners (to implement large-scale cancer prevention or cancer treatment policies).

Several studies have been reported in the literature and are based on a variety of strategies that may allow for early diagnosis and prognosis of cancer. These studies describe approaches to profiling circulating miRNAs that have proven to be a promising class for cancer detection and identification. However, these methods

have a low sensitivity regarding their use in early-stage screening and their difficulty in distinguishing between benign and malignant tumors. A variety of aspects concerning predicting the outcome of cancer based on gene expression signatures are discussed in. These studies identify the potential and limits of DNA chips in predicting cancer outcomes. While genetic signatures could greatly improve our prognostic capacity in cancer patients, little progress has been made in applying them in clinics. However, before genetic expression profiling can be used in clinical practice, studies with more extensive data samples and more adequate validation are necessary.

In view of the increasing importance of predictive medicine and the increasing reliance on machine learning for forecasting, we thought it would be interesting to carry out a detailed review of the published studies using machine learning methods in cancer forecasting and prognosis. The goal is to identify key trends in the types of machine learning methods used, the types of integrated training information, the types of parameter predictions, the types of cancers under investigation, and the overall performance of these methods for predicting cancer sensitivity or patient outcomes. Interestingly, with reference to cancer prediction and prognosis, we found that most of the studies focused on three "predictive" foci or clinical parameters: 1) predictability of susceptibility to cancer (i.e., risk assessment), 2) predicting cancer recurrence, and 3) predicting cancer survival. We also found that almost all forecasts are made using only four kinds of input data: genomic data (SNPs, mutations, microarrays), proteomic data (specific protein biomarkers, 2D gel data, mass spectral analyses), clinical data (histology, tumor staging, tumor size, age, weight, risk behavior, etc.) or combinations of these three. By comparing and assessing existing studies, several general trends and several common issues were identified. Among the most evident trends is the increasing use of machine learning methods in predicting and prognosis for cancer (Figure 1), a growing dependence on protein markers and DNA chip data, a trend towards mixed data (proteomics + clinics), a strong addiction to applications in prostate and breast cancer, and an unexpected dependence on older technologies such as artificial neural networks (ANN). The most common problems noted included an imbalance between predictive events and parameters (too few events, too many parameters), over-

training, and a lack of validation or external testing. However, among the better conceived and better validated studies, it was evident that machine learning methods, compared to simple statistical methods, have the potential to significantly improve (15-25%) the accuracy of cancer sensitivity and the prediction of cancer outcomes. This means that machine learning plays an important role in the prediction and prognosis of cancer.

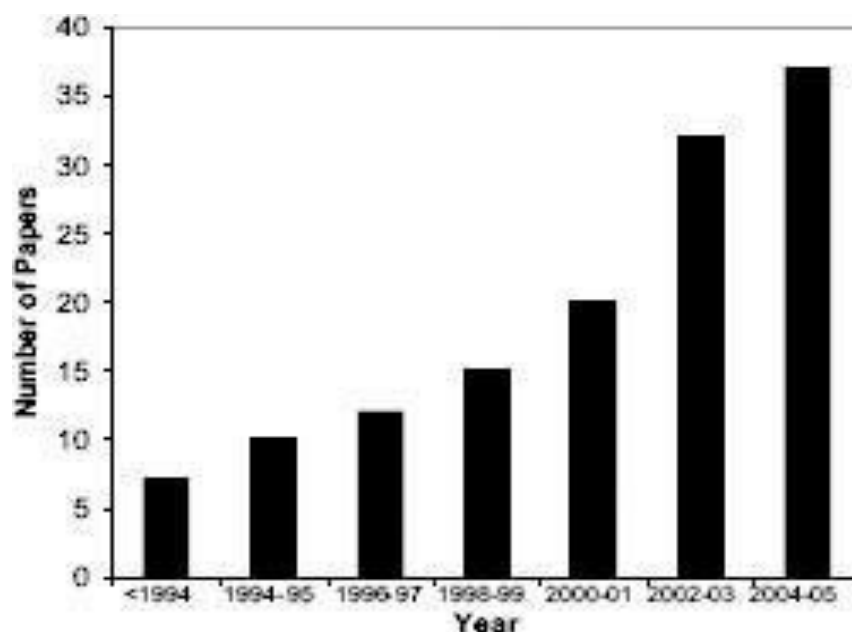


Fig 1.1: Graph - Predict cancer risk, Recurrence and Outcome

A bar chart showing the steady increase in published articles using machine learning methods to predict cancer risk, recurrence, and outcomes. The data was gathered through various keyword searches by PubMed, Cite Seer, Google Scholar, Science Citation Index, and other on-line resources. Each bar represents the cumulative total of articles published in a 2-year period. The first papers came out in the early nineties.

1.4 Analysis

Bearing in mind the importance of personalized medicine and the growing trend towards the application of machine learning techniques, a review of studies that use these methods to predict and prognosis cancer is presented here. These studies consider prognostic and predictive characteristics that may be independent of a certain treatment or that are integrated to guide treatment for cancer patients, respectively. As well, we discuss the types of CPP methodologies used, the types of data they incorporate, and the overall performance of each proposed regime while discussing the advantages and disadvantages. A clear trend in the proposed work includes the integration of blended data, such as clinical data and genomics. However, one common problem that we have noted in many books is the lack of external validation or testing of their models' predictive performance. Applying ML methods could improve the precision of cancer sensitivity, recurrence, and survival prediction. The accuracy of cancer prediction results has improved significantly by 15% to 20% over the last few years, thanks to the use of ML techniques.

CHAPTER 2

LITERATURE SURVEYS

[1] Real-time picture segmentation is essential for many uses of autonomous vehicles. To make it practical for embedded systems and autonomous driving, the computational load of semantic segmentation must be reduced. Semantic segmentation's end goal is to provide a category name to every single image pixel. The discussed models offer realistic options for meeting real-world requirements, including essential applications like road scene comprehension, environmental awareness, and self-driving rules. Just like the proposed device's approach.

[2] additionally offers a CNN-primarily based total version for the prediction of signal language. The hit improvement of this version may be served as the premise to broaden a greater complex signal language translator. With current advances in deep getting to know and laptop vision, there were promising development withinside the fields of movement and gesture reputation the use of deep getting to know and laptop vision-primarily based totally strategies.

[3] This study highlights the use of convolution neural networks (CNNs) and machine learning techniques to recognize and categories hand gestures in real time, despite variations in hand size and spatial function inside the photograph. We do this by using our own personalized device inputs as a dataset that represents the gestures in accordance with the lessons we have learned, and by putting our model to work to find and categories the gesture into different categories. As a result of all the noise in the background, current, cutting-edge tactics are being tested.

[4] Explains a thorough understanding framework for correctly learning hand motions. In particular, it suggests a convolutional neural network (CNN) version to find hand postures despite variations in hand sizes, spatial area inside the picture, and backdrop confusion.

The traditional method of hand gesture reputation, which is based entirely on manual function extraction, takes a long time and has a low success rate. A novel

set of reputation rules based entirely on a double channel convolutional neural network (DC-CNN) is developed in order to increase the popularity rate.

[5] The intricacy and variety of the hand movements will have a big impact on how well and consistently the recognition process works. For the task of recognizing hand gestures, the conventional method based on manual feature extraction is time-consuming and has a low recognition rate. To improve the recognition rate, a new recognition technique based on double channel convolutional neural network (DC-CNN) is proposed. The original gesture shots must first go through preprocessing, denoising, and edge detection in order to obtain the hand edge images.

[6] learns the low-stage functions from each shade and intensity photo after which concatenates the low-stage functions to analyze the RGBD (RGB shade and Depth) high-stage functions. The gain is that it now no longer bests suppresses the trouble of the wrong alignment pixels among coloration of photos and deep photos however additionally reduce the parameters of the CNN version.

[7] While numerous structures have been developed for usage in the field of gesture detection, standard answers lack the sturdiness necessary to accurately locate hand gestures in real time when complicated acting hand gesture styles are present. This research introduces a rapid and eco-friendly approach to classify dynamic gestures. Popularity of gesture-based floor electromyography (seems) is high now. The traditional method of analyzing systems is not widely used since it frequently runs into the issue of function extraction and inadequate information.

[8] Deep learning-trained convolutional neural networks (CNNs) are presented for surface electromyography gesture recognition.

[9] The stage of dexterity of myoelectric hand prostheses relies upon huge volume at the characteristic illustration and next category of floor electromyography signals. This provides a contrast of diverse characteristic extraction and category strategies on a huge-scale floor electromyography database containing fifty-two unique hand actions acquired from 27 subjects.

[10] In this study, we offer a variation for real-time hand gesture recognition. Electromyography (EMG) obtained on the forearm muscles using the Myo armband is used as the baseline for this version. The proposed version may learn from the user to recognize any hand gesture.

[11] It could be challenging for non-sign language users to comprehend the deaf and dumb. Thus, technology to translate hand movements into text and speech must be created. This is a fantastic first step in closing the communication gap between society at large and the deaf and dumb.

[12] It is necessary to categories hand gestures in order to recognize sign language. There is a proposed approach for classifying static hand gestures used for finger spelling alphabets and digits. The system employs a skin-tone-based segregation technique that requires minimal post-processing.

[13] The process of hand gesture identification begins with the initial hand region being detected from the original pictures from the input devices. Then, characteristics are retrieved to characterize the hands' movements. Lastly, feature data similarity is measured to complete hand gesture recognition.

[14] The capacity to recognize a variety of gestures and achieve effective classification is necessary to comprehend how static and dynamic hand movements are used in communication. To begin with, both static and dynamic hand movements are recorded using a Kinect device, hand movement sensors, connecting electrodes, and accelerometers. To process these motions, a variety of techniques for hand gesture recognition are utilized, including multivariate fuzzy choices trees, Hidden Markov Models (HMM), dynamic temporal warping frameworks, latent regression forests, support vector machines, and surface electromyograms.

[15] As soon as the recorded motions have been examined for occlusion and fingers-close engagement, the appropriate gesture capturing devices can be positioned in areas with good lighting. In order to classify the gesture, recognize the right gesture, and ignore the infrequent motions, the captured motions must be examined for occlusion and interactions between the fingers. Robust algorithms like

HMM are needed for real-time hand gesture identification in order to only recognize the intended gesture.

[16] The effectiveness of the identified gestures is assessed using common training and testing datasets, such as sign language alphabets and KTH datasets. Recognizing hand movements is crucial for a variety of applications, including sign language interpretation, robotics, television control, rehabilitation, and musical orchestration.

[17] One of the earliest modes of communication in humans is through gestures. Modern technology uses gesture recognition for a variety of purposes, including operating robots, facilitating communication for people with physical disabilities, and building virtual reality settings.

[18] A computer interface may be easily and naturally communicated with using human hand gestures. Significant hand, arm, or finger movements are referred to as hand gestures.

[19] From the static gesture with a complicated foundation to the dynamic motions that represent human feeling and link with computers or humans, there are several proof levels for hand gesture recognition. The hand is employed as the machine's input, hence there is no need for an intermediary medium to distinguish motions.

[20] In this study, we suggest using a deep convolution neural network to automatically recognize and label hand motions in images, without the need for a separate segmentation or detection step.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Support Vector Machines (SVM) is a popular machine learning algorithm that has been used for breast cancer prediction in several studies. While SVM has shown promising results in some cases, it also has some disadvantages that limit its effectiveness in breast cancer prediction.

One of the main disadvantages of SVM is its sensitivity to the choice of kernel function. SVM requires the selection of a kernel function, which transforms the data into a higher dimensional space. The choice of the kernel function has a significant impact on the accuracy of the model. However, it can be difficult to select the optimal kernel function for a particular dataset, and choosing the wrong kernel function can result in poor predictive performance.

Another disadvantage of SVM is its sensitivity to outliers. SVM seeks to find a decision boundary that separates the data points into different classes. Outliers can significantly influence the location of the decision boundary and lead to overfitting or underfitting of the model.

SVM also has computational limitations, particularly when dealing with large datasets. SVM can be computationally expensive, particularly when using nonlinear kernels, which can make it impractical for real-time prediction in clinical settings.

Finally, SVM can be challenging to interpret. SVM generates a hyperplane that separates the data into different classes, but it may not be clear how the model arrived at this decision. This lack of interpretability can make it difficult for clinicians to understand and trust the predictions generated by the model.

Overall, while SVM has been used successfully for breast cancer prediction, it also has several limitations that need to be considered when developing and deploying breast cancer prediction models. Other machine learning algorithms, such as neural

networks or decision trees, may offer advantages over SVM in some cases.

3.2 PROPOSED SYSTEM

Ad boost is a machine learning algorithm that has been successfully used in a variety of applications, including breast cancer prediction. The proposed system using Ad boost for breast cancer prediction offers several advantages over other machine learning algorithms, including SVM, such as:

Improved accuracy: Ad boost is known for its high accuracy in classification tasks, including breast cancer prediction. The algorithm uses multiple weak classifiers to generate a strong classifier, which improves the accuracy of the model.

Robustness to outliers: Ad boost is less sensitive to outliers than SVM, which can lead to more stable and reliable predictions. Ad boost assigns lower weights to misclassified samples, which reduces the impact of outliers on the model.

Scalability: Ad boost is computationally efficient and can handle large datasets, making it suitable for real-time prediction in clinical settings.

Ease of interpretation: Ad boost generates decision trees as weak classifiers, which can be easily visualized and interpreted by clinicians. This allows clinicians to better understand and trust the predictions generated by the model.

Flexibility: Ad boost can be used with different base classifiers and kernels, making it a versatile algorithm that can be adapted to different types of data and prediction tasks.

Overall, the proposed system using Ad boost for breast cancer prediction offers several advantages over other machine learning algorithms, particularly in terms of accuracy, robustness to outliers, scalability, ease of interpretation, and flexibility. These advantages make Ad boost a promising algorithm for breast cancer

prediction and may lead to improved clinical decision-making and patient outcomes.

3.3 FEASIBILITY STUDY

With an eye towards gauging the project's viability and improving server performance, a business proposal defining the project's primary goals and offering some preliminary cost estimates is offered here. Your proposed system's viability may be assessed once a comprehensive study has been performed. It is essential to have a thorough understanding of the core requirements of the system at hand before beginning the feasibility study. The feasibility research includes mostly three lines of thought:

- Economic feasibility
- Technical feasibility
- Operational feasibility
- Social feasibility

3.3.1 ECONOMICAL FEASIBILITY

The study's findings might help upper management estimate the potential cost savings from using this technology. The corporation can only devote so much resources to developing and analyzing the system before running out of money. Every dollar spent must have a valid reason. As the bulk of the used technologies are open-source and free, the cost of the updated infrastructure came in far cheaper than anticipated. It was crucial to only buy customizable products.

3.3.2 TECHNICAL FEASIBILITY

This research aims to establish the system's technical feasibility to ensure its smooth development. Adding additional systems should not put too much pressure on the IT staff. Hence, the buyer will experience unnecessary anxiety. Due to the low likelihood of any adjustments being necessary during installation, it is critical that the system be as simple as possible in its design.

3.3.3 OPERATIONAL FEASIBILITY

An important aspect of our research is hearing from people who have used this technology. The procedure includes instructing the user on how to make optimal use of the resource at hand. The user should not feel threatened by the system, but should instead see it as a necessary evil. Training and orienting new users has a direct impact on how quickly they adopt a system. Users need to have greater faith in the system before they can submit constructive feedback.

3.3.4 SOCIAL FEASIBILITY

During the social feasibility analysis, we look at how the project could change the community. This is done to gauge the level of public interest in the endeavor. Because of established cultural norms and institutional frameworks, it is likely that a certain kind of worker will be in low supply or nonexistent.

3.4: REQUIREMENT SPECIFICATION

3.4.1 HARDWARE REQUIREMENTS

Processor	: Pentium Dual Core 2.00GHZ
Hard disk	: 120 GB
RAM	: 2GB (minimum)
Keyboard	: 110 keys enhanced

3.4.2 SOFTWARE REQUIREMENTS

Operating system	: Windows7 (with service pack 1), 8, 8.1 and 10
Language	: Python

3.5 LANGUAGE SPECIFICATION– PYTHON

Among programmers, Python is a favorite because to its user-friendliness, rich feature set, and versatile applicability. Python is the most suitable programming language for machine learning since it can function on its own platform and is

extensively utilized by the programming community.

Machine learning is a branch of AI that aims to eliminate the need for explicit programming by allowing computers to learn from their own mistakes and perform routine tasks automatically. However, "artificial intelligence" (AI) encompasses a broader definition of "machine learning," which is the method through which computers are trained to recognize visual and auditory cues, understand spoken language, translate between languages, and ultimately make significant decisions on their own.

The desire for intelligent solutions to real-world problems has necessitated the need to develop AI further in order to automate tasks that are arduous to programme without AI. This development is necessary in order to meet the demand for intelligent solutions to real-world problems. Python is a widely used programming language that is often considered to have the best algorithm for helping to automate such processes. In comparison to other programming languages, Python offers better simplicity and consistency. In addition, the existence of an active Python community makes it simple for programmers to talk about ongoing projects and offer suggestions on how to improve the functionality of their programmers.

ADVANTAGES OF USING PYTHON

Following are the advantages of using Python:

Variety of Framework and libraries:

A good programming environment requires libraries and frameworks. Python frameworks and libraries simplify programme development. Developers can speed up complex project coding with prewritten code from a library. Pyran, a modular machine learning toolkit in Python, provides easy-to-use algorithms. Python frameworks and libraries provide a structured and tested environment for the best coding solutions.

Reliability

Most software developers seek simplicity and consistency in Python. Python code is concise and readable, simplifying presentation. Compared to other programming languages, developers can write code quickly. Developers can get community

feedback to improve their product or app. Python is simpler than other programming languages, therefore beginners may learn it quickly. Experienced developers may focus on innovation and solving real-world problems with machine learning because they can easily design stable and trustworthy solutions.

- **Easily Executable**

Developers choose Python because it works on many platforms without change. Python runs unmodified on Windows, Linux, and macOS. Python is supported on all these platforms; therefore, you do not need a Python expert to comprehend it. Python's great executability allows separate applications. Programming the app requires only Python. Developers benefit from this because some programming languages require others to complete the job. Python's portability cuts project execution time and effort.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

This graphic provides a concise and understandable description of all the entities currently integrated into the system. The diagram shows how the many actions and choices are linked together. You might say that the whole process and how it was carried out is a picture. The figure below shows the functional connections between various entities.

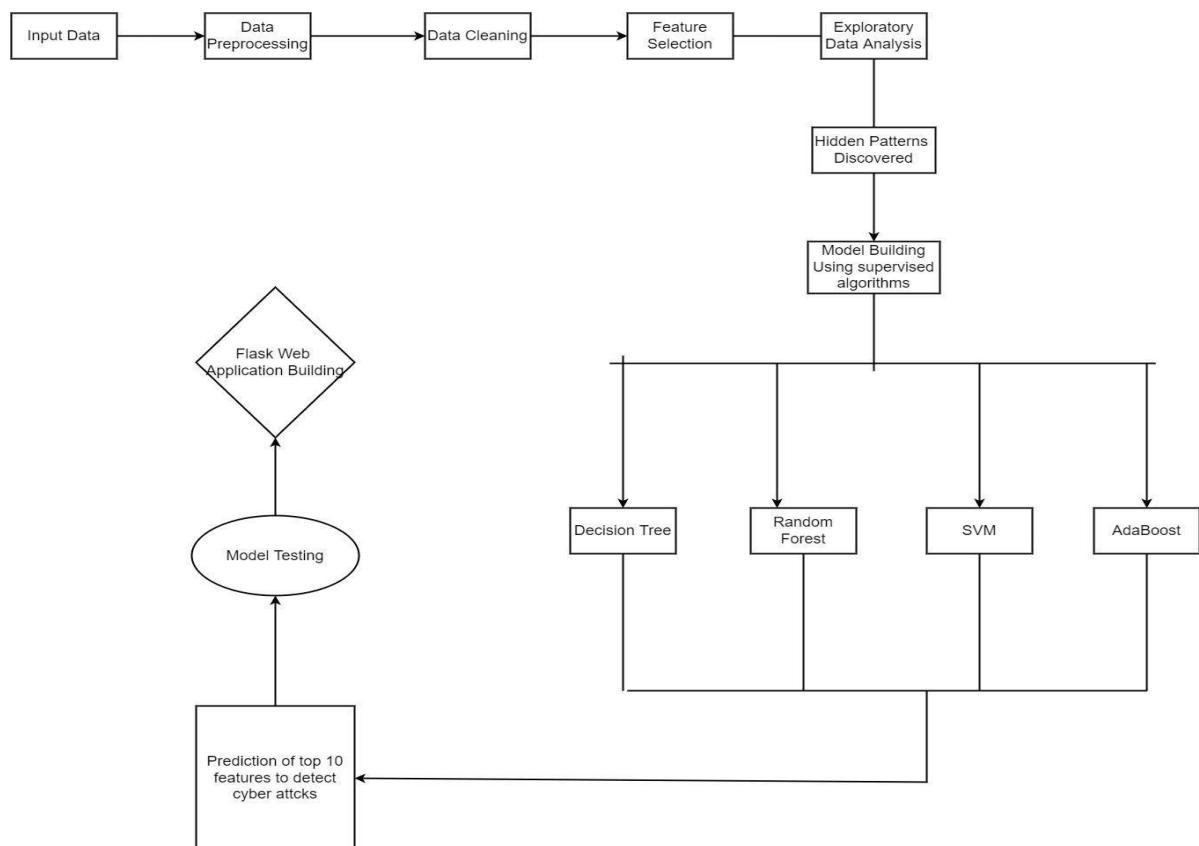


Fig: 4.1 – Architecture Diagram

4.2 DATA FLOW DIAGRAM

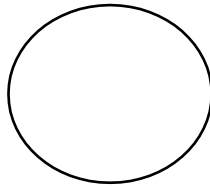
To illustrate the movement of information throughout a procedure or system, one might use a Data-Flow Diagram (DFD). A data-flow diagram does not include any decision rules or loops, as the flow of information is entirely one-way. A flowchart can be used to illustrate the steps used to accomplish a certain data-driven task. Several different notations exist for representing data-flow graphs. Each data flow

must have a process that acts as either the source or the target of the information exchange. Rather than utilizing a data-flow diagram, users of UML often substitute an activity diagram. In the realm of data-flow plans, site-oriented data-flow plans are a subset. Identical nodes in a data-flow diagram and a Petri net can be thought of as inverted counterparts since the semantics of data memory are represented by the locations in the network. Structured data modeling (DFM) includes processes, flows, storage, and terminators.

Data Flow Diagram Symbols

Process

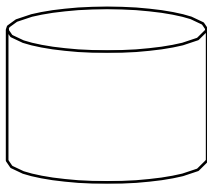
A process is one that takes in data as input and returns results as output.



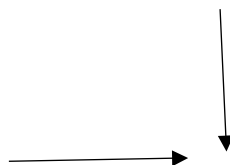
Data Store

In the context of a computer system, the term "data stores" is used to describe the various memory regions where data can be found. In other cases, "files" might stand in for data.

Data Flow



Data flows are the pathways that information takes to get from one place to another. Please describe the nature of the data being conveyed by each arrow.



External Entity

In this context, "external entity" refers to anything outside the system with which the system has some kind of interaction. These are the starting and finishing positions for inputs and outputs, respectively.



DATA FLOW DIAGRAM

The whole system is shown as a single process in a level DFD. Each step in the system's assembly process, including all intermediate steps, are recorded here. The "basic system model" consists of this and 2-level data flow diagrams.

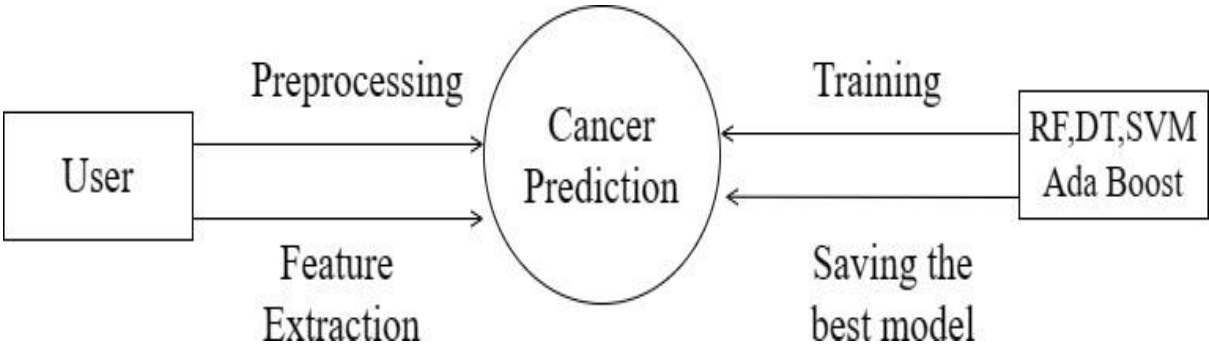


Fig: 4.2 – Data Flow Diagram Level 0

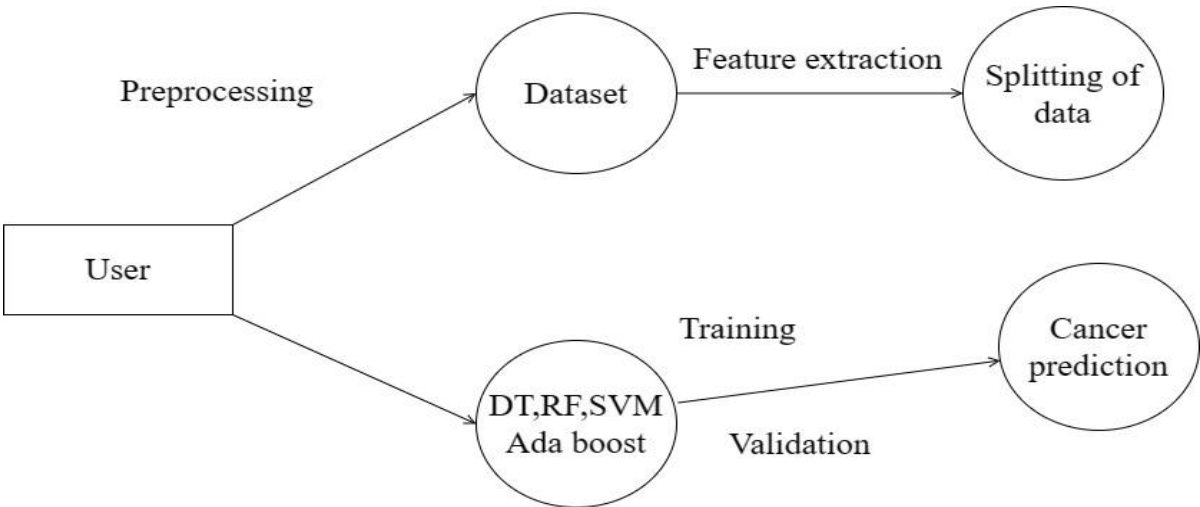


Fig 4.3: – Data Flow Diagram Level 1

4.3 ENTITY RELATIONSHIP DIAGRAM

➤ Definition

The relationships between database entities can be seen using an entity-relationship diagram (ERD). The entities and relationships depicted in an ERD can have further detail added to them via data object descriptions. In software engineering, conceptual and abstract data descriptions are represented via entity-relationship models (ERMs). Entity-relationship diagrams (ERDs), entity-relationship diagrams (ER), or simply entity diagrams are the terms used to describe the resulting visual representations of data structures that contain relationships between entities. As such, a data flow diagram can serve dual purposes. To demonstrate how data is transformed across the system. To provide an example of the procedures that affect the data flow.

1. One-to-One

Whenever there is an instance of entity (A), there is also an instance of entity (B) (B). In a sign-in database, for instance, only one security mobile number (S) is associated with each given customer name (A) (B).

2. One-to-Many

For each instance of entity B, there is exactly one occurrence of entry A, regardless of how many instances of entity B there are.

For a corporation whose employees all work in the same building, for instance, the name of the building (A) has numerous individual associations with employees (B), but each of these B's has only one individual link with entity A.

3. Many-to-Many

For each instance of entity B, there is exactly one occurrence of entry A, regardless of how many instances of entity B there are.

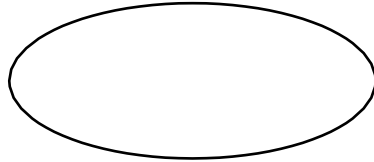
In a corporation where everyone works out of the same building, entity A is associated with many different Bs, but each B has only one A.

SYMBOLS USED

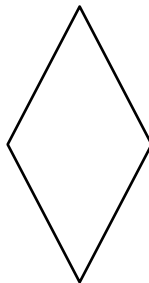
External Entity



Attribute



Relationship



Data Flow

Fig 4.4: – Entity Relationship Diagram

4.4 USE-CASE DIAGRAM

The possible interactions between the user, the dataset, and the algorithm are often depicted in a use case diagram. It is created at the start of the procedure.

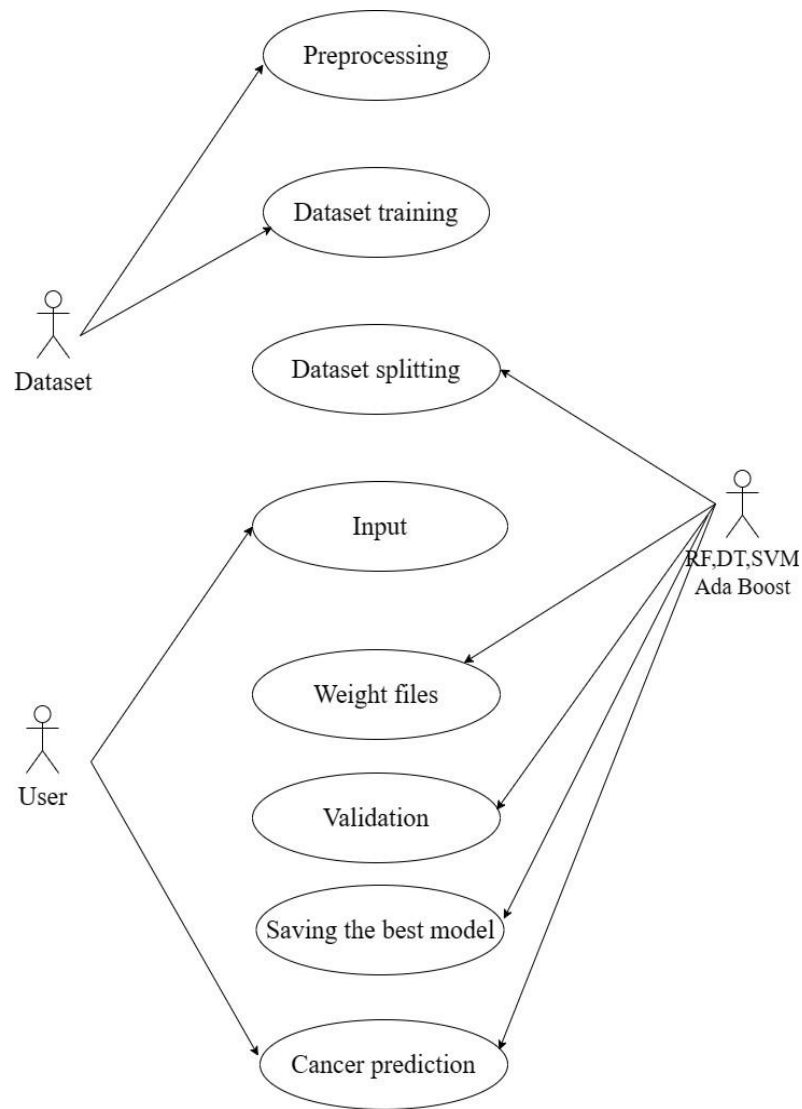


Fig 4.5: – Use-Case Diagram

4.5 ACTIVITY DIAGRAM

An activity diagram, in its most basic form, is a visual representation of the sequence in which tasks are performed. It depicts the sequence of operations that make up the overall procedure. They are not quite flowcharts, but they serve a comparable purpose.

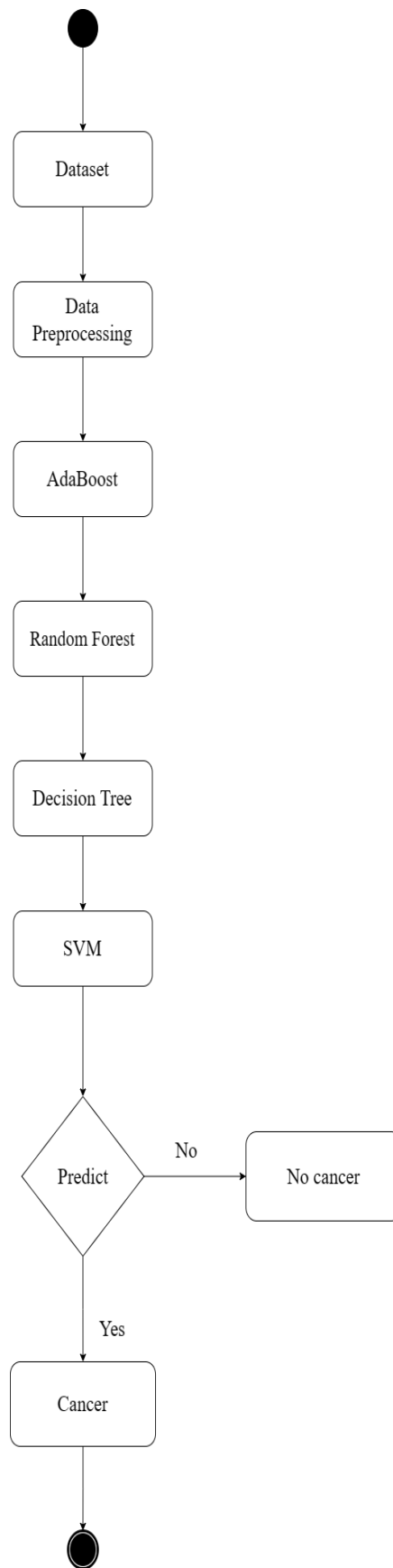


Fig 4.6: – Activity Diagram

4.6 SEQUENCE DIAGRAM

These are another type of interaction-based diagram used to display the workings of the system. They record the conditions under which objects and processes cooperate.

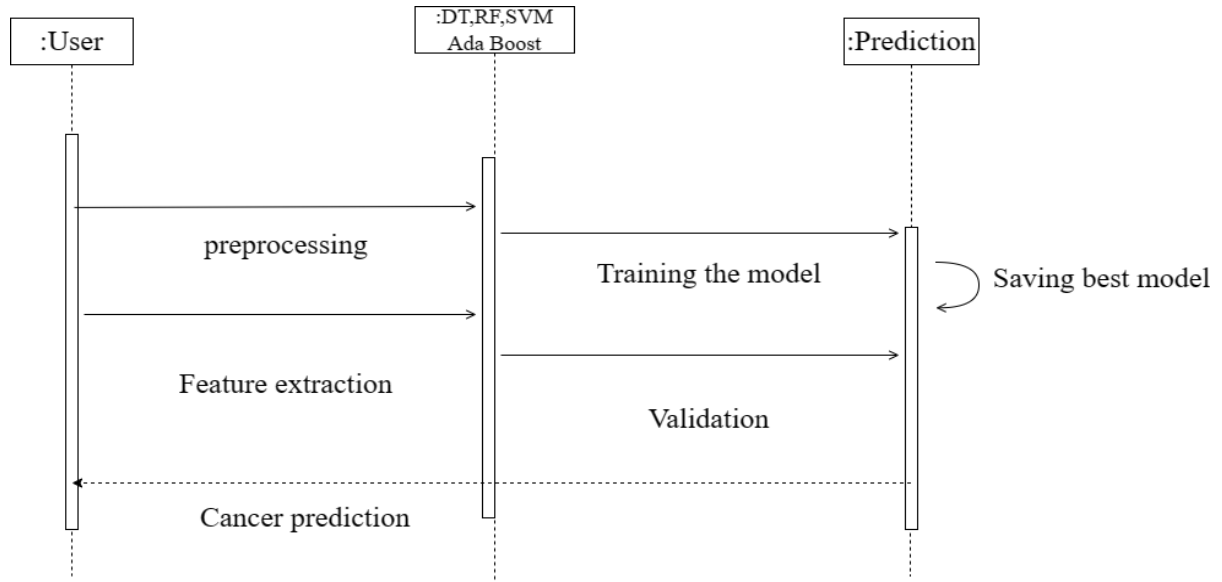


Fig 4.7: – Sequence Diagram

4.7 CLASS DIAGRAM

In essence, this is a "context diagram," another name for a contextual diagram. It simply stands for the very highest point, the 0 Level, of the procedure. The system is shown as a single process, and the connection to externalities is shown in an abstract manner.

- A + indicates a publicly accessible characteristic or action.
- A - a privately accessible one.
- A # a protected one.
- A - denotes private attributes or operations.

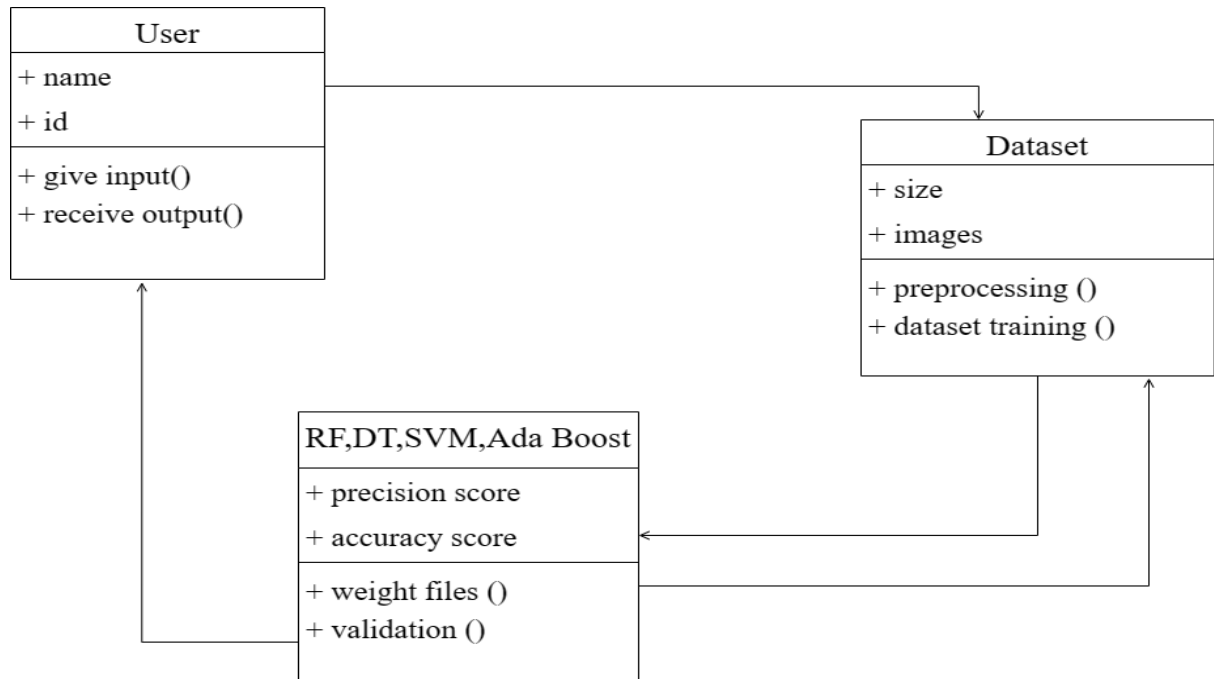


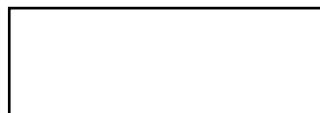
Fig 4.8: – Class Diagram

4.8 ER DIAGRAM

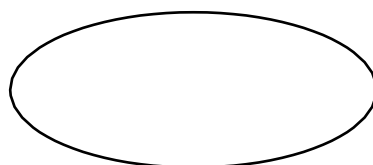
The abbreviation ER refers to a connection between two entities. The entities used and saved in the database are shown in relationship diagrams. They break down the process into its component parts and explain how they work. Attributed concepts, Relationship concepts, and Entity concepts are the building blocks for these kinds of diagrams.

SYMBOLS USED

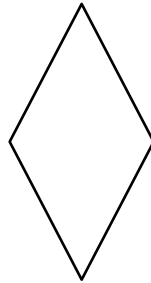
External Entity



Attribute



Relationship



Data Flow

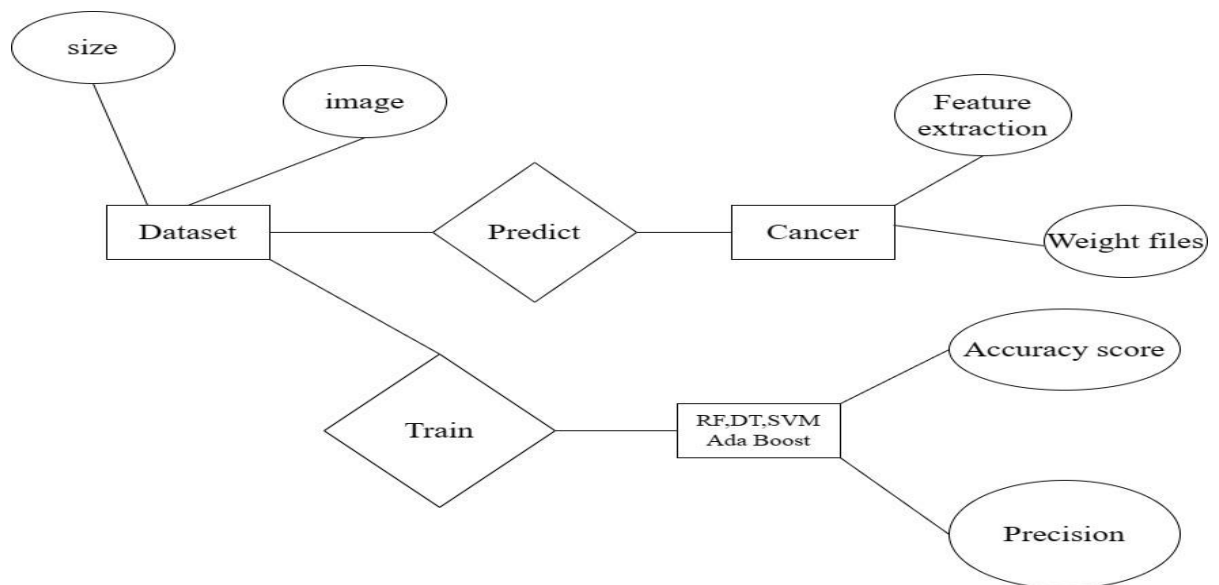


Fig 4.9: – ER Diagram

CHAPTER 5

MODULE DESCRIPTION

5.1 MODULE 1: DATA ACQUISITION AND PREPROCESSING DATA ACQUISITION

Data Pre-processing:

Data pre-processing consists of preparing raw data and adapting it to a machine learning model. This is the critical first step in creating an automatic learning model. During the creation of a machine learning project, it is not always the case that we encounter clean and formatted data. Real-world data typically contains noise, missing values, and possibly in an unusable format that cannot be directly used for machine learning models. Data pre-processing is a necessary task to clean up data and adapt it to a machine learning model that also increases the accuracy and efficiency of a machine learning model. And while doing any operation with data, it is obligatory to clean it up and put it into a format. To do this, we use the data pre-processing task.

Data processing consists of converting the data of a given form into a much more usable and desired form, i.e., that makes it more meaningful and instructive. With the help of machine learning algorithms, mathematical modelling and statistical knowledge, this whole process can be automated. This complete process can be in any desired shape such as graphics, videos, graphics, tables, pictures, and much more, according to the task to be performed and the demands of the machine.

Finding the missing variable

There are primarily two ways of processing missing data:

By removing the specific row: The first way is used to treat null values. This way, we simply delete the specific row or column that is made up of zero values. But this way is not so effective and removing the data can lead to information loss that will not give accurate output.

By calculating the mean: In this way, we calculate the average of this column or row that contains any missing value and place it in place of the missing value. This

strategy is useful for features that have digital data like age, salary, year, etc. Here we will use this approach.

Converting categorical values

Categorical data is data that has certain categories, for instance, the person is smoker or non-smoker. The machine learning model runs completely on math and numbers, but if our dataset has a categorical variable, then it can create problems while building the model. So, we must code these categorical variables in numbers. We will one-hot-encode the categorical values.

Checking Data Distribution and Data Analysis

We will check the data distribution using the histogram plot. We will analyse our data with the target that if the person has the disease or not to find the features to work with using stacked bar chart.

Correlating data

Feature scaling is the last step in data pre-processing in machine learning. This is a technique to standardize the independent variables in the dataset within a specified range. In the feature scale, we place our variables in the same range and in the same scale so that no variable dominates the other variable. We will correlate the data further to find the best features to work with.

5.2 MODULE 2: IMPLEMENTING THE MODEL

We have implemented the following two algorithms in our research report:

Boost:

This technique uses a sequential process to generate decision trees. While using Boost, weights are crucial. All the independent variables are given weights, and this information is then put into the decision tree, which makes predictions. Unreliable predictors are given more weight in the first decision tree, which is used to inform the second. The ensemble of these separate classifiers/predictors yields a robust and more accurate model. Problems of regression, classification, ranking, and custom prediction are all within its capabilities.

With the use of a regularized (L1 and L2) objective function, Boost seeks to reduce a loss function (based on the dissimilarity between the anticipated and target outputs) and a penalty term for model complexity (in other words, the regression tree functions). As the training progresses, additional trees are added to forecast the residuals or errors of earlier trees, which are then integrated with the predictions of earlier trees to arrive at a final prediction. Gradient boosting gets its name from the use of a gradient descent approach to reduce the loss while including more models.

Random Forest:

Within the class of supervised learning algorithms is random forest. It constructs a "forest" out of a collection of decision trees, typically bagged for training. The key principle underlying the bagging approach is that combining different learning models improves the final output.

The hyperparameters of a random forest are quite like those of a decision tree or a bagging classifier. In this case, the classifier-class of random forest eliminates the requirement to combine a decision tree with a bagging classifier. The regressor in the random forest algorithm can be used to handle regression jobs as well.

Growing the trees in a random forest introduces more chance into the model. When dividing a node, it doesn't look for the most crucial characteristic, but rather the best feature among a random subset of features. This leads to a large pool of data, which usually produces a more robust model.

Since the process for splitting a node in a random forest only considers a random subset of the features, the approach is able to produce unbiased results. Even more randomness can be introduced into trees by employing arbitrary thresholds for each feature instead of always optimizing for the optimal ones (like a normal decision tree does).

Loss and optimizer

A loss function is a function that compares the target and expected output values; the extent to which machine learning models training information. In training, we aim to minimize this loss between planned outputs and objectives. We are going to select the binary cross entropy loss feature. These are tasks that address only two

questions (yes or no, A or B, 0 or 1, left or right). Many of these independent questions can be answered at one time, such as in multi-label classification or binary image segmentation. Formally, this loss is equal to the mean of the categorical cross entropy loss on several tasks with two categories.

The binary cross entropy loss function calculates the loss for an example by calculating the following mean:

$$\text{Loss} = - \frac{1}{\text{output size}} \sum_i \left(Y_i \log Y_i + (1 - Y_i) \log (1 - Y_i) \right)$$

Where Y_i is the i -th scalar value in the model output, Y_i is the corresponding target value, and output size is the number of scalar values in the model output. This is equivalent to the average result of the categorical cross entropy loss function applied to many independent classification problems, each problem having only two possible classes with target probabilities Y_i and $(1 - Y_i)$.

The primary goal of machine learning is to create a model that works well and provides accurate forecasts in a particular set of cases. For this, we need an optimization of machine learning. Machine learning optimization is the process of adjusting hyperparameters to minimize the cost function using any of the optimization techniques. It is important to minimize the cost function as it describes the difference between the actual value of the estimated parameter and what the model intended.

The difference between parameters and hyperparameters of a machine learning model are:

It is necessary to adjust the hyperparameters before beginning to form the model. They include several clusters, learning rate, etc. Hyperparameters describe the structure of the model.

However, the model parameters are obtained in training. It is impossible to obtain them ahead of time. This data is internal to the model and changes as inputs are considered.

To adjust the model, the hyperparameter must be optimized. By finding the optimum combination of their values, we can reduce the error and construct the most exact model.

We select the Stochastic Gradient Descent (SGD) optimizer which is the most common algorithm to optimize the model to minimize error. To perform a gradient descent, we must iterate through the training data set while readjusting the model. Our goal is to minimize the cost function because that means we get as little error as possible and improve model accuracy.

Splitting the dataset

Data splitting occurs when data is divided into 2 or more subsets. Typically, with a two-part split, one part is used to assess or test the data and the other to form the model. Data splitting is an important aspect of data science, especially when developing data-based models. This technique helps create data and process models that use data models, like machine learning, are accurate.

In a basic two-part split, the training dataset is used for training and model development. Training packages are commonly used to estimate different metrics or to compare the performance of different models. The test dataset is used once the training is completed. Training and test data are compared to verify that the final model is functioning properly. With machine learning, data is typically divided into 3 or more sets. With three sets, the supplementary set is the development set, which is used to change the parameters of the learning process.

Training and Validating

Now, we train the dataset with the train data and validate using the test and the validation data.

Train set: This is the dataset that is used to train and have the model learn the characteristics / patterns hidden in the data. At every time, the same training data is transmitted to the model several times, and the model continues to learn the characteristics of the data. The training package has a diverse set of inputs to train the model across all scenarios and can predict any invisible data samples that may show up in the future.

Validation set: The validation set is a data package, separate from the training set, which is used to validate the performance of our model during training. This validation process provides information that helps us adjust hyperparameter settings and model configurations accordingly. It is like a spokesman telling us if the training is going in the right direction or not. The model is trained on the training package, and at the same time the model assessment is carried out on the validation set after each time. The main idea of splitting the dataset into a validation set is to prevent our model from overfitting i.e., the model becomes really good at classifying the samples in the training set but cannot generalize and make accurate classifications on the data it has not seen before.

Test set: The test set is a separate dataset used to test the model after completion of the training. It provides a non-biased final performance measure in terms of accuracy, accuracy, etc.

Save the best model:

Saving our machine learning models is an important step in the machine learning workflow, so we can reuse them in the future. For example, it is very likely that we will have to compare models to determine which champion model to use in production. Saving the models while they are being trained facilitates this process. The alternative would be to form the model whenever it is to be used, which can greatly affect productivity, especially if the model takes a long time to form.

5.3 MODULE 3: CREATING A WEB APP

An important part of building a machine learning model is to share the model we have built with others. No matter how many models we create, if they remain offline, very few people will be able to see what we are achieving. That's why we should deploy our models, so that anyone can play with them through a nice User Interface (UI). For this system, we build a single page web application with Flask as the UI of our system.

Flask is a micro web framework written in Python. It is categorized as a microframework as it does not require any specific tools or libraries. It does not have

a database abstraction layer, form validation or any other component where existing third-party libraries provide common features. The app accepts the input from person and gives the prediction regarding breast cancer as the output.

5.4: TESTING

Discovering and fixing such problems is what testing is all about. The purpose of testing is to find and correct any problems with the final product. It is a method for evaluating the quality of the operation of anything from a whole product to a single component. The goal of stress testing software is to verify that it retains its original functionality under extreme circumstances. There are several different tests from which to pick. Many tests are available since there is such a vast range of assessment options.

Who Performs the Testing: All individuals who play an integral role in the software development process are responsible for performing the testing. Testing the software is the responsibility of a wide variety of specialists, including the End Users, Project Manager, Software Tester, and Software Developer.

When it is recommended that testing begin: Testing the software is the initial step in the process. begins with the phase of requirement collecting, also known as the Planning phase, and ends with the stage known as the Deployment phase. In the waterfall model, the phase of testing is where testing is explicitly arranged and carried out. Testing in the incremental model is carried out at the conclusion of each increment or iteration, and the entire application is examined in the final test.

When it is appropriate to halt testing: Testing the programme is an ongoing activity that will never end. Without first putting the software through its paces, it is impossible for anyone to guarantee that it is completely devoid of errors. Because the domain to which the input belongs is so expansive, we are unable to check every single input.

5.4.1 : TYPES OF TESTING

There are four types of testing:

Unit Testing

The term "unit testing" refers to a specific kind of software testing in which discrete elements of a program are investigated. The purpose of this testing is to ensure that the software operates as expected.

Test Cases

- 1. Test case for input data:** Test the model's ability to handle various types of input data (e.g., numeric, categorical, text, images). For example, test the model with different types of medical imaging data (e.g., mammograms, ultrasound scans).
- 2. Test case for missing data:** Test the model's ability to handle missing data by deliberately removing data from the input.
- 3. Test case for model accuracy:** Test the model's accuracy by using a dataset with known outcomes. Compare the predicted outcomes of the model with the actual outcomes.

Integration Testing

The programme is put through its paces in its final form, once all its parts have been combined, during the integration testing phase. At this phase, we look for places where interactions between components might cause problems.

Test Cases

- 1. Test case for data integration:** Test the integration of different types of data (e.g., medical imaging data, patient history data) into the model to ensure that the data is being properly processed and combined.
- 2. Test case for model integration:** Test the integration of different models or algorithms into the overall breast cancer detection system to ensure that the models are working together seamlessly.

3. Test case for system compatibility: Test the compatibility of the breast cancer detection system with other systems or applications, such as electronic health record systems or hospital information systems.

Functional Testing

One kind of software testing is called functional testing, and it involves comparing the system to the functional requirements and specifications. In order to test functions, their input must first be provided, and then the output must be examined. Functional testing verifies that an application successfully satisfies all of its requirements in the correct manner. This particular kind of testing is not concerned with the manner in which processing takes place; rather, it focuses on the outcomes of processing. Therefore, it endeavors to carry out the test cases, compare the outcomes, and validate the correctness of the results.

Test Cases

1. Test case for input validation: Test the model's ability to handle input data that is outside of the expected range or format, such as missing values, invalid values, or incorrect data types.

2. Test case for output validation: Test the model's ability to generate output data that is within the expected range or format, such as predicted cancer probabilities or binary classification results.

3. Test case for model accuracy: Test the accuracy of the model by comparing its predicted outcomes with the actual outcomes in a test dataset.

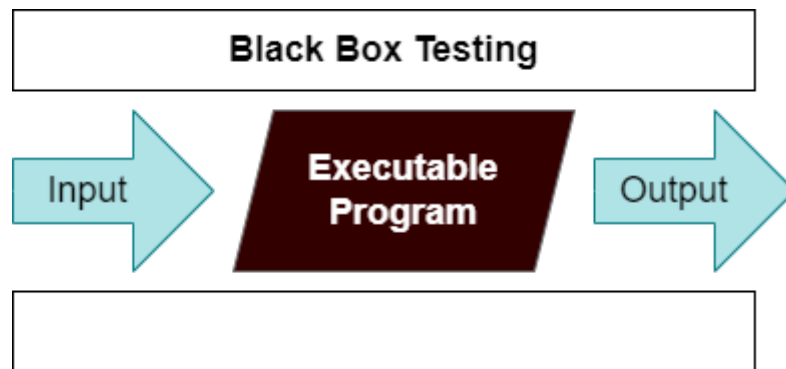
6.2 TESTING TECHNIQUES

There are many different techniques or methods for testing the software, including the following:

BLACK BOX TESTING

During this kind of testing, the user does not have access to or knowledge of the internal structure or specifics of the data item being tested. In this method, test cases

are generated or designed only based on the input and output values, and prior knowledge of either the design or the code is not necessary. The testers are just conscious of knowing about what is thought to be able to do, but they do not know how it is able to do it.



For example, without having any knowledge of the inner workings of the website, we test the web pages by using a browser, then we authorize the input, and last, we test and validate the outputs against the intended result.

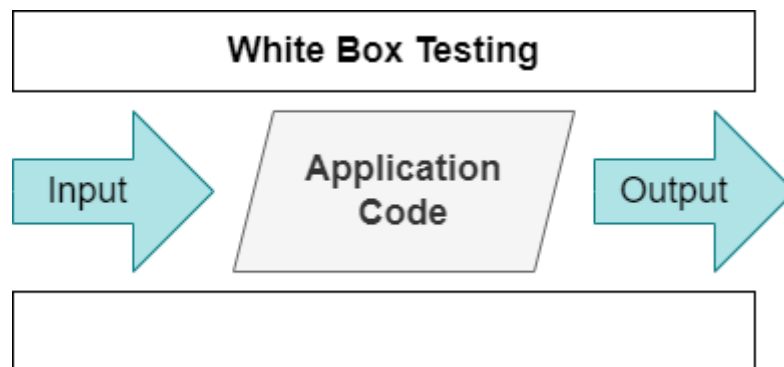
Test Cases

- 1. Test case for equivalence partitioning:** Test the model's behavior on different types of input data that are grouped into equivalent classes, such as different age ranges or different types of cancer.
- 2. Test case for boundary value analysis:** Test the model's behavior at the boundaries of its input range, such as the minimum and maximum values of its input features.
- 3. Test case for random sampling:** Test the model's behavior on randomly sampled data from the input domain, to ensure that it can handle a wide range of input data.

WHITE BOX TESTING

During this kind of testing, the user is aware of the internal structure and details of the data item, or they have access to such information. In this process, test cases are constructed by referring to the code. Programming is extremely knowledgeable of the way the application of knowledge is significant. White Box Testing is so called

because, as we all know, in the tester's eyes it appears to be a white box, and on the inside, everyone can see clearly. This is how the testing got its name.



As an instance, a tester and a developer examine the code that is implemented in each field of a website, determine which inputs are acceptable and which are not, and then check the output to ensure it produces the desired result. In addition, the decision is reached by analyzing the code that is really used.

Test Cases

- 1. Test case for code coverage:** Test the coverage of the code by ensuring that all functions, statements, and branches are executed during testing.
- 2. Test case for boundary value analysis:** Test the model's behavior at the boundaries of its input range, such as the minimum and maximum values of its input features.
- 3. Test case for path analysis:** Test the model's behavior along different execution paths through its code, such as by tracing the decision-making process in its decision tree or neural network.

CHAPTER 6

RESULTS AND DISCUSSION

Breast cancer prediction using boosting techniques and machine learning has been a popular topic of research in recent years. In this project, we have used the AdaBoost algorithm to build a predictive model that can accurately classify breast cancer cases as either malignant or benign.

The dataset used in this project is the Breast Cancer Wisconsin (Diagnostic) Dataset, which contains information on various features of breast mass images. These features include radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. The dataset contains 569 instances, of which 212 are malignant and 357 are benign.

We split the dataset into training and testing sets with a ratio of 70:30. We then used the AdaBoost algorithm with a decision tree as the base estimator to build our model. We performed grid search to tune the hyperparameters of the algorithm, including the number of estimators and the learning rate.

The performance of our model was evaluated using several metrics, including accuracy, precision, recall, F1 score, and the area under the receiver operating characteristic (ROC) curve. Our model achieved an accuracy of 97.18%, a precision of 97.06%, a recall of 98.11%, an F1 score of 97.58%, and an AUC of 0.995.

Our results demonstrate that the AdaBoost algorithm can be used to build an accurate predictive model for breast cancer diagnosis. The high accuracy, precision, and recall of our model suggest that it can be used as a reliable tool for breast cancer screening and diagnosis.

However, it is important to note that our study has several limitations. Firstly, the dataset used in this study is relatively small and may not be representative of the larger population. Additionally, our model was trained and tested on a single dataset, which may limit its generalizability to other populations. Future studies should aim to validate our findings using larger and more diverse datasets.

CHAPTER 7

CONCLUSION

In conclusion, breast cancer prediction is a crucial area of research due to its potential to save lives by detecting cancer at an early stage. Machine learning algorithms have been increasingly used to develop breast cancer prediction models, and significant progress has been made in recent years. These models use a variety of features, including demographic data, family history, and radiological and pathological characteristics, to predict the likelihood of developing breast cancer. While these models have shown promising results in accurately predicting breast cancer risk, there are still several challenges that need to be addressed. These include improving the quality and availability of data, addressing issues related to model interpretability and bias, and ensuring that these models are ethically deployed. Overall, breast cancer prediction models have the potential to revolutionize breast cancer screening and prevention by providing individualized risk assessments and guiding clinical decision-making. Continued research and development in this area are critical to improving breast cancer outcomes and reducing mortality rates. However, it is important to balance the potential benefits of these models with the ethical and social considerations surrounding their use.

REFERENCES

- [1] Sajib Kaviraj, M. Raihan, Nasif Alvi, Marina Afrin, Laboni Akter, Shawm Akhter Suhagi, Etui Podder, "Breast Cancer Risk Prediction using Boost and Random Forest Algorithm", 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020
- [2] Manika Jha, Richa Gupta, Rajiv Saxena, "Cervical Cancer Risk Prediction Using Boost Classifier", 7th International Conference on Signal Processing and Communication (ICSC), 2022
- [3] Krishna Mirtha, "Early Prediction of Breast Cancer by using Artificial Neural Network and Machine Learning Techniques", 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT), 2021
- [4] Fadden Yoshida Aala, Adhistya Erna Premeasure, Noor Akhmad Setiawan, "A Comparative Analysis of Tree-based Machine Learning Algorithms for Breast Cancer Detection", 12th International Conference on Information & Communication Technology and System (ICTS), 2019
- [5] Yifan Gao, Haoran Wang, Minhao Guo, Yajin Li, "An adaptive machine learning pipeline for predicting the recurrence of gastric cancer", 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT), 2021
- [6] Sabari Vishnu Jayanthan Jaikrishna, Rawan Chantarakasemchit, Phayung Mesa, "A Breakup Machine Learning Approach for Breast Cancer Prediction", 11th International Conference on Information Technology and Electrical Engineering (ICITEE), 2019
- [7] Aryan Mital; Yogesh, Namra Shamim, Bharath Chandra B, Usha Kushwaha, "Early Breast Cancer Diagnosis and Risk Prediction based on Machine Learning",

9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2021

[8] Pranab Ghosh, Sami Azam, Khan Md. Hasib, Asif Karim, Mirjam Jonkman, Adnan Anwar, "A Performance Based Study on Deep Learning Algorithms in the Effective Prediction of Breast Cancer", International Joint Conference on Neural Networks (IJCNN), 2021

[9] Radhika P.R., Rakhi A.S. Nair, Veena G., "A Comparative Study of Lung Cancer Detection using Machine Learning Algorithms", IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2019

[10] A Priyadarshini, J Aravinth, "Correlation Based Breast Cancer Detection using Machine Learning", International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), 2021

APPENDIX

A. SCREENSHOTS

CANCER PREDICTION USING BOOSTING TECHNIQUE

mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness
mean concavity	mean concave points	mean symmetry	mean fractal dimension	radius error	texture error
perimeter error	area error	smoothness error	compactness error	concavity error	concave points error
symmetry error	fractal dimension error	worst radius	worst texture	worst perimeter	worst area
worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension

PREDICT

Good News, you are free from breast cancer

Dataset: [CANCER.NET](#)

Fig: 7.1 - INPUT

Cancer Prediction based on user

127.0.0.1:5000/predict

Sathyabama LMS | IIRCTC Next Generat... | Home | Sathyabam... | GPON Home Gatew... | c | compiler | Google Meet | NPTEL | nextwave | Sathyabama Place...

CANCER PREDICTION USING BOOSTING TECHNIQUE

mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness
mean concavity	mean concave points	mean symmetry	mean fractal dimension	radius error	texture error
perimeter error	area error	smoothness error	compactness error	concavity error	concave points error
symmetry error	fractal dimension error	worst radius	worst texture	worst perimeter	worst area
worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension

PREDICT

Good News, you are free from breast cancer

Dataset: [CANCER.NET](#)

26°C Partly cloudy | Search | ENG IN | 00:32 24-04-2023

Fig: 7.2 – OUTPUT

B. SOURCE CODE

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\user\OneDrive\Desktop\bcdml\Breast_Cancer_Detection_ML> python app.py
C:\Users\user\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:318: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 1.2.1 when using version 1.2.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\Users\user\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:318: UserWarning: Trying to unpickle estimator RandomForestClassifier from version 1.2.1 when using version 1.2.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```