Mini Project Title

"BREAST CANCER DETECTION"

Submitted in partial fulfilment of the requirements of the degree

BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

By

Pooja Bhagat **104**

Priyanka Korde **118**

Raghuwardayal Maurya **123**

Rohit Mishra **125**

Supervisor

Prof. Manish Umale

Department of Computer Engineering

Lokmanya Tilak College of Engineering Koparkhairne, NaviMumbai - 400 709

University of Mumbai

(AY 2021-22)

CERTIFICATE

This is to certify that the Mini Project entitled **"Breast Cancer Detection "** is a Bonafide work of **Pooja Bhagat(104), Priyanka Korde(118), Raghuwardayal Maurya(123), Rohit Mishra(125)** submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Computer Engineering".**

Prof. Manish Umale
Guide

Prof. Rajendra Gawali
Head of Department

Dr. Vivek K. Sunnapawar
Principal

Mini Project Approval

This Mini Project entitled "**Breast Cancer Detection"** by **Pooja Bhagat(104), Priyanka Korde(118), Raghuwardayal Maurya(123), Rohit Mishra(125)** is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

Examiners

1……

(Internal Examiner Name & Sign)

2……

(External Examiner name & Sign)

Table of Contents

Chapter	Contents	Pg. No.
1.	Acknowledgements	05
2.	Introduction	06
	2.1) Introduction	06
	2.2) Problem Statement	07
3.	Literature Survey	08
	3.1) Literature Survey	08
	3.1) Survey of Existing System	09
4.	Proposed System	10
	4.1) System Architecture	10
	4.2) Datasets	11
	4.3) Results	12
	4.4) Conclusion	16
5.	References	17

Acknowledgement

I remain immensely obliged to Prof. Manish Umale for providing me with the idea of this topic,and for his/her invaluable support in gathering resources for me either by way of information orcomputer also his/her guidance and supervision which made this project successful.

We would like to thank mini project Coordinators, Dr. S.K. Shinde Vice Principal , Prof. Rajendra Gawli Headof Computer Engineering Department and Dr. Vivek Sunnapwar Principal, LTCoE.

I am also thankful to faculty and staff of Computer Engineering Department and Lokmanya TilakCollege of Engineering, Navi Mumbai for their invaluable support. I would like to say that it hasindeed been a fulfilling experience for working out this project topic

Pooja Bhagat

Priyanka Korde

Raghuwardayal Maurya

Rohit Mishra

INTRODUCTION

2.1 Introduction

Breast cancer (BC) is one of the most common cancers among women worldwide, representing the majority of new cancer cases and cancer-related deaths according to global statistics, making it a significant public health problem in today's society.

The early diagnosis of BC can improve the prognosis and chance of survival significantly, as it can promote timely clinical treatment to patients. Further accurate classification of benign tumors can prevent patients undergoing unnecessary treatments. Thus, the correct diagnosis of BC and classification of patients into malignant or benign groups is the subject of much research. Because of its unique advantages in critical features detection from complex BC datasets, machine learning (ML), deep learning is widely recognized as the methodology of choice in BC pattern classification and forecast modeling.

2.3 Problem Statement

Breast Cancer is one of the leading cancer developed in many countries including India. Though the endurance rate is high – with early diagnosis 97% women can survive for more than 5 years. Statistically, the death toll due to this disease has increased drastically in last few decades. The main issue pertaining to its cure is early recognition. Hence, apart from medicinal solutions some Data Science solution needs to be integrated for resolving the death causing issue.

This analysis aims to observe which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in model selection and hyper parameter selection. The goal is to classify whether the breast cancer is benign or malignant. To achieve this we have used Deep Learning classification methods to fit a function that can predict the discrete class of new input.

LITERATURE SURVEY

3.1 Literature Survey

Twenty-four recent research articles have been reviewed to explore the computational methods to predict breast cancer. The summaries of them are presented below. Chaurasia et al. developed prediction models of benign and malignant breast cancer. Wisconsin breast cancer data set was used. The dataset contained 699 instances, two classes (malignant and benign), and nine integer-valued clinical attributes such as uniformity of cell size. The researchers removed the 16 instances with missing values from the data set to become the data set of 683 instances. The benign were 458 (65.5%) and malignant were 241 (34.5%). The experiment was analyzed by the Waikato Environment for Knowledge Analysis (WEKA). Naive Bayes, RBF Network, and J48 are the three most popular data mining algorithms were used to develop the prediction models. The researchers used 10- fold cross-validation methods to measure the unbiased estimate of the three prediction models for performance comparison purposes. The models' performance evaluation was presented based on the methods‟ effectiveness and accuracy.

Experimental results showed that the Naive Bayes had gained the best performance with a classification accuracy of 97.36%; followed by RBF Network with a classification accuracy of 96.77% and the J48 was the third with a classification accuracy of 93.41%. In addition, the researchers conducted sensitivity analysis and specificity analysis of the three algorithms to gain insight into the relative contribution of the independent variables to predict survival. The sensitivity results indicated that the prognosis factor

3.2 Survey of Existing System

Author & Ref.	Method	Findings	Dataset
Shubham Sharma et al. [16]	Random Forest, KNN and Naïve Bayes.	KNN was a good classifier in terms of accuracy.	Wisconsin Breast Cancer dataset from UCI Repository.
R. Preetha et al. [17]	Data Mining techniques	Detect the hidden cancer associated for classification.	Wisconsin breast cancer dataset.
Majid Nawaz et al. [18]	Deep Learning Convolution neural network	It got 95.4% accuracy when compared with state-of-art models and DenseCNN model used for this.	BreakHis Dataset is used
NareshKhuriwal et al. [19]	Deep learning	It achieved 98% accuracy by using CNN.	Mammogram MIAS database.
Ajay kumar et al. [20]	Classification techniques like SVM, KNN, Naïve Bayes and Decision Tree.	By using BCDW11, it gave 97.13% accuracy and using WBCD32, SVM gave 97.89% accuracy.	BCDW11 and WBCD32 dataset from UCI Repository.
Sri Hari Nallamala et al. [21]	Machine learning techniques	It achieved the 98.50% precision.	Wisconsin Breast Cancer dataset.
R.Chtihrakkannan, P.Kavitha et al. [22]	Machine learning techniques	It achieved 96% accuracy by using DNN.	Mammogram images.
Weal E.Fathy et al. [23]	Deep learning	It achieved 96% area under ROC and 99.8% sensitivity and 82.1% specificity.	Digital Database for Screening Mammography dataset.
Nikita Rane et al. [24]	Machine learning techniques	According to this, enhancement in machine learning gave better results.	Wisconsin Breast Cancer Dataset.
PamuwatMekha et al. [25]	Deep learning	The author compared the machine learning techniques and deep learning. It achieved the 96.99% accuracy with deep learning.	Breast Cancer Wisconsin dataset.
Mahmoud Khademi et al. [26]	Probabilistic Graphical models and deep belief network	It used the graphical model and deep belief network with manifold learning to find out the better accuracy.	Netherlands Cancer Institute dataset, METABRIC breast cancer dataset, Ljubljana breast cancer dataset and WDBC.

PROPOSED SYSTEM

4.1 System Architecture

In this project we will use Data Mining and Machine Learning Algorithms to detect breast cancer, based off of data. Breast Cancer (BC) is a common cancer for women around the world. Early detection of BC can greatly improve prognosis and survival chances by promoting clinical treatment to patients. We will use the UCI Machine Learning Repository for breast cancer dataset. Url:http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29 The dataset used in this story is publicly available and was created by Dr. William H. Wolberg, physician at the University Of Wisconsin Hospital at Madison, Wisconsin, USA. To create the dataset Dr. Wolberg used fluid samples, taken from patients with solid breast masses and an easy-to-use graphical computer program called Xcyt, which is capable of perform the analysis of cytological features based on a digital scan. The program uses a curve-fitting algorithm, to compute ten features from each one of the cells in the sample, than it calculates the mean value, extreme value and standard error of each feature for the image, returning a 30 real-valuated vector .

Attribute Information:

1. ID number 2) Diagnosis (M = malignant, B = benign) 3–32)

Ten real-valued features are computed for each cell nucleus:

2. radius (mean of distances from center to points on the perimeter)

3. texture (standard deviation of gray-scale values)

4. perimeter

5. area

6. smoothness (local variation in radius lengths)

7. compactness ($perimeter^2$ / area — 1.0)

8. concavity (severity of concave portions of the contour)

9. concave points (number of concave portions of the contour)

10. symmetry

11. fractal dimension (“coastline approximation” — 1)

The mean, standard error and “worst” or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

4.2 Datasets

Data Set Information:

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

Separating plane described above was obtained using Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree Construction Via Linear Programming." Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society, pp. 97-101, 1992], a classification method which uses linear programming to construct a decision tree. Relevant features were selected using an exhaustive search in the space of 1-4 features and 1-3 separating planes.

The actual linear program used to obtain the separating plane in the 3-dimensional space is that described in: [K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:
ftp ftp.cs.wisc.edu
cd math-prog/cpo-dataset/machine-learn/WDBC/

Creators:

1. Dr. William H. Wolberg, General Surgery Dept.
University of Wisconsin, Clinical Sciences Center
Madison, WI 53792
wolberg '@' eagle.surgery.wisc.edu

2. W. Nick Street, Computer Sciences Dept.
University of Wisconsin, 1210 West Dayton St., Madison, WI 53706
street '@' cs.wisc.edu 608-262-6619

3. Olvi L. Mangasarian, Computer Sciences Dept.
University of Wisconsin, 1210 West Dayton St., Madison, WI 53706
olvi '@' cs.wisc.edu

Donor:

Nick Stree

4.3 Results

+ Code + Text

Importing the packages

```
import pandas as pd
import numpy as np

from sklearn.datasets import load_breast_cancer
from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt
import seaborn as sns

from keras.models import Sequential
from keras.layers import Dense

import tensorflow as tf
```

Loading the dataset

```
[2] data = load_breast_cancer()
```

Having a look on dataset

```
[3] data.keys()
```

```
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module'])
```

```
[4] print(data['DESCR'])
```

```
.. _breast_cancer_dataset:

Breast cancer wisconsin (diagnostic) dataset
--------------------------------------------

**Data Set Characteristics:**

    :Number of Instances: 569

    :Number of Attributes: 30 numeric, predictive attributes and the class

    :Attribute Information:
        - radius (mean of distances from center to points on the perimeter)
        - texture (standard deviation of gray-scale values)
        - perimeter
        - area
        - smoothness (local variation in radius lengths)
        - compactness (perimeter^2 / area - 1.0)
        - concavity (severity of concave portions of the contour)
```

```
[8] data['data'][-1]

array([7.760e+00, 2.454e+01, 4.792e+01, 1.810e+02, 5.263e-02, 4.362e-02,
       0.000e+00, 0.000e+00, 1.587e-01, 5.884e-02, 3.857e-01, 1.428e+00,
       2.548e+00, 1.915e+01, 7.189e-03, 4.660e-03, 0.000e+00, 0.000e+00,
       2.676e-02, 2.783e-03, 9.456e+00, 3.037e+01, 5.916e+01, 2.686e+02,
       8.996e-02, 6.444e-02, 0.000e+00, 0.000e+00, 2.871e-01, 7.039e-02])

[9] data['target'][::-1]

array([1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0,
       1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1,
       1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0,
       0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,
       0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0,
       1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1,
       1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1,
       0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0,
       1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1,
       1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0,
       1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1,
       0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Converting the arrays to dataframe

Converting the arrays to dataframe

```
pd.DataFrame(data['data'], columns = data['feature_names'])
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419	0.07871	...
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812	0.05667	...
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.2069	0.05999	...
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597	0.09744	...
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	0.1809	0.05883	...
...	...	...	...	...	...	...	...	...	...	...	...
564	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726	0.05623	...
565	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752	0.05533	...
566	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	0.1590	0.05648	...
567	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	0.2397	0.07016	...
568	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	0.1587	0.05884	...

569 rows × 30 columns

```
[11] data['target_names']

array(['malignant', 'benign'], dtype='<U9')
```

worst radius	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension
25.380	17.33	184.60	2019.0	0.16220	0.66560	0.7119	0.2654	0.4601	0.11890
24.990	23.41	158.80	1956.0	0.12380	0.18660	0.2416	0.1860	0.2750	0.08902
23.570	25.53	152.50	1709.0	0.14440	0.42450	0.4504	0.2430	0.3613	0.08758
14.910	26.50	98.87	567.7	0.20980	0.86630	0.6869	0.2575	0.6638	0.17300
22.540	16.67	152.20	1575.0	0.13740	0.20500	0.4000	0.1625	0.2364	0.07678
...	...	...	...	...	...	...	...	...	...
25.450	26.40	166.10	2027.0	0.14100	0.21130	0.4107	0.2216	0.2060	0.07115
23.690	38.25	155.00	1731.0	0.11660	0.19220	0.3215	0.1628	0.2572	0.06637
18.980	34.12	126.70	1124.0	0.11390	0.30940	0.3403	0.1418	0.2218	0.07820
25.740	39.42	184.60	1821.0	0.16500	0.86810	0.9387	0.2650	0.4087	0.12400
9.456	30.37	59.16	268.6	0.08996	0.06444	0.0000	0.0000	0.2871	0.07039

+ Code + Text

Unsupported Cell Type. Double-Click to inspect/edit the content.

Evaluating the model on testing data

```
[21] model.evaluate(x_test , y_test)
```

```
2/2 [==============================] - 0s 5ms/step - loss: 0.0096 - accuracy: 1.0000
[0.009568220935761929, 1.0]
```

```
[22] model.evaluate(x_val, y_val)
```

```
2/2 [==============================] - 0s 7ms/step - loss: 0.0610 - accuracy: 0.9714
[0.06098715960979462, 0.9714285731315613]
```

Got an Accuracy of 0.9706

Display the predicted labels and actual label

```
for i in range(10):
  sample = x_test[i]
  sample = np.reshape(sample, (1,30))

  if (model.predict(sample)[0][0] > 0.5):
    print("-Benign")
```

0s completed at 1:57 PM

Using the model to perdict label for given features

```
# Recreate the exact same model, including its weights and the optimizer
new_model = tf.keras.models.load_model('breast_cancer_model.h5')

# Show the model architecture
new_model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 32)                992

 dense_1 (Dense)             (None, 64)                2112

 dense_2 (Dense)             (None, 128)               8320

 dense_3 (Dense)             (None, 64)                8256

 dense_4 (Dense)             (None, 32)                2080

 dense_5 (Dense)             (None, 16)                528

 dense_6 (Dense)             (None, 8)                 136

 dense_7 (Dense)             (None, 1)                 9

=================================================================
Total params: 22,433
Trainable params: 22,433
Non-trainable params: 0
_________________________________________________________________
```

+ Code + Text

```
=================================================================
Total params: 22,433
Trainable params: 22,433
Non-trainable params: 0
_________________________________________________________________
```

[25]

```
# 0 -> maglignat 1->beginine
features = np.array([[1.288e+01, 2.892e+01, 8.250e+01, 5.143e+02, 8.123e-02, 5.824e-02,
       6.195e-02, 2.343e-02, 1.566e-01, 5.708e-02, 2.116e-01, 1.360e+00,
       1.502e+00, 1.683e+01, 8.412e-03, 2.153e-02, 3.898e-02, 7.620e-03,
       1.695e-02, 2.801e-03, 1.389e+01, 3.574e+01, 8.884e+01, 5.957e+02,
       1.227e-01, 1.620e-01, 2.439e-01, 6.493e-02, 2.372e-01, 7.242e-02]])

prediction = new_model.predict(features)
print(prediction)

if ( prediction > 0.5):
    print("-Benign")
    print('NO fear of Heart Disease But for Better Understanding can Consult your Doctor!')
else:
    print("-Malignant")
    print("There are Chances of Heart Disease! Consult your Doctor Soon!")
```

```
[[0.]]
-Malignant
There are Chances of Heart Disease! Consult your Doctor Soon!
```

```
data['target'][::]
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0,
       1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,
       1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0,
```

0s completed at 1:57 PM

4.4 Conclusion

Breast cancer detection is a challenging problem because it is most popular and harmful disease. Breast cancer is growing every year and there is less chance to recover from this disease. For detection of breast cancer, machine learning and deep learning techniques are used. It is concluded from the previous research, the machine learning techniques give better results in their own field.

The previous research is conducted through many machine learning techniques with some enhancement and augmentation in dataset for the better performance. But it is concluded that machine learning gives better results on linear data. It is also concluded from the previous research, when the data is in the form of images where the machine is failed. To solve the problem of machine learning techniques, an innovative technique is used. Deep learning is recently developed technique that frequently used in data science. For the classification of the breast cancer images data, a deep learning based technique CNN is used. CNN mostly works on the images dataset. In the previous research, it is also concluded that CNN gives better results as compared to machine learning techniques.

REFERENCES

https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0262349#:~:text=Deep%20learning%20is%20one%20of,the%20diagnosis%20of%20breast%20cancer.

https://www.researchgate.net/publication/348604972_A_Review_Paper_on_Breast_Cancer_Detection_Using_Deep_Learning

https://www.researchgate.net/publication/342303246_Breast_Cancer_Detection_and_Prediction_using_Machine_Learning