

**Article I. Cancer Vision: Advanced Breast
Cancer**

Article II. Prediction with Deep Learning

A PROJECT REPORT

Submitted by

AKSHAYASHREE.Y

KAVITHASRI.S

YOGAHARSHINI.S

VAISHNAVI.S

*in partial fulfilment for the award of the degree
of*

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

**DHANALAKSHMI SRINIVASAN COLLEGE OF ENGINEERING AND
TECHNOLOGY, EAST COAST ROAD MAMALLAMPURAM**

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2024

Article III. Ideation Phase

Project Goal: The goal of the project "Cancer Vision: Advanced Breast Cancer Prediction with Deep Learning" is to develop an advanced deep learning model that can accurately predict the presence of advanced breast cancer in medical images. The project aims to leverage the power of artificial intelligence and deep learning algorithms to improve early detection and diagnosis of advanced breast cancer, thereby potentially enhancing patient outcomes and survival rates.

By analyzing medical images such as mammograms, ultrasounds, or MRI scans, the deep learning model will be trained to identify specific patterns, features, and abnormalities associated with advanced breast cancer. The ultimate objective is to create a reliable and efficient tool that can assist healthcare professionals in making accurate diagnoses and treatment decisions, leading to timely interventions and improved patient care.

Target Audience:

- **Healthcare Professionals:** The developed deep learning model aims to assist healthcare professionals, such as radiologists, oncologists, and clinicians, in the early detection and diagnosis of advanced breast cancer. These professionals can benefit from the model's predictions and insights to make informed decisions regarding further testing, treatment planning, and patient management.
- **Breast Cancer Researchers:** Researchers in the field of breast cancer can be interested in the project's outcomes and advancements. The developed deep learning model can contribute to the existing body of knowledge and research on breast cancer detection and prediction, potentially leading to improved diagnostic techniques and patient outcomes.
- **Medical Imaging Specialists:** Professionals specializing in medical imaging, including radiographers and imaging technicians, can find value in the project's developments. The deep learning model can serve as a tool to support

their work by providing automated analysis and interpretation of breast cancer-related images, helping in identifying potential abnormalities or areas of concern.

- **Patients and Patient Advocacy Groups:** Ultimately, the project aims to benefit patients with breast cancer. Accurate and early detection of advanced breast cancer can lead to timely interventions, better treatment planning, and improved patient outcomes. Patients and patient advocacy groups can be interested in the project's progress as it may have a direct impact on their healthcare journey.

Use Cases:

- **Early Detection:** The deep learning model can be used as a screening tool to analyze mammograms or other breast imaging modalities. It can assist in the early detection of advanced breast cancer by identifying suspicious areas or abnormalities that may be indicative of cancer. Early detection can lead to timely intervention and potentially improve patient outcomes.
- **Decision Support for Radiologists:** Radiologists can utilize the deep learning model as a decision support system during the interpretation of breast cancer imaging. The model can provide additional insights and predictions based on its analysis of the images, helping radiologists make more accurate diagnoses and treatment recommendations.
- **Treatment Planning:** The deep learning model's predictions can aid oncologists and clinicians in treatment planning for patients with advanced breast cancer. By analyzing the imaging data, the model can assist in determining the extent and characteristics of the cancer, guiding decisions regarding surgery, chemotherapy, radiation therapy, or other treatment modalities.
- **Follow-up Monitoring:** After treatment, the deep learning model can be utilized for post-treatment monitoring and surveillance. It can analyze follow-up imaging scans to detect any signs of cancer recurrence or metastasis, allowing for early intervention and appropriate adjustments to the patient's care plan.
- **Research and Clinical Trials:** The project's deep learning model can contribute to research efforts and clinical trials related to advanced breast cancer. It can be used to

analyze large datasets of medical images, identifying patterns and trends that can aid researchers in understanding the disease better and developing more effective treatment strategies

Key Features:

- **Deep Learning Model:** The project will involve the development of a deep learning model specifically designed for the prediction of advanced breast cancer. The model can be built using convolutional neural networks (CNNs) or other architectures suitable for image analysis. It will be trained on a large dataset of labeled medical images to learn and identify relevant patterns and features associated with advanced breast cancer.
- **Image Preprocessing:** Prior to feeding the images into the deep learning model, preprocessing techniques can be applied to enhance image quality, remove noise, and standardize the data. This may involve resizing, normalization, denoising, and other techniques to ensure consistency and optimal performance of the model.
- **Multi-Modality Support:** The deep learning model can be designed to support various types of medical images commonly used in breast cancer diagnosis, such as mammograms, ultrasounds, or MRI scans. This multi-modality support can enhance the model's versatility and ability to analyze different image types, enabling comprehensive breast cancer detection.
- **Prediction and Probability Estimation:** The deep learning model can provide predictions indicating the presence or absence of advanced breast cancer in the analyzed images. Additionally, it can estimate the probability or confidence level associated with each prediction, allowing healthcare professionals to gauge the reliability of the model's results.

Potential Challenges:

- **Availability and Quality of Data:** Access to a diverse and representative dataset of labeled medical images of advanced breast cancer can be a challenge. Obtaining a sufficiently large and high-quality dataset that covers different patient populations, imaging modalities, and disease variations is crucial for training an accurate and robust deep learning model.
- **Data Imbalance:** Breast cancer datasets often suffer from class imbalance, where the number of negative cases (no advanced breast cancer) outweighs the positive cases (presence of advanced breast cancer). This imbalance can lead to biased model training and lower predictive performance for the

minority class. Techniques such as data augmentation, oversampling, or using specialized loss functions should be employed to mitigate this issue.

- **Annotation and Expertise:** Accurately annotating medical images for the presence of advanced breast cancer requires expertise and domain knowledge from medical professionals. Ensuring consistent and reliable annotations across the dataset can be challenging and may require collaboration with radiologists or oncologists to validate and refine the annotations.
- **Model Interpretability:** Deep learning models are often considered black-box models, making it challenging to interpret their decisions and understand the underlying factors contributing to predictions. Developing methods to explain and interpret the deep learning model's predictions can enhance trust and confidence among healthcare professionals and facilitate better integration of the model into clinical workflows.
- **Generalization and External Validation:** Deep learning models trained on specific datasets may struggle to generalize well to new and unseen data. It is crucial to validate the model's performance on external datasets or conduct clinical trials to assess its real-world efficacy and generalizability. Robust validation and evaluation protocols should be established to ensure reliable performance assessment.

Define The Problem Statement

Introduction: Breast cancer is a significant global health concern, affecting millions of women worldwide. Early detection and accurate diagnosis are crucial for effective treatment and improved patient outcomes. The project "Cancer Vision: Advanced Breast Cancer Prediction with Deep Learning" aims to leverage the power of artificial intelligence (AI) and deep learning algorithms to develop an advanced tool for the prediction of advanced breast cancer.

Problem Statement: Breast cancer is a significant health concern globally, and the timely detection and diagnosis of advanced breast cancer are crucial for effective treatment and improved patient outcomes. However, current methods of detecting and predicting advanced breast cancer may have limitations in terms of accuracy, efficiency, and scalability. The problem addressed by the project "Cancer Vision: Advanced Breast Cancer Prediction with Deep Learning" is the need for an advanced and reliable tool that utilizes deep learning algorithms to accurately predict the presence of advanced breast cancer in medical images. This tool aims to overcome the shortcomings of existing methods and provide healthcare professionals with enhanced decision support for early detection and diagnosis.

Data Collection:

- **Data Sources:** Identify potential sources of medical images that contain cases of advanced breast cancer. This can include hospitals, medical centers, research institutions, and public databases. Collaborations with healthcare providers and institutions specializing in breast cancer diagnosis and treatment can facilitate access to relevant datasets.
- **Ethical and Legal Considerations:** Ensure compliance with ethical and legal requirements regarding patient data privacy and protection. Obtain necessary approvals from relevant ethics committees, institutional review boards (IRBs), or data governance bodies to access and use patient data for research purposes. Adhere to data anonymization protocols to protect patient privacy

Data Preprocessing:

- **Data Cleaning:** The medical image dataset may contain noise, artifacts, or inconsistencies that can impact the performance of the deep learning model. Data cleaning techniques, such as noise removal, artifact detection, and quality control checks, should be applied to ensure the integrity of the data.
- **Image Resizing and Normalization:** Medical images can have varying resolutions and sizes. Resizing the images to a standardized resolution can facilitate model training and reduce computational requirements. Additionally, normalization techniques, such as intensity normalization, can enhance the comparability and consistency of the images across different patients and imaging modalities.
- **Image Augmentation:** Data augmentation techniques can be applied to artificially increase the diversity and variability of the dataset. This can involve techniques like rotation, flipping, zooming, and adding noise to generate additional training samples. Data augmentation helps the model generalize better and improves its ability to handle variations in patient positioning, image quality, and other factors.

Feature Extraction:

- **Convolutional Neural Networks (CNNs):** CNNs are commonly used in deep learning for image analysis tasks. They consist of multiple layers, including convolutional layers that apply filters to the input images to extract local features. The filters capture different visual patterns, such as edges, textures, or shapes, and hierarchical layers help in learning more complex features.

- **Pretrained Models and Transfer Learning:** Due to the limited availability of large labeled medical image datasets, transfer learning can be employed. Pretrained CNN models, such as VGGNet, ResNet, or Inception, trained on large-scale datasets like ImageNet, can be utilized. These models have learned general features from diverse images and can be fine-tuned or used as feature extractors for the medical image dataset specific to advanced breast cancer

Model Selection:

- **Convolutional Neural Networks (CNNs):** CNNs have proven to be highly effective in image analysis tasks and are commonly used for medical image classification. They can learn complex spatial features and patterns from images, making them well-suited for detecting abnormalities in breast cancer images. CNN architectures such as VGGNet, ResNet, or DenseNet can be considered.
- **Transfer Learning:** Transfer learning involves using pre-trained models trained on large-scale datasets, such as ImageNet, and fine-tuning them on the specific task at hand. This approach can leverage the learned representations and accelerate the training process, especially when the available dataset for breast cancer is limited.
- **Ensemble Models:** Ensemble models combine the predictions of multiple models to improve overall performance. By training multiple models with different architectures or using different training strategies, ensemble models can enhance the robustness and generalization capabilities of the predictions

Training and Evaluation:

- **Dataset Preparation:** A comprehensive dataset of medical images, including mammograms, ultrasounds, or MRI scans, needs to be collected. The dataset should be carefully curated, ensuring a diverse representation of advanced breast cancer cases and including both positive and negative samples.
- **Data Preprocessing:** The collected dataset should undergo preprocessing steps to standardize the data and enhance its quality. This may include resizing, normalization, noise removal, and other techniques to ensure consistency and optimal performance of the model.
- **Model Architecture Selection:** The appropriate deep learning architecture, such as a convolutional neural network (CNN), needs to be selected based on the project requirements. The architecture should be capable of

analyzing medical images and extracting relevant features associated with advanced breast cancer.

- **Model Training:** The selected model is trained using the prepared dataset. This involves feeding the images into the model and iteratively adjusting the model's parameters to minimize the prediction errors. The training process typically employs optimization algorithms like stochastic gradient descent (SGD) or variants to update the model's weights.

Fine-tuning and Optimization: Hyperparameter Tuning: Fine-tuning involves optimizing the hyperparameters of the deep learning model. Hyperparameters include learning rate, batch size, number of layers, kernel size, and activation functions. A systematic search or automated techniques like grid search or Bayesian optimization can be employed to find the optimal set of hyperparameters that maximize the model's performance

Deployment: The deployment of the project "Cancer Vision: Advanced Breast Cancer Prediction with Deep Learning" involves several considerations to ensure successful integration into clinical practice. Here are some important aspects to address during deployment:

- **Regulatory Compliance:** Ensure compliance with relevant regulations and guidelines, such as data privacy and protection regulations (e.g., GDPR), medical device regulations (e.g., FDA approvals), and ethical considerations. It's important to adhere to legal and ethical standards to protect patient privacy, ensure data security, and comply with applicable regulations.
- **Integration with Clinical Workflow:** Integrate the deep learning model seamlessly into the existing clinical workflow to maximize its usability and adoption. This may involve collaborating with healthcare professionals and IT departments to determine the optimal integration points, such as integrating with picture archiving and communication systems (PACS) or radiology information systems (RIS), to streamline the model's usage.
- **User Interface and Visualization:** Develop a user-friendly interface that enables healthcare professionals to interact with the deep learning model effectively. The interface should provide clear and intuitive visualization of model predictions and other relevant information, allowing users to interpret and utilize the results easily.
- **Performance Monitoring and Maintenance:** Implement a system for continuous monitoring of the deep learning model's performance and accuracy over time. Regular updates and maintenance should be conducted to address potential issues, update the model with new data or techniques, and ensure the model remains effective and up-to-date

Empathize & Discover

1. **Healthcare Professionals:** Engage with radiologists, oncologists, and other healthcare professionals involved in breast cancer detection and diagnosis. Understand their current workflow, challenges in identifying advanced breast cancer, and the impact it has on patient outcomes. Gather insights into their expectations, requirements, and how a deep learning solution can augment their decision-making process.
2. **Patients:** Talk to breast cancer patients or patient advocacy groups to understand their experiences, concerns, and needs. Gain insights into their journey, including the diagnostic process, potential delays, and the impact of advanced breast cancer on their lives. Discover their expectations regarding early detection, timely interventions, and the role they envision for AI technologies.
3. **Data Availability:** Collaborate with healthcare institutions and medical imaging centers to identify the challenges they face in collecting and sharing breast cancer-related data. Explore the availability, quality, and privacy concerns associated with obtaining labeled medical images. Empathize with data custodians and understand the potential benefits and risks associated with data sharing.
4. **Regulatory and Ethical Considerations:** Investigate the legal and regulatory landscape surrounding the use of AI in medical imaging and diagnosis. Identify any ethical concerns related to patient privacy, consent, and the responsible use of AI technologies in healthcare. Empathize with the need for transparent and ethical practices in developing and deploying the deep learning solution.
5. **Technical Feasibility:** Assess the existing state of deep learning techniques and their applicability to breast cancer prediction. Understand the computational requirements, model complexity, and infrastructure needed for training and deploying the deep learning model. Discover potential limitations, scalability issues, and any technical challenges that need to be addressed.
6. By empathizing with the stakeholders and understanding their perspectives, challenges, and requirements, you can gain valuable insights to inform the project's direction, objectives, and design decisions

Brainstorm & Prioritize Ideas

1. **Incorporate Transfer Learning:** Explore the use of transfer learning techniques by leveraging pre-trained models, such as ImageNet, to initialize the deep learning model. This approach can accelerate training and potentially improve the model's performance by leveraging knowledge learned from large-scale image classification tasks.
2. **Ensemble of Models:** Build an ensemble of multiple deep learning models with different architectures or hyperparameters to improve prediction accuracy and reduce model variance. This ensemble approach can combine the strengths of individual models and provide more robust predictions.
3. **Active Learning:** Implement an active learning strategy where the deep learning model interacts with human experts to actively request labels for challenging or uncertain samples. This iterative process can help prioritize the annotation effort, improve the model's performance, and reduce the need for a large fully labeled dataset.
4. **Integration with Electronic Health Records (EHR):** Explore the integration of the deep learning model with electronic health records to leverage

additional patient information, such as clinical history, genetic data, or biopsy results. Integrating diverse data sources can enhance the model's predictive capabilities and provide a more comprehensive assessment of advanced breast cancer risk.

5. **Explainable AI (XAI):** Develop methods to provide explanations or visualizations of the deep learning model's predictions, highlighting the regions or features in the medical images that contribute to the prediction. This can help build trust and enhance the interpretability of the model, facilitating its acceptance and adoption in clinical practice.
6. **Robustness to Variations:** Investigate methods to make the deep learning model robust to variations in imaging protocols, image quality, or demographic factors. Adapting the model to handle such variations can improve its generalizability and ensure consistent performance across different healthcare settings and patient populations.
7. Once you have identified the brainstormed ideas, you can prioritize them based on feasibility, potential impact, and alignment with project goals and resources

Project Design Phase – Part 1

Phase 0 — Data Preparation

We will use the UCI Machine Learning Repository for breast cancer [dataset](http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29).

<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:

1. radius (mean of distances from center to points on the perimeter)
2. texture (standard deviation of gray-scale values)
3. perimeter
4. area
5. smoothness (local variation in radius lengths)
6. compactness ($\text{perimeter}^2 / \text{area} - 1.0$)
7. concavity (severity of concave portions of the contour)
8. concave points (number of concave portions of the contour)
9. symmetry
10. 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.

Objectives

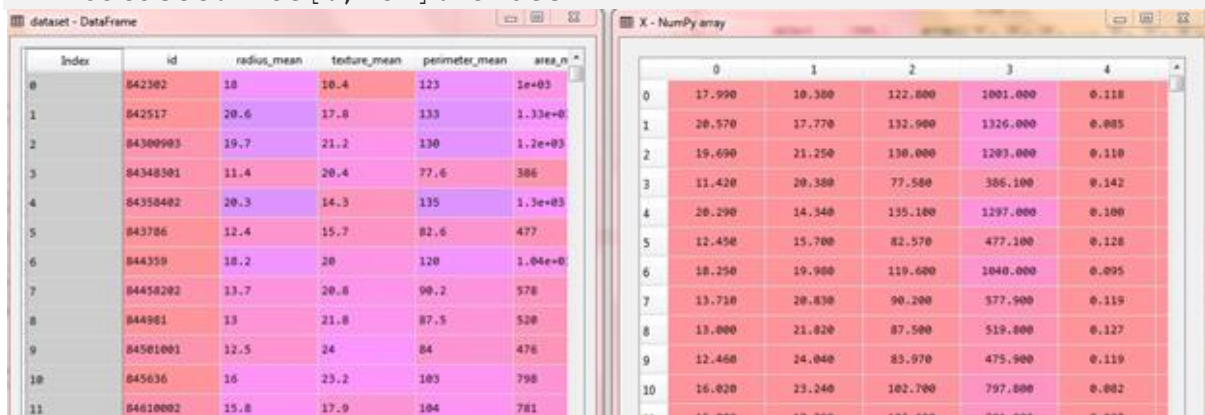
This analysis aims to observe which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in model selection and hyper parameter selection. The goal is to classify whether the breast cancer is benign or malignant.

To achieve this i have used machine learning classification methods to fit a function that can predict the discrete class of new input.

Phase 1 — Data Exploration

We will be using **Spyder** to work on this dataset. We will first go with importing the necessary libraries and import our dataset to Spyder :

```
#importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd#importing our cancer dataset
dataset = pd.read_csv('cancer.csv')
X = dataset.iloc[:, 1:31].values
Y = dataset.iloc[:, 31].values
```



The screenshot shows two windows in the Spyder IDE. The left window, titled 'dataset - DataFrame', displays a table with 6 columns: Index, id, radius_mean, texture_mean, perimeter_mean, and area_n. The right window, titled 'X - NumPy array', displays a table with 5 columns: 0, 1, 2, 3, and 4, representing the first five features of the dataset.

Index	id	radius_mean	texture_mean	perimeter_mean	area_n
0	842302	18	10.4	123	1e+03
1	842517	20.6	17.8	133	1.33e+03
2	84300903	19.7	21.2	130	1.2e+03
3	84348301	11.4	20.4	77.6	386
4	84358402	20.3	14.3	135	1.3e+03
5	843706	12.4	15.7	82.6	477
6	844359	18.2	20	120	1.04e+03
7	84458202	13.7	20.8	90.2	578
8	844901	13	21.8	87.5	520
9	84501001	12.5	24	84	476
10	845636	16	23.2	103	790
11	84610002	15.8	17.9	104	781

	0	1	2	3	4
0	17.990	10.300	122.000	1001.000	0.118
1	20.570	17.770	132.900	1326.000	0.085
2	19.690	21.250	130.000	1203.000	0.110
3	11.420	20.380	77.500	386.100	0.142
4	20.290	14.340	135.100	1297.000	0.100
5	12.450	15.700	82.570	477.100	0.128
6	18.250	19.900	119.600	1040.000	0.095
7	13.730	20.830	90.200	577.900	0.119
8	13.000	21.820	87.500	519.000	0.127
9	12.460	24.040	85.970	475.900	0.119
10	16.020	23.240	102.700	797.800	0.082
11	15.780	17.890	104.000	781.000	0.082

Fig : Dataset and X set after importing the dataset

We can examine the data set using the pandas' **head()** method.

```
dataset.head()
```

	id	radius_mean	...	fractal_dimension_worst	diagnosis
0	842302	17.99	...	0.11890	M
1	842517	20.57	...	0.08902	M
2	84300903	19.69	...	0.08758	M
3	84348301	11.42	...	0.17300	M
4	84358402	20.29	...	0.07678	M

Fig : top 5 data of our dataset

We can find the dimensions of the data set using the panda dataset 'shape' attribute.

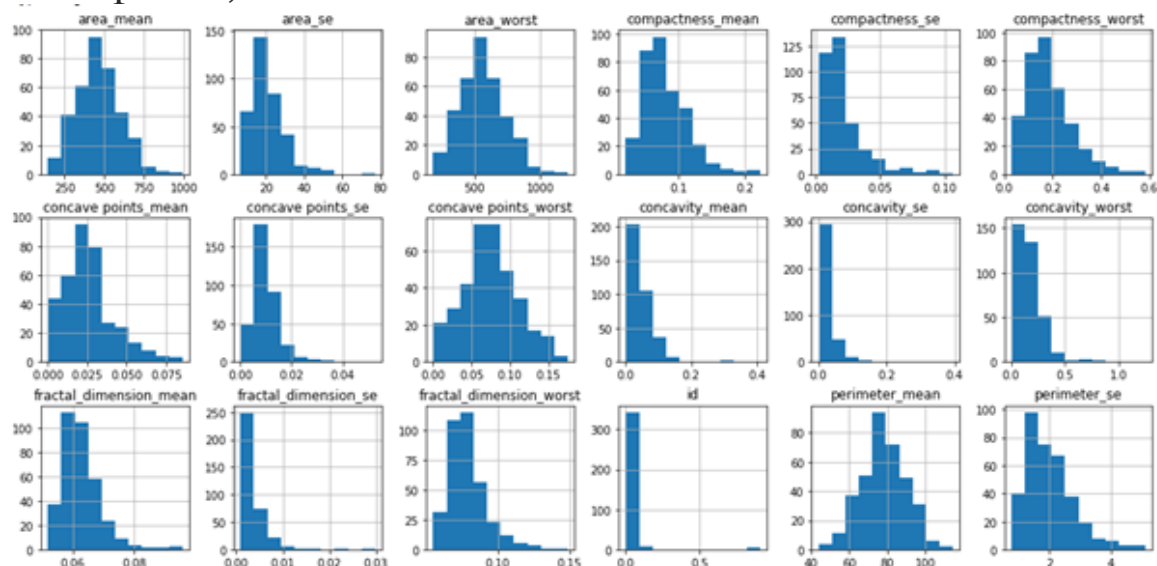
```
print("Cancer data set dimensions :  
{0}".format(dataset.shape))Cancer data set dimensions : (569, 32)
```

We can observe that the data set contain 569 rows and 32 columns. ‘*Diagnosis*’ is the column which we are going to predict , which says if the cancer is M = malignant or B = benign. 1 means the cancer is malignant and 0 means benign. We can identify that out of the 569 persons, 357 are labeled as B (benign) and 212 as M (malignant).

```
diagnosis  
B    357  
M    212  
dtype: int64
```

Visualization of data is an imperative aspect of data science. It helps to understand data and also to explain the data to another person. Python has several interesting visualization libraries such as Matplotlib, Seaborn etc.

In this tutorial we will use pandas’ visualization which is built on top of matplotlib, to find the data distribution of the features.



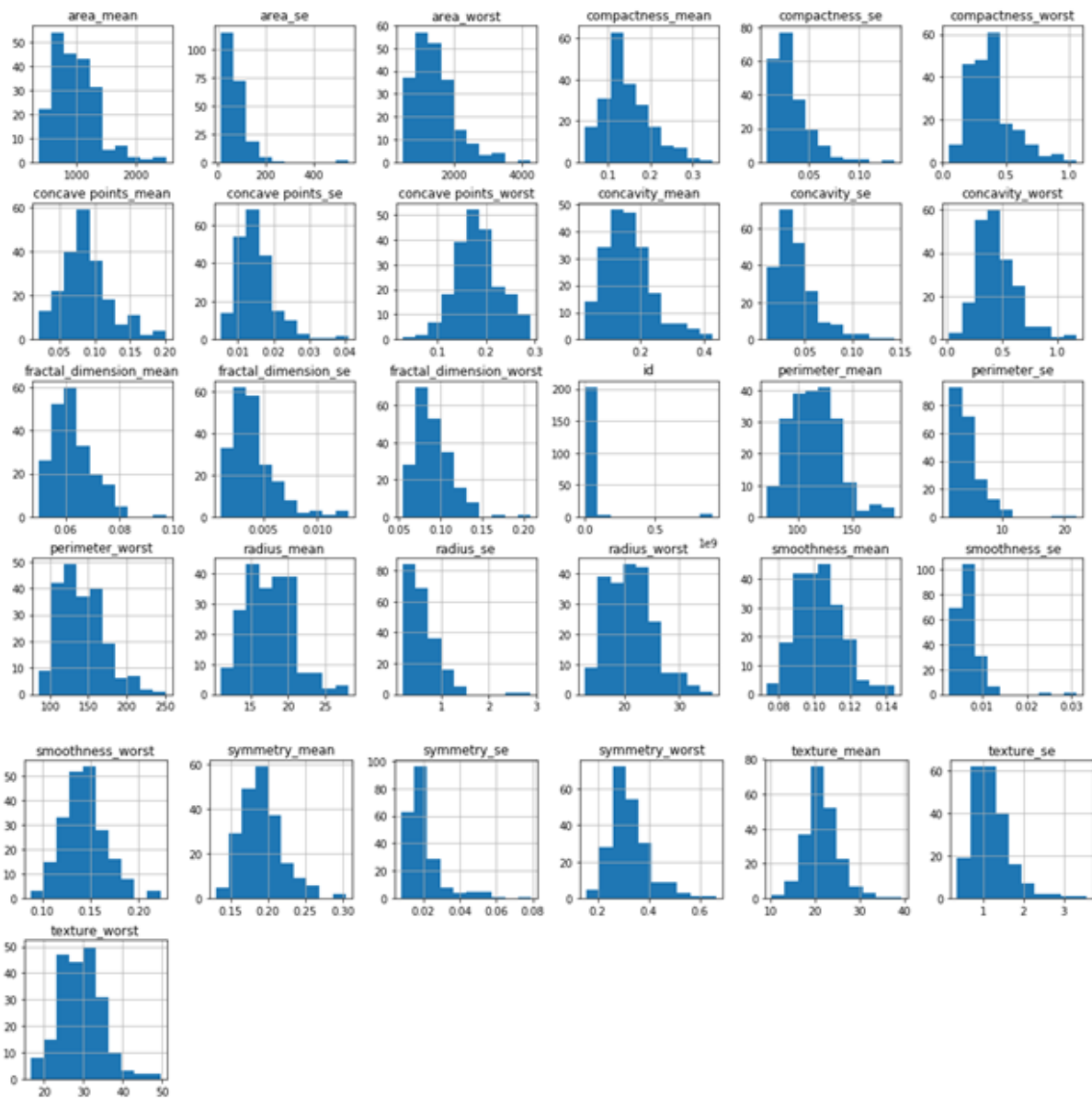
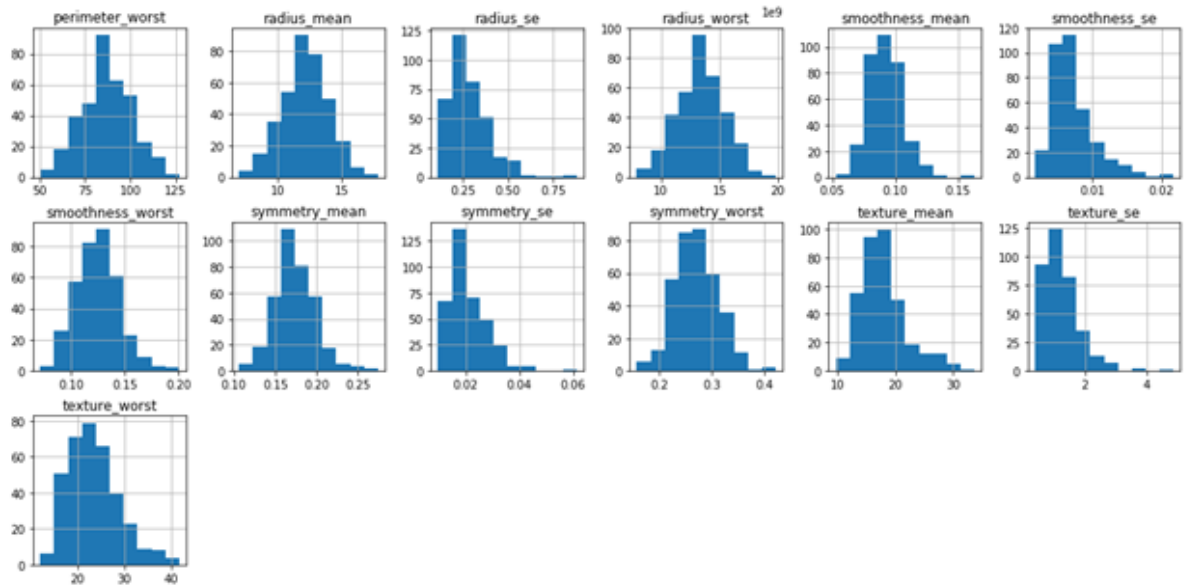


Fig : Visualization of Dataset

Missing or Null Data points

We can find any missing or null data points of the data set (if there is any) using the following pandas function.

```
dataset.isnull().sum()  
dataset.isna().sum()
```

```
id          0  
radius_mean 0  
texture_mean 0  
perimeter_mean 0  
area_mean 0  
smoothness_mean 0  
compactness_mean 0  
concavity_mean 0  
concave points_mean 0  
symmetry_mean 0  
fractal_dimension_mean 0  
radius_se 0  
texture_se 0  
perimeter_se 0  
area_se 0  
smoothness_se 0  
compactness_se 0  
concavity_se 0  
concave points_se 0  
symmetry_se 0  
fractal_dimension_se 0  
radius_worst 0  
texture_worst 0  
perimeter_worst 0  
area_worst 0  
smoothness_worst 0  
compactness_worst 0  
concavity_worst 0  
concave points_worst 0  
symmetry_worst 0  
fractal_dimension_worst 0  
diagnosis 0  
dtype: int64
```

Fig : Observe missing data

Phase 2 — Categorical Data

Categorical data are variables that contain label values rather than numeric values. The number of possible values is often limited to a fixed set.

For example, users are typically described by country, gender, age group etc.

We will use Label Encoder to label the categorical data. Label Encoder is the part of SciKit Learn library in Python and used to convert categorical data, or text data, into numbers, which our predictive models can better understand.

```
#Encoding categorical data values
from sklearn.preprocessing import LabelEncoder
labelencoder_Y = LabelEncoder()
Y = labelencoder_Y.fit_transform(Y)
```

Index	0
0	M
1	M
2	M
3	M
4	M
5	M
6	M
7	M
8	M
9	M
10	M
11	M
12	M
13	M
14	M

Fig: Diagnosis Data without Encoding

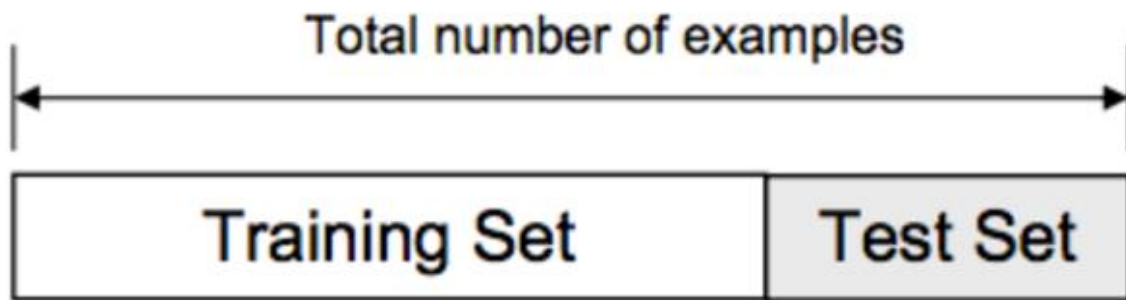


Fig: Training and test set

Phase 3 — Feature Scaling

Most of the times, your dataset will contain features highly varying in magnitudes, units and range. But since, most of the machine learning algorithms use Euclidean distance between two data points in their computations. We need to bring all features to the same level of magnitudes. This can be achieved by scaling. This means that you're transforming your data so that it fits within a specific scale, like 0–100 or 0–1.

We will use StandardScaler method from SciKit-Learn library.

```
#Feature Scalingfrom sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Project Design Phase - Part 2

Phase 4 — Model Selection

This is the most exciting phase in Applying Machine Learning to any Dataset. It is also known as Algorithm selection for Predicting the best results.

Usually Data Scientists use different kinds of Machine Learning algorithms to the large data sets. But, at high level all those different algorithms can be classified in two groups : supervised learning and unsupervised learning.

Without wasting much time, I would just give a brief overview about these two types of learnings.

Supervised learning : Supervised learning is a type of system in which both input and desired output data are provided. Input and output data are labelled for classification to provide a learning basis for future data processing. Supervised learning problems can be further grouped into **Regression** and **Classification** problems.

A **regression** problem is when the output variable is a real or continuous value, such as “salary” or “weight”.

A **classification** problem is when the output variable is a category like filtering emails “spam” or “not spam”

Unsupervised Learning : Unsupervised learning is the [algorithm](#) using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance.

In our dataset we have the outcome variable or Dependent variable i.e Y having only two set of values, either M (Malign) or B(Benign). So we will use Classification algorithm of supervised learning.

We have different types of classification algorithms in Machine Learning :-

1. Logistic Regression
2. Nearest Neighbor
3. Support Vector Machines
4. Kernel SVM
5. Naïve Bayes
6. Decision Tree Algorithm
7. Random Forest Classification

Lets start applying the algorithms :

We will use sklearn library to import all the methods of classification algorithms.

We will use LogisticRegression method of model selection to use

Logistic Regression Algorithm,

```
#Using Logistic Regression Algorithm to the Training Set
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, Y_train)

#Using KNeighborsClassifier Method of neighbors class to use Nearest Neighbor algorithm
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(X_train, Y_train)

#Using SVC method of svm class to use Support Vector Machine Algorithm
from sklearn.svm import SVC
classifier = SVC(kernel = 'linear', random_state = 0)
classifier.fit(X_train, Y_train)

#Using SVC method of svm class to use Kernel SVM Algorithm
from sklearn.svm import SVC
classifier = SVC(kernel = 'rbf', random_state = 0)
classifier.fit(X_train, Y_train)

#Using GaussianNB method of naive_bayes class to use Naïve Bayes Algorithm
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, Y_train)

#Using DecisionTreeClassifier of tree class to use Decision Tree Algorithm
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classifier.fit(X_train, Y_train)

#Using RandomForestClassifier method of ensemble class to use Random Forest Classification algorithm
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
classifier.fit(X_train, Y_train)
```

We will now predict the test set results and check the accuracy with each of our model:

```
Y_pred = classifier.predict(X_test)
```

To check the accuracy we need to import confusion_matrix method of metrics class. The confusion matrix is a way of tabulating the number of mis-classifications, i.e., the number of predicted classes which ended up in a wrong classification bin based on the true classes.

```
from sklearn.metrics import confusion_matrix  
cm = confusion_matrix(Y_test, Y_pred)
```

We will use Classification Accuracy method to find the accuracy of our models. Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

$$\text{Accuracy} = \frac{\text{Number of Correct predictions}}{\text{Total number of predictions made}}$$

Fig: Accuracy

To check the correct prediction we have to check confusion matrix object and add the predicted results diagonally which will be number of correct prediction and then divide by total number of predictions.

	0	1
0	87	3
1	3	50

Fig: Confusion Matrix

After applying the different classification models, we have got below accuracies with different models:

1. Logistic Regression — 95.8%

2. Nearest Neighbor — 95.1%

3. Support Vector Machines — 97.2%

4. Kernel SVM — 96.5%

5. Naive Bayes — 91.6%

6. Decision Tree Algorithm — 95.8%

7. Random Forest Classification — 98.6%

Project Development Phase

During the project development phase of "Cancer Vision: Advanced Breast Cancer Prediction with Deep Learning," several important steps and considerations should be taken into account. Here is an outline of the typical development phase:

Project Planning: Define the project goals, objectives, and deliverables. Determine the scope of the project, including the specific features and functionalities of the deep learning model. Establish a project timeline, resource requirements, and potential collaborations with medical professionals or institutions.

Data Collection and Preparation: Acquire a diverse and representative dataset of medical images containing both positive and negative cases of advanced breast cancer. Ensure proper anonymization and compliance with privacy regulations. Preprocess the data, including resizing, normalization, and augmentation techniques, to enhance data quality and address any imbalances or biases.

Model Development: Select an appropriate deep learning architecture, such as convolutional neural networks (CNNs), and design the model architecture based on the project requirements. Train the model using the prepared dataset, optimizing hyperparameters and employing techniques like regularization or transfer learning, if necessary. Iteratively validate and refine the model to improve its performance.

Evaluation and Validation: Assess the performance of the developed deep learning model using appropriate evaluation metrics, such as accuracy, precision, recall, and F1 score. Conduct cross-validation or split the dataset into training, validation, and test sets to ensure reliable performance estimation. Validate the model's predictions on external datasets or collaborate with medical professionals to conduct clinical trials for real-world validation.

Integration and Deployment: Integrate the trained deep learning model into a user-friendly software application or platform. Ensure compatibility with existing healthcare systems and compliance with relevant standards. Develop an intuitive user interface that allows healthcare professionals to input medical images and obtain predictions and probability estimates regarding advanced breast cancer.

Ethical Considerations: Address ethical concerns related to data privacy, informed consent, and the responsible use of AI in healthcare. Comply with relevant regulations and guidelines, ensuring patient data security and confidentiality. Implement mechanisms for model explainability and transparency to foster trust and facilitate ethical decision-making.

Continuous Improvement: Monitor the model's performance and collect user feedback to identify areas for improvement. Continuously update and retrain the model using new data to enhance its accuracy and robustness. Stay updated with advancements in deep learning techniques and incorporate relevant improvements to the model over time.

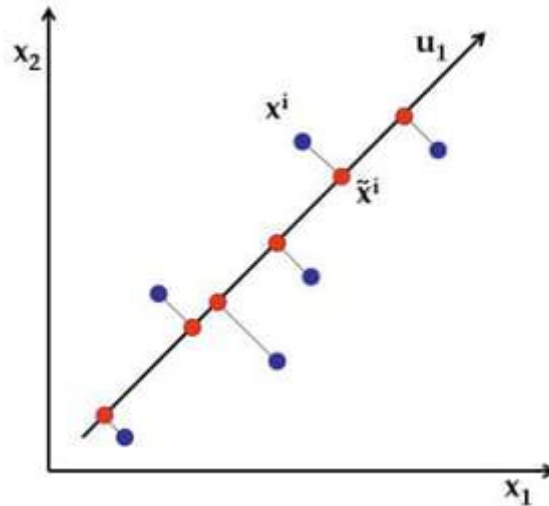
Throughout the project development phase, close collaboration with medical professionals, data scientists, and other stakeholders is crucial to ensure the model's clinical relevance, reliability, and real-world impact.

Technical Architecture

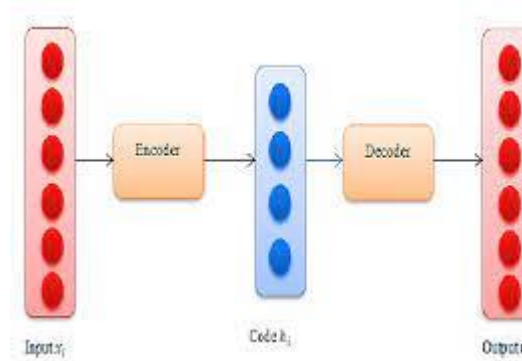
III. BACKGROUND OF THE STUDY

The Linear Discrimination (LDA) analysis and the Autoencoder neural network used to reduce complexity. The basic structure is described as follows A. STUDY OF THE MAJOR COMPONENT LDA is one of the most common methods for the reduction of linear dimensions, and it is considered to be a multivariate technique. It finds the significant elements in records that are unrelated constants using the eigenvalue matrix and vectors, each representing a specific variation of the data. Let $X = \{x_i\}_{i=1}^m$ mark the training data collection. x_i represents a D dimension variable, and it has the genomic patterns in this report. The goals of the LDA are:

- 1) To collect the main thing of data from x_m ;
 - 2) Compressing the X dimension by maintaining only necessary information.
- LDA aims to minimize the variance of the predicted data, as shown in figure . This means that orthogonal representations of the existing data are called on the new k -dimensional space



Taking the path of the image with a vector u_1 (dimensional D). Each x_i a data point is then transformed into a scalar value of $u_1^T x_i$. The average of the data predicted is equal to $u_1^T \bar{X}$, where \bar{X} corresponds to the average sample range provided by $\bar{X} = \frac{1}{m} \sum_{i=1}^m x_i$. In addition, the variation of the results predicted will be: $\frac{1}{m} \sum_{i=1}^m (u_1^T x_i - u_1^T \bar{X})^2 = u_1^T S u_1$, Where S belongs for all samples to a standard covariance matrix: $S = \frac{1}{m} \sum_{i=1}^m (x_i - \bar{X})(x_i - \bar{X})^T$. LDA now seeks to optimize the predicted variance $u_1^T S u_1$ in addition to u_1 .



B. NEURAL NETWORK AUTOENCODER

As illustrated in figure 3, an Autoencoder is a feed-forward neural network that is often specialized in the processing of depictions or successful decoding of original data $X = \{x_i\}_{i=1}^m$. It allows the $g(f(x_i)) \approx x_i$ system to be understood as an indication for estimating and representing the input data produced from a finite amount of available activations. Then the overall Autoencoder architecture is broken down into the following parts: 1) The data tools x_i ; 2) An encoder function f ; 3) A hidden representation or “code” $h_i = f(x_i)$; 4) A role of decoder g ; 5) Input modules called “rebuilding” $ri = g(h_i) = g(f(x_i))$; 6) $L(x_i, ri)$

loss function $\text{Scalar } k_{xi} - \text{rik2}$ computing that tests how successful the restoration r_i has originally from x_i data. The Autoencoder is designed to maximize L over Learning Examples X . The predicted values are reduced. The Autoencoder is simpler, and the encoder functionality eliminates non-linear dimensions if the amount of the hidden layer is more significant than one. Authenticators are trained to identify the best input compression feature on these cached layers, where the measurements of the hidden layers are smaller than that of the data. Alternatively, the Autoencoder can be trained to map the feature to a larger space

Project Development Phase

In this section, the proposed LDA and AE based deep learning framework have been discussed. Next, it defines five data sets for gene expression that uses a specific pre-processing method. A. DATASETS AND PRE-PROCESSING OF GENE EXPRESSION Information on gene expression has been downloaded from the GEO database. Every study includes 129,158 Genomic versions of the Affymetrix microarray platform, and each profile has 22,268 mutations, equivalent to 978 landmarks and 21,290 aim genes. Especially from the LINCS cloud, the alternative approach has tested in five different data sets for breast cancer. The information is given in-depth in Table 1.

TABLE 1. Samples have been removed if patients within 5 years are censored for treatment.

Records	Bad result	Better result	Final result	Samples were taken
GSE2990	84	154	238	12
GSE3494	32	101	133	125
GSE9195	25	87	112	186
GSE17705	34	168	202	15
GSE17907	27	189	216	61

Patients use various immune and physiological factors in the five samples to affect the prognosis of outcomes. For example, all ER-positive patients in GSE3494 include ER-positive and ER-negative patients in addition to other data sets. In view of the classification mission, the pre-processing with the 5 data sets has carried out in two steps: The first argument is to follow the dataset partitioning method with a weak prognostic (set to 1) and reliable predictive (set to 0) division of all people with cancer. Data on aid has eliminated from the review of patients receiving adjuvant or screened for 5 years. In addition, it has been quantified the five datasets with a MAS 5.0 algorithm and converted all of the samples into an Expression gene ID, because the microarray models are used to calculate gene expression value.

- GENERAL APPROACH

The aim is to integrate both feature selection and feature elimination with profound learning basics that it learns from genomic profiles quite

concisely and establish a more accurate classification of cancer prognoses. The flowchart of the method is shown in Figure

- **DEEP LEARNING ASSISTED UNSUPERVISED FEATURE LEARNING**

Two stages are comprised of this deep learning approach are discussed as follows • **LDA:** Considering that data on gene expression is extensive, containing repetitive, noisy data, the LDA program (as defined in section I-A) is utilized as the tool for selecting features to reduce genomic model complexity. LDA conducts a linear estimation of the existing data and, in the meantime, maintains essential information. • **Autoencoder:** The result is simply a continuous representation of the source data since LDA has been applied. An optimized version of LDA characteristics is subsequently incorporated in a feature extraction system in addition to raw attributes, to capture non-linear interactions between the expressions of different genes. For feature extraction, it uses an Autoencoder neural network.

- **DEEP LEARNING ASSISTED SUPERVISED CLASSIFIER LEARNING**

The features extracted from the proposed two-phase unregulated attribute learning methodology are ultimately applied to a set of labels for classifier learning to forecast clinical outcomes for cancer patients. The method of classification is subject to test labels that indicate directed classification preparation. The key components for women with breast cancer therapies are outlined below. • First, provided multiple genomic profiles $X = \{x_i\}_{i=1}^m$ • Second, the primary analysis component is used to learn compressed feature sets $\hat{X} = \{\hat{x}_i\}_{i=1}^m$, for $1 \leq m \leq P$ $\hat{x}_i = \frac{1}{u} \sum_{t=1}^T x_{it}$; • Third, the raw expression of genes X and compressed feature \hat{X} is combined into \tilde{X} , where $\tilde{X} = \{\tilde{x}_i\}_{i=1}^m$ $\tilde{x}_i = (\hat{x}_i, x_i)$ $i=1$. The inputs of the autoencoder neural network \tilde{X} are considered to allow deep feature learning techniques for more complex representation $h(2)$. • The compressed features \hat{X} and profile $h(2)$ are eventually combined

to form an integrated classifier in a robust X 0 manner. Therefore, an LDA-Ada compartment is designed to use LDA compressed results as input properties to deal with a twostage classifying feature learning system

DETAILS OF IMPLEMENTATION A

ALIGNMENT OF DATA

For example, compact characteristic vectors in different dimensions can be accomplished when the LDA algorithm is used for the expression for dimension reduction tests because the scale and the source value of the neural network differ from one of the individual datasets. To fit the model, it has been built into data of different sizes; it has placed all the functional vectors with null values in a single direction without compromising performance.

FEATURE EXTRACTION TARGET FUNCTION

The second step has to develop a deep neural network by hierarchically stacking many Autoencoders in the proposed Feature Analysis.

THE NON-LINEAR METHOD The autoencoder and decoder portion consists of several nonlinear processing layers which are analyzed based on the mixture of the initial \tilde{X} data that has been used as data entry to analyze the non-linear model and the corresponding mathematical formulation has been equated as follows in the Eq(1): $h(1) = \sigma(\omega_1 \tilde{X} + b(1))$, $h(j) = \sigma(\omega_j h(j-1) + b(j))$, $j = 2, \dots, n$. (1) As discussed in the Eq (1) n indicates the surface number, and σ specifies the aspect of activation. $h(j)$, ω_j and $b(j)$ represent in the j th layer as a hidden vector, mass index, and bias vector.

LOSS IN REBUILDING

Autoencoder tries to reduce the discrepancy in inputs \tilde{X} to restored output $sh(n)$. Despite the infinite number of parameters in the quantum system and the existence of small samples, the possibility of overfitting is challenging for training deep neural networks. It is placed some sparsity penalties on the hidden layers to mitigate this issue, thus the loss of reconstruction as follows in equation (2): $L_{rec} = \|\tilde{X} - h(n)\|_2^2 + \eta \sum_{j=1}^n \|b(j)\|_2^2$. (2) Here, $h(n)$ are the effects, and

η is a hyperparameter for balancing the biases of the individual parts. The network is defined heuristically as a single input layer, 3 exposed layers, and a single output layer.

AUTOENCODER ACTIVATION FUNCTION

The ELU (Exponential Linear Unit) has been used as the activator function to speed up formation in deeper neural networks, and to increase precise classification σ is shown in equation (3) and (4) $\sigma \square h(j) \square = \square \square \square \square \square h(j)$ if $h(j) > 0$ $\alpha(\exp h(j) - 1)$ if $h(j) \leq 0$ (3) $\sigma \square h(j) \square = (1$ if $h(j) > 0$ $\sigma h(j) + \alpha$ if $h(j) \leq 0$ (4) In this case, the ELU hyperparameter α (set $\alpha = 1.0$) controls the value to which an ELU saturates for negative input. 4) CUSTOMIZATION It uses a customization approach based on a stochastic gradient. In combination with RMSProp, which operates in a very positive fashion either in lines or in non-stationary processes with different parameter values from original and second instants, the Adaptive training scores measure the advantages of both AdaGrad and RMSProp. The conditions for this work are 64 lots, iteration times are 10k, and the limit for learning is 0.001. 5) STANDARDIZED STARTUP It has initialized the distinctions as 0 and the matrix of weight ω_j in each layer with the standard uniform distribution as described in the neural network training. $\omega(j) \sim U \square -1 \sqrt{k}, 1 \sqrt{k} \square$ (5) If $U[-a, a]$ is an interval uniform distribution $-1 \times 10^{-4}, 1 \times 10^{-4}$ and k is the clock layer scale (number of columns ω).

CURRENT TECHNICAL REVIEW

The most frequently used IHC markers are Ki-67, estrogen receptors, progesterone receptor, P53 protein, and the human epidermal factor-2. One of the most important diagnostic and predicting indicators in the growth of breast cancer is a nonhistone protein. Radiation and chemotherapy are incredibly susceptible to patients with high Ki-67. For breast cancer, Ki-67 expression has a substantial predictive and prognostic benefit. 1) REVIEW OF DEEP LEARNING METHODS Automated Ki-67 scoring and identification of points via a deep learning approach have not yet been tried. This paper contains significant contributions

Design of a revolutionary in-depth thought method for the recognition and evaluation of stained hotspots. • Inclusion of the validation framework in the latest deep learning system. • This is an

additional benefit for the vital quantification of Ki-67 scoring techniques already developed.

EXPERIMENTAL CONFIGURATION

- Preparing slides and acquiring images.
- The slide with its histological parts has stained with a monoclonal antibody Ki-67. In compliance with the institutional protocols, all procedures such as slide planning, image processing, etc. are carried out.

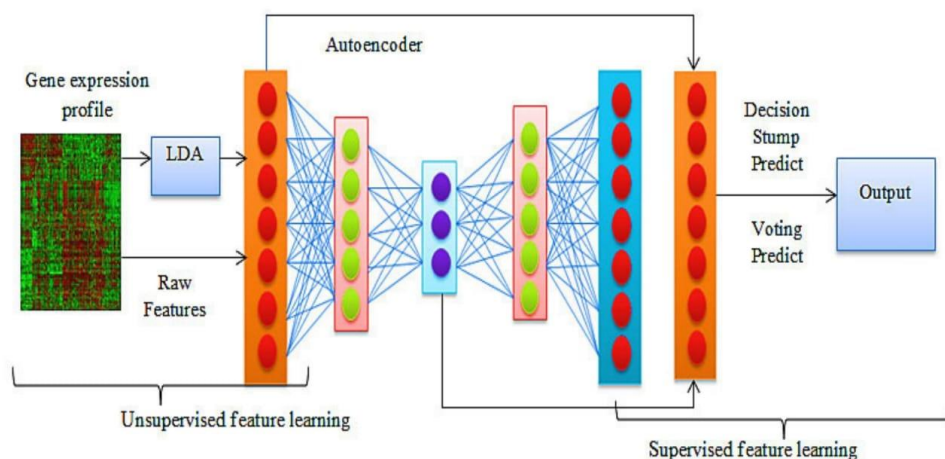


FIGURE 4. Flowchart of the proposed method.

CONVOLUTIONAL NETWORK LAYERS

A CN consists of multi-sub-sampling, sub-sampling, nonlinear, and entirely interconnected layers. Let f is a CN with N number layers serial composition or function (f_1, f_2, \dots, f_N) . Mappings can be expressed between input (w) and output (u) as shown in the following equation (6): $u = f(w; X_1, X_2, X_3, \dots, X_N) = f_1(w; X_1)$ of 2 ($.; X_2$) \dots of $N-1$ ($.; X_{N-1}$) of N ($.; X_N$) (6) f_N has historically been delegated to carry out spatial convolution or non-linear activation or classification. Where X_N shows bias and weight vector for the n th layer f_N . According to the range of η training data $\square (w(i), u(i))_{\eta i=1}$, vectors $(X_1, X_2, X_3, \dots, X_N)$ can be determined as follows: $\arg \min_{X_1, X_2, X_3, \dots, X_N} \frac{1}{\eta} \sum_{i=1}^{\eta} f_{\text{Loss}}(f \square w(i); X_1, X_2, X_3, \dots, X_N \square, u(i))$ (7) where f_{Loss} implies loss function. Stochastic reduction and reverse propagation strategies can accomplish Equation 7. In the computation of a feature map, a convolutional layer typically utilizes convolutional filters. The feature map FM_h in the equation at m level is shown in equation (8)

$FM_h^m = f \square \alpha_h^m + X_j FM_{h-1}^j \times G_{hjm} \square$ (8) FM_{h-1}^j in and FM_h^m out are some characteristics of input and output. Biases and kernels respectively are G_{hjm} and α_h^m . Two components create feature maps for each convolution layer. The local receptive area is the first element, and mutual weights are the second part. The benefits of this method are the ability to measure the image size of the data and to create a positive difference in local regions. The following equation is used to determine the function. $9_j = \max(\psi_{n \times n} i z(n, n))$ (9) Here ψ is the input image, z indicates the role of the window and n listed simply the size of the input patch. Rectified Linear Units (ReLUs) are used as a tool for activation and descent gradient. $q(r) = \max(0, r)$ (10) where q shows the output component of the model with the r information, each layer has the same size for input and output. Let us consider X and Y , respectively, for the data and the finite output spaces. The decision tree also applies to decision nodes as internal branch nodes, indexing them with D . Similarly, nodes of prediction are suggested by P as terminal nodes. The decision-making feature fd has allocated for each decision node $0 \in D_{fd} (.,) : X \rightarrow [0, 1]$. The probability distribution π_p over Y is available in each $p \in P$ projection node. If the reference $x \in X$ enters node of judgment d , it will propagate to the right or the left of the substratum based on a $fd(x;)$. The final results for sample x of the tree T with decisions parameterized with oscillating are shown by the following equation $PT[y|x, , \pi] = X_{p \in P} \pi_p y \mu_p(x |)$ (11) In this scenario, $\pi_p y$ and $\pi = (\pi_p)_{p \in P}$ is a likelihood that the sample would enter leaf p on class y and identify by $\mu_p(x |)$ when $x \in X$, $P_p \mu_p(x |) = 1$. Decision nodes are based on the stochastic routine and have been described as: $fd(x;) = \sigma(f_r(x;))$ (12) The sigmoid function $\sigma(x)$ in this case is set to $\sigma(x) = 1 / (1 + e^{-x})$. the decision forest is called a group of decision-making trees and is defined by the following equation (13) $F = \{T_1, T_2, \dots, T_z\}$ (13) Let I is a $= I_1, I_2, I_3, \dots, I_Q$, in which Q displays a set of pixels and I_Q means the size of the gray-level of a pixel L , $K = (K_1, K_2, K_3, \dots, K_Q$ where $K_Q \in LL = \{0, 1\}$ can be generalized to a set with positive labels. $K^* = \arg \min_k \{Y(I|K,)Y(K)\}$ (14) $Y(K)$ is a distribution of Gibbs. Equation 15 can be written as in the Expect-Maximization algorithm. $K^* = \arg \min_{K \in k} \{U(I|K,)U(K)\}$ (15) Here U

corresponds to urinary potential or energy of chance and is indicated by $U(I|K,) = X Q " (I Q - \mu K Q) 2 2 \sigma 2 K + \ln \sigma K \# (16)$ According to the theorem, it assumes a Gaussian distribution with parameters $\sigma_{xi} = \mu_{xi}$, σ_{xi} follows the segmented area strength. This theory cannot model events in real life. (Gamma Mixture Model)GMM is, therefore, the best choice for engineers for dynamic delivery. Below are the calculations of a GMM with c elements. $\sigma_i = \{ \mu_{i,1}, \sigma_{i,1}, w_{i,1}, \dots, (\mu_{i,c}, \sigma_{i,c}, w_{i,c}) (17)$

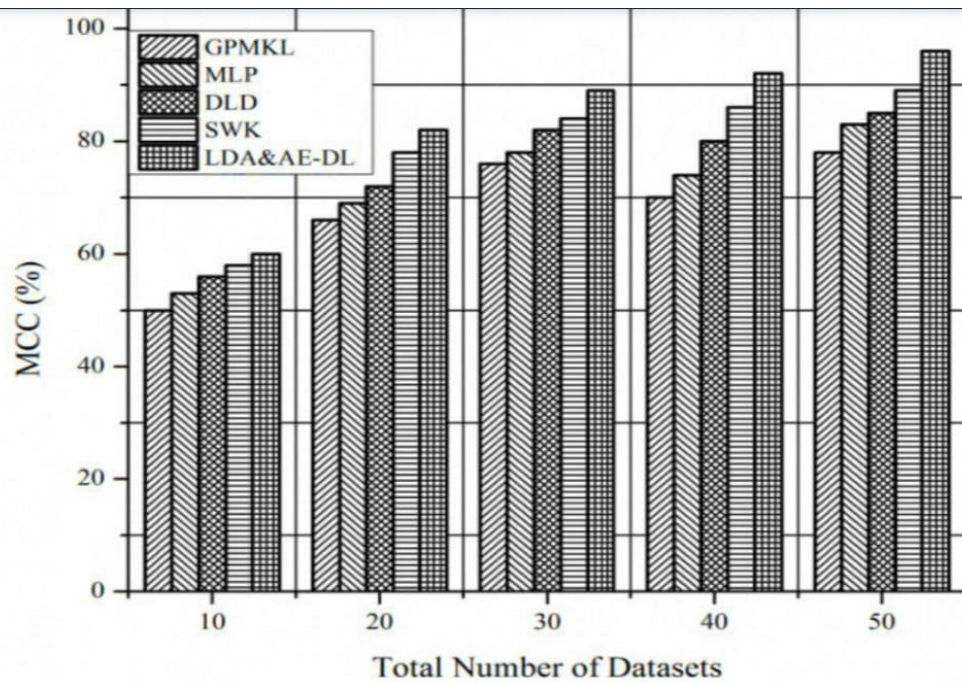
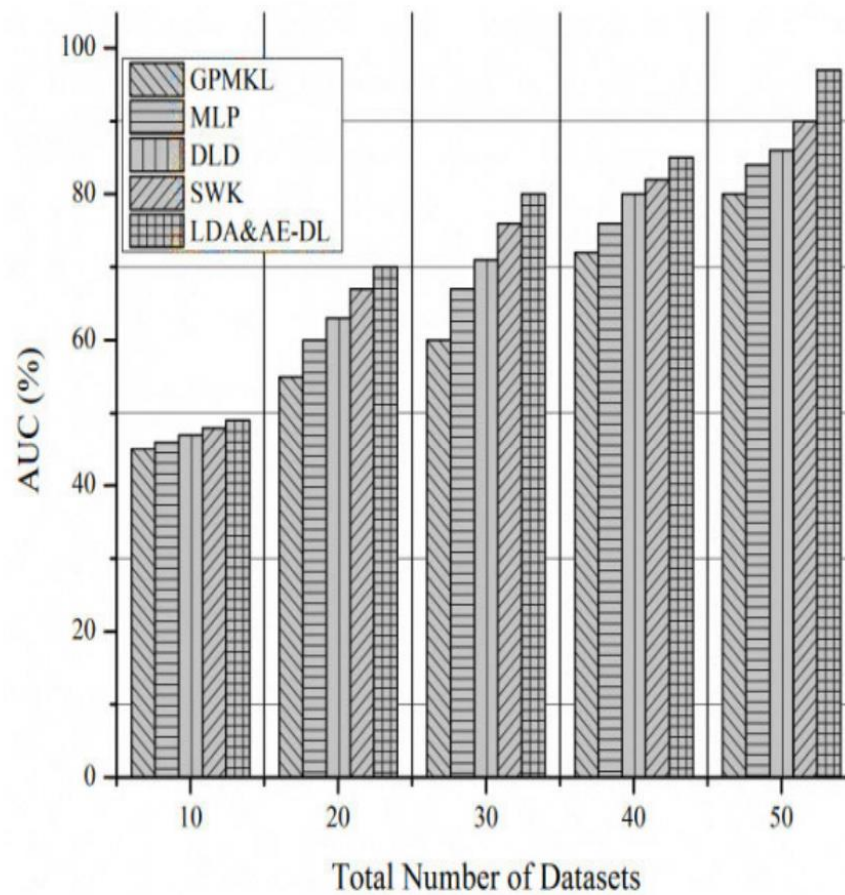
Performance & Final Submission Phase

Compared to the LDA&AE-DL classification performance with those of other classifiers, the LDA-AE works better than other methods such as GPMKL, MLP, DLD, SWK, which are built with LDA compressed features for all the evaluation metrics. LA-AE-DL shows a better performance, while LDA-DL works well only in positive cases. Hence, Deep learning has been effectively eliminating the problem of uneven distribution of training and improves classifier's capacity to generalize breast cancer. FIGURE 5. AUC evaluation. Figure 5 shows that this advanced ensemble classification provides the best AUC(Area Under Curve) performance over TABLE 2. MCC evaluation. several datasets, although the gene set methods are not based on the datasets. Hence, the gene-set ways that achieve higher AUC efficiency than that of the two gene classifiers. The MCC scores indicate a similar phenomenon. Table 2 shows the evaluation of the Matthews correlation coefficient of proposed LDA&AE-DL in comparison with GPMKL, MLP, DLD, SWK. FIGURE

TABLE 2. MCC evaluation.

Total Number of Datasets	GPMKL	MLP	DLD	SWK	LDA&AE-DL
10	50.6	53.3	56.8	58.4	60.3
20	66.5	69.1	72.8	78.5	82.2
30	76.8	78.2	82.1	84.3	89.9
40	70.5	74.3	80.2	86.1	92.7
50	78.4	83.6	85.1	89.7	96.8

MCC evaluation. From Figure 6, the proposed classification system (especially those based on physical signature), performed well on a dataset. This is because these two datasets have both a negative lymph node and a positive lymph node, whereas the other datasets have only lymph node-negative patients. Remarkably, the high performance of specific advanced ensemble classifiers [22], [23] achieves 96.7% MCC, which outperforms the others substantially. The above observation shows that the method being proposed is less susceptible to unbalanced data sets and is indeed more stable, compared with the four other categories which show dramatic changes in the various datasets. However, the AUCs and MCCs are compared with further four combined methods in public datasets. Figure shows the outcomes to demonstrate the efficiency of a particular way more in detail.



Complete analysis indicates that this AUC classification reaches more than 75.3%, while the two genetic classification devices have a significantly worse outcome of 75.8%. However, both genetic classification devices can only achieve AUC 97.4%. The MCC indexes will demonstrate a similar phenomenon

CONCLUSION

This paper incorporates a linear discriminant analysis (LDA) with an Autoencoder neural network with deep learning techniques to learn from the gene expression information with most characteristic features. This uses the deep learning algorithm at the stage of classification to create an advanced ensemble classification for the prediction. Hence, the suggested system has more prediction capacity with deep learning classification compared to other techniques, which are shown in evaluation results. This analysis showed excellent ability to generalize quickly and explicitly improve the performance of the prediction of the results with 98.27% of accuracy, which has been automatically obtained from the network. This approach has great potential for generalization, and it must be further enhanced with more public data sets