Chapter 1

Introduction

1.1 Introduction

Cancer is a disease in which some of the body's cells grow uncontrollably and spread to other parts of the body. Cancer can start almost anywhere in the human body, which is made up of trillions of cells. Normally, human cell grow and multiply (through a process called cell division) to form new cells as the body needs them. When cells grow old or become damaged, they die and new cells take their place. Sometimes this orderly process breaks down, and abnormal or damaged cells grow and multiply when they shouldn't. These cells may form tumors, which are lumps of tissues. Tumors can be cancerous (Malignant) or non-cancerous (Benign). Cancerous tumors spread into; or invade nearby tissues and can travel to distant places in the body to form new tumors and it is called malignant tumors. Benign tumors don't spread into, or invade; nearby tissues. When removed, benign tumors usually don't grow back, whereas cancerous tumors sometimes do. There are different types of cancer like skin cancer , breast cancer , lung cancer, prostate cancer, kidney (renal) cancer. The most prevalent and serious cancer that affect women is the breast cancer. A large number of women succumb to breast cancer each year. Recently after skin cancer, breast cancer is the second most hazardous cancer diagnosed in women worldwide and becomes the reason for death. Breast Cancer is a cancer that develops from breast tissues. The first symptom of breast cancer is usually an area of thickened tissue in the breast or as lump in the breast or an armpit or breast pain does not change with the monthly cycle, pitting like the surface of an orange or color changes such as redness in the skin of the breast, a rash around or an one nipple, discharge from a nipple which may contain blood, a sunken or inverted nipple, a change in the size or shape of the breast, pealing or flaking or scaling of the skin of the breast or nipple [ref: medicalnewstoday.com/articles/37136]. The several risk factors that cause breast cancer are age, genetics, a history of breast cancer or breast lumps, dense breast tissue, estrogen exposure and breast feeding, body weight, alcohol consumption, radiation exposure, hormone treatments [ref: https://www.medicalnewstoday.com/articles/37136.php]. As per the survey report of GLOBOCAN 2012, Ferlay et al. (2014) found that, 1067 million women were detected with breast cancer [38]. Also, the authors discovered that the breast cancer has highest proportion of 25% out of all cancers within women. In the research of Global Cancer Statistics 2018, Bray et al. (2018) stated that, 2.1 million new cases of breast cancer were identified and out of all registered cases of breast cancer, 53% were diagnosed as malignant [21]. As per the survey report of WHO, each year 2.1 million women are impacting with breast cancer and also large number of

women die due to deficiency in early diagnosis and early treatment [21]. In 2018, 627,000 women were died due to breast cancer. As per the survey report of WHO in 2020, 2.3 million women were diagnosed with breast cancer and 685000 were died worldwide [who.int/news-room/fact-sheets/detail/breast-cancer].

Early detection of breast cancer is important as it is associated with an increased number of available treatment options, increased survival and improved quality of life. Majority of the all affected women are middle-aged that is in the age of 50's & 60's [109]. Early detection provides the best chance of effective treatment. The earlier the stage of breast cancer the better the chance of survival. The very early detection of breast cancer have a 97 to 100% chances of cure but once it spreads to the lymph nodes or elsewhere, the chance of cure goes down significantly. If an early diagnosis is made, patients can avoid the cost of different tests such as mammograms, ultrasounds, other imaging tests, biopsies, well as at the same time they can reduce the number of frequent visits to the doctor which can help them mentally and financially. Early diagnosis can also save that doctor's time and they can reach to the more patients.

It is most essential to identify and cure breast cancer in its early stage. While successful treatment depends on early detection, the diagnosis of breast cancer is difficult due to the dense breast tissues with the detection being subject to human error, the doctors looked for a way to improve the accuracy of the diagnosis. With the help of computer aided technologies and Artificial Intelligence (AI), it is possible to make early diagnosis of breast cancer. Development of such a tool or system is required to make early diagnosis of breast cancer using soft computing techniques.

In medical science, an extensive and diverse spectrum of applied mathematics research is being conducted. AI or ML is all about mathematics, which in turn helps in creating algorithm that can learn data to make an accurate predication. Machine Learning (ML) is an emerging technique which provides an efficient way to enhance the knowledge in data in order to improve the performance of the disease predictive models. There are server ML algorithm like Support Vector Machine (SVM), Artificial Neural Network (ANN), Deep Learning (DL), etc.. Using these algorithms AI is built into machines. The basic requirements for any intelligent behavior is learning. Soft computing approaches are being used in medical science by researches worldwide. The thesis is concerned with the diagnosis of breast cancer through the application of various soft-computing approaches, with a particular emphasis on Kernel-based methodologies. Classification plays an important role in medical science where data mining techniques are used to diagnose and analyse disease at

an early stage.

1.1.1 Literature survey

Many researchers have worked in diagnosis of breast cancer using various soft computing techniques. Liu et. al. used SVM for classification of breast cancer data and achieved 96.71% accuracy with polynomial kernel and 97.07% accuracy with radial basis function kernel [70]. Chen et. al. and Keerthi have classified the breast cancer after applying various feature selection techniques like rule extraction, roughest based feature selection, Genetic Algorithm (GA) etc and obtained good classification accuracy [27, 41]. Polat and Salih developed the least square SVM Classifier and Obtained 98.53% accuracy [95, 103]. Also Akay, Maglogiannis et. al. and Osareh et. al. had built SVM and compared with other classifiers like Bayesian, ANN , K-nearest neighbors probabilistic neural network and obtained nearby 97% of classification accuracy [7, 74, 90].

For different data set like Wisconsin Breast Cancer, Wisconsin Diagnostic Breast Cancer, Wisconsin Prognostic Breast Cancer, Aalaei et. al. employed ANN with GA based feature selection and a Particle Swarm optimization algorithm based classifier (PS- classifier) to diagnosis of breast cancer [1].Abdel-Zaher et. al. employed Deep Neural Network as a classifier with recursive feature elimination technique [3].Karabatak et. al. developed ANN classifier based on association rule and implemented on WBC data set [57]. Agarap Abien Fred M. experimented Six ML method on WBC data set namely Gated Recurrent Unit with SVM; Linear Regression, Multilayer Perceptron (MLP), Nearest Neighbor (NN) search [6].P.R. Innocent et. al. conducted a study of fuzzy methods for medical diagnosis in nursing assessment using Type-II fuzzy sets (2007) [51]. Many authors namely Baig et. al., Awotunde et. al. and Madkour et. al. have developed a control system using fuzzy logic in diagnosis of various disease like brain tumor, malaria, wherping cough, chickenpox etc [16, 15, 73]. Elif Derya Übeyli proposed an integrated view of ANFIS to detect breast cancer and tested on WBC [122]. Seyedesh S.N. et al. designed a hierarchical fuzzy neural system with Extended Kalman Filter (EKF) [86]. M. Ashraf et. Al. introduced an information gain technique with ANFIS for breast cancer classification [14]. Chakravarthy and Ghosh demonstrated scale based clustering with Radial Basis function network [26]. Kiyan and Ypldrim proposed statistical neural network topology in RBFN and Compared with various classifiers like ANFIS, ANN, RBFN and evaluated on WBC data set and they achieved 97.55% success rate [64].

1.2 Methodology for classification

The literature survey depicts that most of the research have been focused on the diagnosis of the breast cancer and the researchers have proposed various predictive models using the benchmark data sets namely Breast Cancer Wisconsin (Diagnostic) Data sets. This data sets are widely available on the University of California at Irvine (UCI) Machine Learning repository. Our aim is to diagnose the breast cancer based on Breast Cancer Wisconsin (Diagnostic) Data sets. Features of these data sets 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. All features are assigned numeric values with four significant digits. In the data set, diagnosis is also specified by "2" for malignant and "4" for benign breast tumors. Hence, this is classification problem.

Using the same data sets, we have proposed various predictive models along with the time analysis for classification of breast tumor using various soft computing techniques like Artificial Neural Network (ANN), Support Vector Machine (SVM), Deep Learning (DL), Radial Basis Function Network (RBFN) and Adaptive Neuro Fuzzy Inference System (ANFIS). We employed various optimization techniques like Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam), Limited-memory Broyden fletcher Goldfarb Shanno (L-BFGS), Particle Swarm Optimization (PSO) in training the proposed models. We have also proposed various feature reduction techniques like Principal Component Analysis (PCA), Independent Component Analysis (ICA), Relief Based algorithm in data pre-processing. Proposed predictive models are implemented on WDBC and WBC data sets for classification of tumor into benign or malignant. Details of the data sets are mentioned in Appendix. The comparative analysis of our predictive models with predictive models of other researchers is carried out in detail and we found that the predictive models proposed by us gives highest classification accuracy in just few seconds.

1.3 Organization of the thesis

The layout of the thesis along with the proposed methodologies used in constructing the predictive models for classification of breast cancer (described in chapters 3 to 6) is as follows.

1.3.1 Chapter 1: Introduction

This chapter mainly deals with the motivation as well as literature survey of the breast cancer.

1.3.2 Chapter 2: Mathematical Preliminaries

This chapter concerns with the mathematical concepts used throughout the study.

1.3.3 Chapter 3: The Ultimate kernel machine based on Support Vector Machines

In machine learning, Support Vector Machines are supervised learning models with associated learning algorithms that analyse data for classification, regression analysis and outliers detection. This algorithm has a good generalisation ability, better performance and a robust mathematical theory. Machine learning, optimization techniques from operations research, and kernel functions from functional analysis are all combined in this approach. It is often referred to as a large margin classifier. When it comes to diagnose breast cancer, SVM has proven to be extremely effective. To build the cost-effective kernel machine for breast cancer diagnosis, the tools of PCA and k-fold Cross-Validation (CV) techniques are employed. The model is implemented on WDBC and WBC data sets to check the condition of the tumor for its malignancy. Classification accuracy and time computation are obtained and comparative experimental results are analysed under different conditions. For WBC data set, 100% accuracy is obtained using Polynomial kernel in just 0.03 second.

1.3.4 Chapter 4: Regularized Deep Neural Network with hybrid approach of Independent Component Analysis

Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised. One of the main advantages of deep learning lies in being able to solve complex problems that require discovering hidden patterns in the data and/or a deep understanding of complex relationships between a large number of interdependent variables. It not only has the ability to tackle nonlinear programming problems with the restrictions of equality and inequality, but it also

has a greater overall performance.

In this study, we investigate the use of Regularized Deep Neural Network (R-DNN) for the prediction of breast cancer. A variety of optimization techniques, such as Limited-memory Broyden Fletcher Goldfarb Shanno (L-BFGS), Stochastic Gradient Descant (SGD), Adaptive Moment Estimation (Adam), and activation functions like as Tanh, Sigmoid, and Rectified Linear Unit (ReLu) are used in the simulation of R-DNN. The Independent Component Analysis (ICA) approach is used to identify the most effective features to be used in the study. To measure the efficacy of the model, training and testing of the proposed network is carried out using the WDBC and WBC data sets. The detailed analysis of the accuracy is carried out and compared to the accuracy of other author's model. We find that the proposed network attains the highest accuracy.

1.3.5 Chapter 5: A Hybrid Approach of Adaptive Neuro Fuzzy Inference System and Novel Relief Algorithm

An Adaptive Neuro-Fuzzy Inference System or Adaptive Network-based Fuzzy Inference System (ANFIS) is a kind of artificial neural network that is based on Takagi–Sugeno fuzzy inference system. The technique was developed in the early 1990s [53, 55]. Since it integrates both neural networks and fuzzy logic principles, it has potential to capture the benefits of both in a single framework. Its inference system corresponds to a set of fuzzy IF–THEN rules that have learning capability to approximate nonlinear functions [4]. Hence, ANFIS is considered to be a universal estimator [52].

ANFIS provides accelerated learning capacity and adaptive interpretation capabilities to model complex patterns and apprehends nonlinear relationships. It is possible to identify two parts in the network structure, namely premise and consequence parts. In more details, the architecture is composed by five layers. The first layer takes the input values and determines the membership functions belonging to them. It is commonly called fuzzification layer. The second layer is responsible of generating the firing strengths for the rules. The role of the third layer is to normalize the computed firing strengths, by dividing each value for the total firing strength. The fourth layer takes as input the normalized values and the consequence parameter set. The values returned by this layer are the defuzzificated ones and those values are passed to the last layer to return the final output.

The proposed model introduces a hybrid strategy of effectively diagnosing breast cancer by using a novel Relief algorithm for feature selection with an Adaptive Neuro-Fuzzy Inference System. The efficiency of this proposed hybrid model and the ANFIS model without using any feature selection technique are estimated using WBC data set. The study finds that the new hybrid model has attained highest accuracy of 99.30% and is ideal for detecting breast cancer.

1.3.6 Chapter 6: Ensemble Based Lasso Ridge Radial Basis Function Network

Radial Basis Function Networks (RBFNs) are commonly used artificial neural network for function approximation problems. RBFNs are distinguished from other neural networks due to their universal approximation and faster learning speed. The RBF network is a type of feed forward neural network composed of three layers, namely the input layer, the hidden layer and the output layer. The first layer corresponds to the inputs of the network, the second is a hidden layer consisting of a number of RBF non-linear activation units and the last one corresponds to the final output of the network. Activation functions in RBFNs are conventionally implemented as Gaussian functions. The input layer is not a computation layer, it just receives the input data and feeds it into the special hidden layer of the RBF network. The computation that is happened inside the hidden layer is very different from most neural networks, and this is where the power of the RBF network comes from. The output layer performs the prediction task such as classification or regression.

RBF neural network architecture, which includes Lasso and Ridge Regularisation and Ensemble learning, is used in the proposed approach. This has several advantages including greater approximation capabilities and shorter processing time. Various RBF networks come together to form an ensemble. This study uses ensemble RBF networks to detect breast cancer. We achieved 100% of classification accuracy for diagnosis of breast cancer. Also novel RBF kernel is introduced with Particle Swarm Optimization technique and obtained 97% classification accuracy.

1.4 Conclusion

The predictive models based on various soft computing techniques like Support Vector Machines, Deep Neural Network, Adaptive Neuro Fuzzy Inference System and Radial Basis Function Network along with different optimization techniques and different architectures are built and implemented on WDBC and WBC data sets for classification of breast cancer. The comparative analysis of the performance of the classification accuracy obtained by the various researchers in their models with our respective proposed models is carried out. The proposed predictive models attained highest classification accuracy in very less time.

1.5 Future scope

If the large amount of clinical data, pathological data, genomic data and images of mammogram are available for Indian women, we can design the computer aided expert system using deep learning techniques. Such expert system can assist the medical professionals to predict the breast cancer in early stage and may reduce the laboratory cost for further investigation.

Chapter 2

Preliminaries

This chapter provides some basic definitions and theorems which are useful in understanding the concepts discussed in successive chapters.

2.1 Linear algebra

Linear algebra plays a requisite role in machine learning due to vectors availability and several rules to handle vectors. Mostly classifiers or regressor problem in machine learning are tackle by linear algebra.

1. **Quadratic form [10]:**
   Let $A = [a_{ij}]$ be $n \times n$ symmetric matrix and $X = \big(x_1, x_2, ..., x_n\big)^T \in \Re^n$. The Quadratic form $Q : \Re^n \to \Re$ defined as:

$$Q(x) = x^T Ax = \big[x_1, x_2, ..., x_n\big] \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$= \sum_{i,j=1}^{n} a_{ij} X_i X_j; a_{ij} = a_{ji}$$

   Where, A is called matrix of the quadratic form.

2. **Definite and Semi definite Quadratic form [10]:**
   Let $Q(x) = x^T AX$ be a quadratic form, where A is symmetric matrix is said to be

   (a) Positive definite if
   $x^T AX > 0; \forall x \neq 0$
   $x^T AX = 0; \forall x = 0$

   (b) Positive semi definite if
   $x^T AX \geq 0; \forall x \neq 0$
   $x^T AX = 0; \forall x = 0$

   (c) Negative definite if
   $x^T AX < 0; \forall x \neq 0$
   $x^T AX = 0; \forall x = 0$

   (d) Negative semi definite if
   $x^T AX \leq 0; \forall x \neq 0$
   $x^T AX = 0; \forall x = 0$

3. **Moore Penrose Pseudo inverse:**
   Let A be an $m \times n$ matrix, then the Moore penrose pseudo inverse of A is

denoted by A^T and defined as $A^\dagger = \big(A^T A\big)^{-1} A^T$ and it holds the following properties.

(a) $AA^\dagger A = A$

(b) $A^\dagger A A^\dagger = A^\dagger$

(c) $\big(A^\dagger A\big)^T = A^\dagger A$

(d) $\big(AA^\dagger\big)^T = AA^\dagger$

4. **Eigen vector and eigen value [10]:**
   Let A be a $n \times n$ matrix. A eigen vector of a matrix A is a nonzero vector X such that $AX = \lambda X$ for some scalar λ. λ is called eigen value of A. If there is a non-trivial solution of X of $AX = \lambda X$; such X is called eigen vector corresponding to λ. λ is an eigen value of an $n \times n$ matrix A if and only if $\big(A - \lambda I\big)X = 0$

5. **Hyperplane [10]:**
   A hyperplane $H \in \Re^n$ is consist of $\big(x_1, x_2, ..., x_n\big)$ points which satisfy a linear equation $a_1x_1 + a_2x_2 + ... + a_nx_n = b$. Where, the vector $u = [a_1, a_2, ..., a_n]$ of coefficient is non-zero. In $\Re^n$ it is line, in $\Re^3$ it is plane and in higher dimension it is called as hyperplane.

6. **Gram matrix [19]:**
   Gram matrix is matrix that contains the evaluation of the kernel function on all pairs of training set. Mathematically it represents as:
   Let $X \in \Re^n$ be non-empty and function $k : X \times X \to \Re$ (or $\mathbb{C}$) is given where, $\big(x_1, x_2, ..., x_n \in X\big)$. Then $n \times n$ matrix with elements $K_{ij} = k\big(x_i, x_j\big)$ is called Gram matrix or kernel matrix of k w.r.t. $x_1, x_2, ..., x_n$.

2.2 Optimization

1. **Hessian matrix [99]:**
   A square matrix of second ordered partial derivatives of a scalar function f is known as the Hessian matrix. It is used in linear algebra and for calculating local maxima or minima points. Let H be a Hessian matrix of the function $f : \Re \to \Re$, where all second order partial derivatives of f exist and are continuous throughout domain and the function is $f\big(x_1, x_2, ..., x_n\big)$. Then Hessian matrix is defined as:

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial {x_1}^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial {x_2}^2} & \cdots & \frac{\partial^2 f}{\partial x_2 x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 f}{\partial x_n x_1} & \frac{\partial^2 f}{\partial x_n x_2} & \cdots & \frac{\partial^2 f}{\partial {x_n}^2} \end{bmatrix}$$

2. **Karush-Kuhn-Tucker (KKT) condition [99]:**
   Consider optimization problem:

$$\begin{aligned} \min_{x \in \Re^n} \quad & f(x) \\ \text{subject to constraints,} & \\ g_i(x) \leq 0; i = 1, 2, ..., m & \\ h_j(x) = 0; j = 1, 2, ..., r & \end{aligned}$$

   Define the general Lagrangian,

$$L\big(x, \alpha, \beta\big) = f(x) + \sum_{i=1}^{m} \alpha_i g_i(x) + \sum_{\beta_j}^{h_j(x)}$$

   where $\alpha and \beta$ are Lagrange's multipliers. The Karush-Kuhn-Tucker (KKT) condition are as follows:

$$\begin{aligned} \frac{\partial}{\partial x_i} L\big(x^*, \alpha^*, \beta^*\big) = 0; i = 1, 2, ..., n (\text{Stationary}) \\ \frac{\partial}{\partial \beta_i} L\big(x^*, \alpha^*, \beta^*\big) = 0; i = 1, 2, ..., r (\text{Stationary}) \\ \alpha_i^* g_i\big(x_i^*\big) = 0; i = 1, 2, ..., m (\text{Complementary slackness}) \\ g_i(x^*) \leq 0; i = 1, 2, ..., m (\text{Primal feasibility}) \\ \alpha_i^* \geq 0; i = 1, 2, ..., m (\text{Dual feasibility}) \end{aligned}$$

2.3 Functional analysis

1. **Inner product [19]:**
   Let V be a vector space over the field $\mathbb{K} = \mathbb{C} or \mathbb{R}$. An inner product on a vector space V is a function $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{K}$, which satisfying following properties:

$\forall u, v, w \in V$ and $\alpha \in \mathbb{K}$

(a) Symmetry: $\langle u, v\rangle = \langle v, u\rangle$

(b) Additivity: $\langle u + v, w\rangle = \langle u, w\rangle + \langle v, w\rangle$

(c) Homogeneity: $\langle \alpha u, v\rangle = \alpha\langle u, v\rangle$

(d) Positivity: $\langle u, u\rangle = 0$

(e) Nondegenerate: $\langle u, u\rangle = 0$ if and only if $u = 0$

2. **Inner product space [19]:**
   let V over the field $\mathbb{K}$. A vector space V with an inner product i.e. $\big(V, \langle\cdot,\cdot\rangle\big)$ is called inner product space.

3. **Dot product [19]:**
   Let $f(\cdot) = \sum_{i=1}^{n} \alpha_i k\big(\cdot, x_i\big)$ and $g(\cdot) = \sum_{j=1}^{n} \beta_j k\big(\cdot, x_j{}'\big)$ be $\mathbb{K}$-valued functions. Then dot product between f and g is defined as:

$$\langle f, g\rangle = \sum_{i,j=1}^{n} \alpha_i \beta_j k\big(x_i, x_j{}'\big)$$

   Then dot product $\langle f, g\rangle$ is symmetric and positive definite.

4. **Norm [66]:**
   Let V be a inner product space over $\Re$ or $\mathbb{C}$. For $u \in V$ is a real number. The norm of u (or length) is defined as:

$$\|u\| = \sqrt{\langle u, v\rangle}$$

   A norm holds the following properties:

   (a) $\|u\| \geq 0$; $\forall u \in V$

   (b) $\|u\| = 0$; if an only if $u = 0$

   (c) $\|\alpha u\| = |\alpha|\|u\|$; $\forall \alpha \in \Re$, $\forall u, v \in V$

   (d) $\|\alpha u + v\| \leq \|u\| + \|v\|$; $\forall u, v \in V$

   A norm is defined as a metric or distance on V which is defined as:

$$d(u, v) = \|u - v\| = \sqrt{\big(u_1 - v_1\big)^2 + \big(u_2 - v_2\big)^2 + ... + \big(u_n - v_n\big)^2}$$

5. **Kernel function [19]:**
   In machine learning, Kernel is also known as "Kernel Trick". It is used to classify non-linear problem using linear classifier. Let X be a nonempty subset of $\Re^n$. A function $k : X \times X \to \Re$, such that $k\big(x, x^{'}\big)$ gives a real number giving the similarity between two patterns x and $x^{'}$ is called kernel function.

6. **Mercer's theorem [19]:**
   Let X be a closed subset of $\Re^n; n \in \mathbb{N}$. If $k : X \times X$ be a symmetric function i.e. $k(x, x^{'}) = k(x^{'}, x)$ where $x \in \Re^n$ then k to be a valid kernel call Mercer's kernel.
   For any finite set of points $x_i \in X$ and $\forall a_i \in \Re$, the necessary and sufficient condition is $\sum_{i,j=1}^{n} a_i ajk(x_i, x_j) \geq 0$ i.e. the corresponding kernel matrix k is symmetric positive semi definite.

7. **Reproducing kernel [19]:**
   Let real valued positive definite kernel is k and let $\mathcal{X} \in \Re^n$ be a non empty set. The non linear function $\phi : \mathcal{X} \to \Re^{\mathcal{X}}$, which is defined as, $\phi : x \to k(\cdot, x)$ and $\Re^{\mathcal{X}}$ be the space of functions from $\mathcal{X}$ to $\Re$, i.e. $\Re^{\mathcal{X}} = \{\phi : \mathcal{X} \to \Re\} \in \Re^{\mathcal{X}}$. Construct a vector space containing the images of input patterns under the mapping ϕ as: $f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i)$.

8. **Reproducing Hilbert Space [19]:**
   Let $\mathcal{X}$ be a non-empty set. $\Re^{\mathcal{X}}$ is a Hilbert space of functions: $F : \mathcal{X} \to \Re$, provided with the dot product and the norm. Let $k : \mathcal{X} \times \mathcal{X} \to \Re$ is a reproducing kernel, i.e. $\langle f, k(x, \cdot)\rangle = f(x); \forall f \in \mathcal{H}$ and k span $\Re^{\mathcal{X}}$, then $\Re^{\mathcal{X}}$ is called Reproducing Hilbert Space.

2.4 Python preliminaries

To implement all mathematical algorithms, we have used python programming language. We have used different python packages to build Kernel Based models. Following python libraries have been used in our programming.

1. **NumPy (NUmerical PYthon) [92]:**
   NumPy is a library for the python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. It is used for scientific computing are the application of machine learning and deep learning.

```
Import numpy as np
```

2. **SciPy (Scientific Python) [92]:**
   It is used for scientific computing and technical computing. It contains modules for optimization, linear algebra, integration, interpolation, special function and other tasks common in science f engineering.

```
Import scipy as sp
```

3. **Pandas [92]:**
   It is an open source data analysis library for providing easy-to-use data structures and data analysis tools. It help to perform data analysis and data manipulation in Python language.It offers data manipulation and data operation for numerical tables and time series. It provide an easy way to create, manipulate and wrangle the data.

```
Import pandas as pd
```

4. **Matplotlib [92]:**
   It is plotting library for creating static, animated and interactive visualization in python. It offers endless charts and customization from histograms to scatter plots. It also offers away of colors, themes palettes and other options to customize and personalize plots.

```
From matplotlib import pyplot as plt
```

5. **seaborn [92]:**
   It is based on Matplotlib library which is use for data visualization. It is use for making statistical graphics in python. Also, it integrates closely with pandas data structures. It is more comfortable in handling Pandas data frames.

```
Import seaborn as sns
```

6. **Scikit-learn (Sklearn) [92]:**
   One of the most useful library namely Sklearn has been used for building machine learning model. It contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimension reduction.

(a) To standardized the features of the data set by removing the mean and scaling them to unit variance, standardization is used. It is calculated as $z = \frac{x-\mu}{\sigma}$. Where, μ is mean of the training data set and σ is standard deviation of the training data set.

```
1        From skelearn import standardscalar
2
```

(b) PCA or ICA is used to reduce the dimensionality without losing information from any features and speed up the learning algorithm with lower dimension.

```
1        From sklearn.Decomposition import PCA
2        From sklearn.Decomposition import ICA
3
```

(c) To split train-test data set to K consecutive folds, this package is used. Each fold is then used once as a validation while the 3-1 remaining folds from the training set.

```
1        From sklearn.Model-selection import Kfold
2
```

(d) Train-test package is used to divide data randomly into train-test data set.

```
1        From sklearn.model_selection import
   train_test_split
2
```

(e) To evaluate the performance of the model such as accuracy, recall, precision, F-Score for classification, confusion matrix is useful.

```
1        From sklearn.metrics import confusion matrix
2
```

(f) To calculate the classification score and report of the machine learning model following is used.

```
1        From sklearn.metrics import accuracy_score
2        From sklearn.metrics import classification_report
3
```

(g) To binarize labels in one vs all following preprocessing package is used.

```
1        From sklearn.preprocessing import label_binarize
2
```

(h) To measure the quality of the output of the classification model ROC and AUC curv is used.Roc curves typically feature true positive rate on the Y-axis and false positive rate on the X-axis.

```
From sklearn.metrics import roc_auc_curve
```

(i) To find the validatiton score of the machine learning model which calculate the average over cross validation folds and following package is used.

```
From sklearn.model_selection import
cross_val_score
```

(j) Support vectore Machines (SVMs) are a set of supervised learning methods used for classification, regression and outlier detection.

```
From sklearn import svm
```

(k) To build the neural network model this package is used.It implements a Multi-Layer-Perceptron (MLP) algorithm that trains using Back-propagation.

```
From sklearn.neural_network import MLPClassifier
```

2.5 Machine learning:

1. **Confusion matrix:**
   A confusion matrix is $n \times n$ matrix which is used to evaluate the performance of the classification model. There n is number of target classes. Matrix comparison are made between actual target values by machine learning model. It is defines in Table 1.

2. **Accuracy:**
   Accuracy is one of the most important performance measure for evaluating classification model which is define as :

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \times 100\%$$

3. **Precision:**
   Precision is also known as true positive rate. It measures that, among all

positive predicted samples how many samples are actually positive and it is defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\%$$

4. **Recall:**
   Reall measure that, among all the samples how many of that actually positive were found. It is defined as:

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\%$$

5. **F-score:**
   It measures model's efficiency and used to evaluate binary classification model. For imbalanced data set, if classification accuracy is measured according to accuracy, then the classifier can predict the value of the majority class for all predictions and achieve high classification accuracy which is not correct. This drawback can overcome by evaluating the classification accuracy using F-score. F-score is balance between Precision and recall. It is harmonic mean of precision and recall. It is defined as:

$$F\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\%$$

6. **Matthews's correlation coefficient (MCC):**
   MCC measure the difference between the predicted values and actual values

Table 2.1: Confusion Matrix

		Actual class	
		Positive (P)	Negative (N)
Predicted class	Positive (P)	TP	FP
	Negative (N)	FN	TN

*Terms:
TP: Number of correctly classified data from positive class
FP: Number of wrong classified data from positive class
FN: Number of wrong classified data from negative class
TN: Number of correctly classified data from negative class

form the confusion matrix.

$$MCC = \frac{\big(TP \times TN\big) - \big(FP \times FN\big)}{\sqrt{\big(TP + FP\big)\big(TP + FN\big)\big(TN + FP\big)\big(TP + FN\big)}}$$

7. **Kappa Statistic:**
   Cohen's kappa is more informative than overall accuracy when working with unbalanced data. Also it is consider as classification accuracy.

$$k = \frac{\text{Accuracy} - p_e}{1 - p_e}$$

8. **ROC-AUC curve:**
   Receiver Operator Characteristic (ROC) curve is a graph showing the performance of a classification model at all classification thresholds. The curve plots False-positive rate on x-axis vs true positive rate y-axis. ROC is as probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes.

9. **Universal approximation theorem [45]:**
   Let $\phi(\cdot)$ be a non constant, bounded and monotonically increasing continuous function. Let I_n be the n-dimensional unit hypercube [0,1]^n. Then $\forall f \in C(I_n)$ and $\forall \epsilon > 0, \exists p \in N$, set of real constants, $\alpha_j, \theta_j \in R, w_{ij}$, where $i = 1, 2, ..., n$ and $j = 1, 2, ..., m$ such that $F(x) = \sum_{j=1}^{m} \alpha_j \phi\Big(\sum_{i=1}^{n} w_{ij}^T x_i - \theta_j\Big), x \in I_n, w \in \Re^{n \times m}$ as an approximation of function $f(\cdot)$ independent of ϕ, i.e. for $x \in I_m, |F(x) - f(x)| < \epsilon, \forall x \in I_m$.

Chapter 3

The Ultimate Kernel Machine Based on Support Vector Machines

This chapter discusses Support Vector Machines, a classic kernel-based supervised learning technique, for classifying breast tumours as malignant or benign. Several kernel functions, including polynomial, linear, and Gaussian, are employed in breast tumour identification, and their impact on classification is examined through the use of several different metrics. Using Principal Component Analysis and k-fold cross validation, a cost-effective kernel-based model is created for breast cancer diagnosis. The WDBC and WBC data collection has been utilized to verify the accuracy of the model. In Section 3.1, a broad introduction regarding Support Vector Machines (SVM) and its applications in diverse fields is provided. The mathematical foundations of SVM are laid up in Section 3.2. Principal Component Analysis is used for feature reduction which is explained in detailed in section 3.3. In section 3.4 k-fold Cross Validation is explain. Experiments with the suggested SVM are discussed in section 3.5, and the chapter concludes with a summary in section 3.6.

3.1 Introduction

The support vector machine is based on the Vapnik Chervonenkis (VC) theory of statistical learning and effectively executes structural risk minimization. The SVM algorithm was created in 1963 by Vladimir N. Vapnik and Alexey Ya Chervonenkis. Later, in the 1960s, Boser, Cortas, and Vapnik created the supervised learning SVM at AT &T Bell Laboratories [30].

Over sixty years ago, in 1936, R.A. Fisher proposed the first method for pattern recognition. Moreover, Fisher suggested a linear decision function in the event that the two distributions are not normal [39]. An optimal quadratic decision function was found when two populations with standard distributions were considered.

The study of perceptrons, or neural networks, was initiated by Rosenblatt in 1962. The associated decision rule for building linear hyper planes was thus developed by Rosenblatt. There were, however, two issues that arose at this time: one of them is: a conceptual one, requiring the construction of a generalised separating hyperplane; and second is: a technical one, involving the computational interpretation of high-dimensional space. By creating the linear decision function with the largest margin between the two classes' vectors, Vapnik was able to determine the generalised optimal separation hyperplane in 1965 [20] [30].

In 1992, Boser Guyon and Vapnik found a solution to the technical problems that had been preventing them from properly handling the high-dimensional feature space. They came up with the algorithm to increase the margin that existed between the training data set and the decision boundary [20]. They suggested nonlinear classifiers that might be created by applying the kernel method to the maximum margin hyperplane. Since the data are not linearly separable, Cortos and Vapnik established the idea of soft margin in 1993 [123].

SVM classifier for non-linear data was developed by Vapnik and Chervonenk with the assistance of statistical learning theory. In its most basic form, support vector machines (SVM) is a binary categorization strategy; however, it may also be used for multi-class classification. The primary goal is to reduce the amount of computational work required to deal with high-dimensional data. For the purpose of classification, a support vector machine (SVM) builds a hyperplane or set of hyperplanes in a higher-dimensional space. The goal of the Support Vector Machine (SVM) algorithm is to locate a hyperplane in an N-dimensional space that can classify the data points in a separate manner. Utilizing Kernel techniques for the categorization of non-linear data. The efficiency of SVM is quite high. The kernel-based support vector

machine does not deal with the higher dimensional space directly, but rather it is dependent on the dot product of the input. Any non-linear function, or kernel, that can transform the data into a higher dimensional feature space and separate the data by determining the ideal border between the available outputs can be used in place of the dot product. SVM performs well with high-dimensional data because the data are automatically regularised and it prevents over-fitting with high-dimensional data. This is one of the reasons why SVM is used in many fields, including data classification, facial expression classification, text classification, speech recognition, and many more.

SVM is one of the best learning algorithms because it avoids overfitting, generalises classification results well, and works well in high-dimensional spaces using a variety of kernel functions. Thus, it is relevant in the medical field, particularly in the classification of diseases. Kourou et al. conducted research on a variety of machine learning approaches and conducted experiments using the WDBC and WBC data sets [65]. In the study by Huang et al., an SVM classifier was built with a variety of kernels for the purpose of breast cancer classification [49]. The researchers found that using an RBF kernel in conjunction with a feature selection strategy such a genetic algorithm resulted in the highest accuracy. The hybrid method was proposed by Liu et al. for the categorization of breast cancer [71]. Shravya et. al. achieved 92.7% classification accuracy with the utilisation of logistic regression, SVM, and k-nearest neighbour [110]. In addition, it is utilised in the fields of bioinformatics, medication development [24], the diagnosis of diabetes [67], the forecasting of electrical load [47], pattern recognition, and image processing. Additionally, it is helpful in the detection of spam and the diagnosis of faults [116], among other applications.

The Vapnik-Chervenenkis (VC) theory and the Structural Risk Minimization (SRM) premise are the foundations of the Support Vector Machine (SVM). Its purpose is to discover the optimal balance between minimising the training error and increasing the margin as much as possible. The fundamental concept of SVM is to locate or create a hyperplane that can partition the data into a certain number of categories. It is possible for the data to be separated in a linear or non-linear way. It is simple to find a hyperplane that separates the data linearly in the case of linearly separable classes. In the case of classes that are not linearly separable, it can be challenging to locate a hyperplane that can separate the data in a linearly. Therefore, in this scenario, the support vector machine (SVM) maps the input into a higher dimensional feature space using kernel methods so that it can be linearly separated.

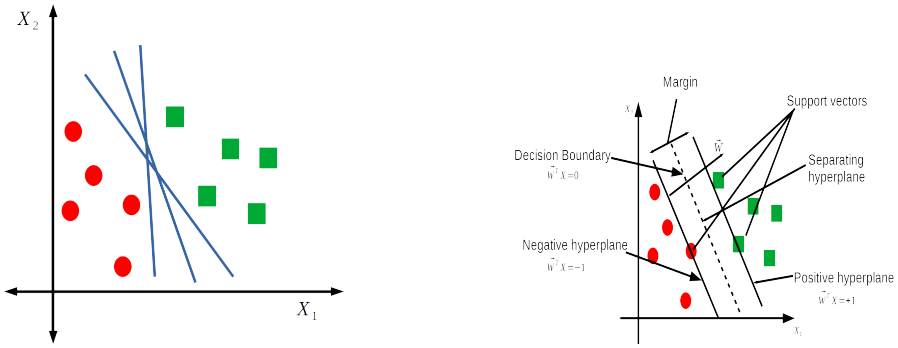
3.2 Mathematical formulation

3.2.1 Linearly Separable case - Hard margin

Consider two-class (binary) classification problem. For training data set, consider $\mathcal{T} = \{X_i, d_i\}$, where $i = 1, 2, \cdots, Q, X_i \in \Re^n$ and $d_i \in \{-1, +1\}$. Here, each input vector has number of component features. Each input set has corresponding target output d_i. Let decision function be a bipolar Signum function which is define as in eq. 3.1:

$$f(X) = sign\big(W \cdot X + b\big) \tag{3.1}$$

Here, " $\cdot$ " is a scalar or inner product. Hence, $W \cdot X$ can be also written as $W^T X$. The data are classified by the decision function based on whether the quantity's value is positive or negative.



(a) Which hyperplane? (b) Maximize margin

Figure 3.1: Maximal Margin Classifier

There are many hyperplane that can separates the two classes. The decision boundary should be as far away from the data of both classes as possible. The main goal of SVM is to find an optimal separating hyperplane that maximize the separating margin between the two classes of data. By linearly separability we can find an oriented hyperplane which is defined by a set of weights W and a bias b as shown in fig. 3.1. It separates the positive data from the negative ones. The general equation of the separating hyperplane is given by $W \cdot X + b = 0$ or $W^T X + b = 0$. The hyperplanes $W \cdot X^+ + b = +1$ for the closet point on the positive side and $W \cdot X^- + b = -1$ for the closet point on the negative side. Canonical hyperplanes are hyperplanes that pass through the point where $W \cdot X^+ + b = +1$ and $W \cdot X^- + b = -1$ is defined. The region that lies between these canonical hyperplanes is referred to as the Margin. Margin is defined as the distance between the dividing hyperplane and

the point that is closest to both classes. It is the data points that lie on the Margin that are referred to as the support vectors. In other words, the support vectors are the data points that are located on the hyperplane that are the nearest (or closest). Maximizing the margin of separation is similar to maximising the margin between hyperplanes. Therefore, $M = \frac{2}{|W|}$ is the entire margin between any two canonical hyperplanes. As a result, increasing the margin M to its highest possible level is the same as decreasing $|W|$. Therefore, it turns into a quadratic constrained optimization problem, in which the objective function is to be minimized while adhering to the constraints that are stated below in eq. 3.2:

$$\text{Minimize: } \frac{1}{2}\|W\|^2 \text{ subject to constraint } d_i\Big(W^T X_i + b\Big) \geq 1, \forall i \tag{3.2}$$

This quadratic constraint optimization formulation can be reduced to minimize of the Lagrangian Multiplier technique. It is define as sum of the objective function and the Q constraints multiplied by their respective Lagrangian Multipliers α_i Which is known as Primal Lagrangian function and defined as follows in eq. 3.3:

$$\text{Minimize: } L_p\big(W, b, \Lambda\big) = \frac{1}{2}\|W\|^2 - \sum_{i=1}^{Q} \lambda_i \Big[d_i(W^T X + b) - 1\Big] \tag{3.3}$$

Where, $\Lambda = \lambda_1, \lambda_2, ..., \lambda_Q$ is Lagrangian multipliers and $\lambda_i \geq 0$. The solution to the constraint optimization problem is determined by finding the critical points of the primal Lagrangian function and then by minimizing the primal Lagrangian function $L_p\big(W, b, \Lambda\big)$ with respect to primal variable W and b and maximize with respect to dual variable $\lambda_i \geq 0$. Differentiating L_p with respect to W and b and set it equal to zero we get following eq. 3.4 and eq. 3.5:

For W:

$$\frac{\partial}{\partial W} L_p\big(W, b, \Lambda\big) = W - \sum_{i=1}^{Q} \lambda_i d_i X_i = 0 \tag{3.4}$$

For b:

$$\frac{\partial}{\partial W} L_p\big(W, b, \Lambda\big) = 0 - \sum_{i=1}^{Q} \lambda_i d_i \tag{3.5}$$

Substitute this results into L_p i.e. in eq. 3.3 in order to eliminate W and b, it gives the Wolf Dual Lagrangian as follows in eq. 3.6 which is quadratic optimization problem with linear constraints.

$$L_D(\Lambda) = \sum_{i=1}^{Q} \lambda_i - \frac{1}{2} \sum_{i=1}^{Q} \sum_{j=1}^{Q} \lambda_i \lambda_j d_i d_j (X_i \cdot X_j) \tag{3.6}$$

Due to dual formulation, it must be maximized with respect to λ_i subject to the constraints as shown in eq. 3.7:

$$\sum_{i=1}^{Q} \lambda_i d_i = 0 \text{ and } \lambda_i \geq 0; \;\; i = 1, 2, ..., Q \tag{3.7}$$

Karush-Kuhn-Tucker (KKT) condition is the necessary and sufficient condition for convex optimization problem with linear constraints and also it governs the duality problem. By applying KKT condition to the Lagrangian function,$\lambda_i \big[d_i (X_i \cdot W + b) - 1 \big] = 0; \lambda_i \neq 0$. We get the decision function which classifies points based on the eq. 3.6. Eq. 3.6 and eq. 3.7 can be written as in the matrix form as follows:

$$\text{Maximize: } \; L_D(\Lambda) = \Lambda \cdot 1 - \frac{1}{2} \Lambda^T H \Lambda \;\; \text{s.t.c., } \;\; \Lambda \cdot D = 0 \;\&\; \Lambda \geq 0$$

Where, H is Hessian matrix having elements, $H_{ij} = d_i d_j (X_i X_j)$. Which is called quadratic programming optimization problem with linear constraints. This gives optimized Lagrangian multipliers $\hat{\Lambda} = (\hat{\lambda_1}, \hat{\lambda_2}, ..., \hat{\lambda_Q})^T$. Some observations for KKT condition:

1. The solution for minimizing $L_p(W, b, \lambda)$ with respect to W and b and subject to $\lambda \geq 0$ is the same as the solution of maximizing $L_p(W, b, \lambda)$ with respect to λ and subject to appropriate constraints.

2. The Lagrangian multipliers are non-negative.

3. For equality constraints:
   The differential of $L_p(W, b, \lambda)$ w.r.t. the Lagrangian multiplier is zero at the solution point.

4. For inequality constraints:
   Either Lagrangian multiplier is zero and constraint is satisfied independently or multiplier is non-zero and constraint is satisfied with equality.

From the KKT condition, for each point $\lambda_i \big[d_i (W \cdot X_i + b) - 1 \big] = 0$ or either $\lambda_i = 0$ or $\lambda_i (W \cdot X_i + b) = 1$. Where we have two conditions for λ_i's as follows"

1. The data point for which lagrangian multipliers $\lambda_i > 0$ are called support vectors. i.e. The data points which satisfy $d_i(W \cdot X_i + b) = 1$ are the support vectors.

2. If $\lambda_i = 0$ then data points are non-support vectors. Such point does not contribute to the separating hyperplane.

For the solution of the dual problem, consider λ^*, W^*, and b^* and we get,

$$W^* = \sum_{k=1}^{ns} \hat{\lambda_k} d_k X_k$$

Where, ns is number of support vectors and $b^* = \frac{1}{d_s} - W^* X_s$ which is evaluated from the complementary condition. However, b is calculated by averaging over all support vectors.

$$b^* = \frac{1}{ns}\left[\sum_{l=1}^{ns}\left(\frac{1}{d_l} - W^* \cdot X_i\right)\right]$$

Hence, after substituting value of W^* and b^* into eq. 3.1, the separating hyperplane or discriminant function is given by eq. 3.8. Eq. 3.8 consider only those data points which separated the negative and positive classes. and those data points are known as support vectors (ns).

$$f(x) = sign\left(\sum_{i=1}^{ns} d_i \hat{\lambda_i}(X \cdot X_i) + b^*\right) \tag{3.8}$$

If $f(x) = +1$ then X is classified as positive data point. and if $f(x) = -1$ then X is classified as negative data point.

3.2.2 Linear Non-separable case - Soft margin

SVM with hard margin is utilised in situations in which the data can be separated linearly and there are no instances of incorrect classification. On the other hand, if the data cannot be separated in a linearly, we have the option of using a more wide margin for classification in order to reach a higher level of generality. This means that it can be separated in a non-linear approach and that it permits some of the data points to be unclassified. Even when the data can be separated linearly, the margin may be so small that the model is over-fit or particularly sensitive to outliers.

This can happen even though the data can be separated linearly. In addition, to assist in the adaptability of the model, we may select a larger margin by utilising soft margin SVM. This would allow the model to be more flexible. because of this, in the scenario that the training data is not linearly separable and there is overlap between the classes. To avoid mis-classification of noisy data points, slack variable ξ_i can be added. Hence, new optimization problem can be reformulated as follows in eq. 3.9

$$\text{Minimize:} \quad f(W, \xi_i) = \frac{1}{2}\|W\|^2 + C\sum_{i=1}^{Q}\xi_i \quad \text{s.t.c.} \quad d_i(W \cdot X_i) + b \geq 1 - \xi_i \tag{3.9}$$

Here, a penalty value C is introduced for the points that cross the boundaries to take into account the mis-classification errors. Parameter C also controls the over-fitting issues. A soft margin is created within which a mis-classified data lies. The width of this soft margin is controlled by a penalty parameter C. Also C controls the weighting to make the $\|W\|^2$ small.

Now, Lagrangian of the given quadratic optimization problem in terms of primal variable is given as follows:

$$L_p = \frac{1}{2}\|W\|^2 + C\left(\sum_{i=1}^{Q}\xi_i\right) - \sum_{i=1}^{Q}\lambda_i\Big[d_i(W \cdot X_i + b) - 1 + \xi_i\Big] - \sum_{i=1}^{Q}\gamma_i\xi_i$$

Where, λ_i and γ_i are Lagrangian multipliers and $\lambda_i \geq 0$ and $\gamma_i \geq 0$.

Same as Hard margin case, from the KKT condition and at saddle points, partial derivatives w.r.t. primal variable vanishes. In both cases there is only one difference which is the upper bound on the Lagrangian multiplier λ_i

3.2.3 Non-linear separable case - Kernel trick

It is usually not possible to make a straight line of classification between the two groups. In the year 1992, Bernhard Boser, Isabelle Guyon, and Vladimir Vapnik presented a general concept of kernel trick to maximum – margin hyperplanes. Their goal was to extend the capabilities of the linear learning machine so that it could deal effectively with non-linear scenarios. When data points in the input space cannot be separated by a linear decision boundary, the points are projected onto a higher-

dimensional feature space. In this specific case, a non-linear kernel function is used in place of every dot product. The data points are moved from the input space to the higher-dimensional feature space using a non-linear function $\varphi : X \to \mathcal{H}$, where X is a non-empty set and $\mathcal{H}$ is a Hilbert space.

Data mapping from input space to higher dimensions feature space using non-linear functions like kernels is computationally intensive. When compared to other methods of transforming data into higher dimensions, kernel tricks are more time and cost effective. Non-linear kernel function is applied directly without calculating non-linear kernel function ϕ at each data point. $X \to \varphi(X) = (a_1\phi_1(X), a_2\phi_2(X), ..., a_n\phi_n(X))$. That can be formulated utilising an infinite number of feature variables. Therefore, rather than working in X-space, we are now working on $\mathcal{H}$. Using $\varphi(X)$ as the input variable instead of X, we apply the methods of the soft-margin classifier. Using this mapping, the discriminant function in the feature space can be represented as follows:

$$f(x) = sign\left(\sum_{i=1}^{ns} d_i\hat{\lambda}_i\phi(X)\cdot\phi(X_i)^T + b\right)$$

Where, $X_i^T X$ in the input space is represented as $\phi(X)\cdot\phi(X_i)^T$ in the feature space.

Hence, the optimal separating hyperplane is given as follows:

$$\sum_{i=1}^{ns} d_i\hat{\lambda}_i\phi(X)\cdot\phi(X_i)^T + b = 0$$

The functional form of the mapping $\phi(X_i)$ implicitly defined by the choice of kernel: $K(x_i, X_j) = \phi(X_i)^T\cdot\phi(X)$. The input values of X can be written in the form of a dot product $X_i \cdot X_j$. Hence, dot product is the operation which performs over X and in feature space, it becomes $\phi(X_i)\cdot\phi(X_j)$. In addition, the Kernel function ought to be Mercer's Kernel (insert cross reference), which means that it ought to be a positive semi definite function. The term "Reproducing Kernel Hilbert Space" refers to the feature space that is associated to a certain kernel [19]. A kernel function that, in a higher feature space, is equivalent to the dot product of two feature vectors is called the dot product kernel function. There are a variety of well-known Mercer's kernels like Fisher's Kernel, Graph kernel, Kernel Smoother, Polynomial kernel, Gaussian kernel, Linear kernel, etc. which are compatible with SVM.

Various Kernel functions are used in SVM [46]. The most common Kernels used in SVM are defined as follows.

1. Polynomial Kernel : $K(X_i, X_j) = (1 + X_i^T X_j)^p$, p is polynomial order.
2. Gaussian kernel : $K(X_i, X_j) = exp\left(\frac{||X_i - X_j||^2}{2\sigma^2}\right)$, σ is the parameter controlling the width of the Gaussian kernel.
3. Linear kernel: $K(X_i, X_j) = X_i^T X_j$

3.3 Principal Component Analysis

The dimensionality of a large data set can be reduced using an unsupervised method called principal component analysis. Large data sets are becoming more common and are often difficult to comprehend. Principal component analysis (PCA) is a statistical method that increases interpretability of large data sets by reducing the number of dimensions in the data set without diminishing generality or information. In order to achieve this goal, the initial variables are replaced by a new set that is referred to as the principle components (PCs) [54]. These principal components are not connected with one another. This is accomplished by including a number of new variables that are unrelated to one another in order to increase the variance. In 1901, Karl Pearson introduce Principal Component analysis as similar of the principal axis theorem in mechanics and further it was individually developed and named by Harold Hotelling in the 1930s [91] [48]. PCA has different goals like the extraction of the data's most significant feature, reduction of features while maintaining the information's integrity, in image compression, visualization of multi-dimensional data. The dimension of the original data can be reduce dimension to k, if original data consist of n dimension such that $k \leq n$. Following are the steps involved in applying PCA:

1 Standardization

This stage standardises the continuous initial variables to ensure each contributes equally to the evaluation. If the initial variable ranges differ significantly, the larger ranges will dominate the smaller ranges, causing bias in the outcomes. This issue can be avoided by converting the data to comparable scales. Mathematically, Subtract the mean from each variable's value and divide by the standard deviation i.e. $x_{new} = \frac{x-\mu}{\sigma}$; μ=mean and σ = Standard deviation. Using this standardisation method, the values of all the features will be converted to the same scale.

2 Compute Covariance matrix

The purpose of generating a covariance matrix for the full data set is to examine the dispersion of the input data's features (variables) relative to the mean. There can be redundancy in data sets since features (variables) are often highly associated with each other. Covariance matrix helps to find these correlations. for 4 samples and 4 features, covariance can be written as follows:

$$\begin{bmatrix} var(f_1) & cov(f_1, f_2) & cov(f_1, f_3) & cov(f_1, f_4) \\ cov(f_2, f_1) & var(f_2) & cov(f_2, f_3) & cov(f_2, f_4) \\ cov(f_3, f_1) & cov(f_3, f_2) & var(f_3) & cov(f_3, f_4) \\ cov(f_4, f_1) & cov(f_4, f_2) & cov(f_4, f_3) & var(f_4) \end{bmatrix}$$

3 Compute eigen vector and eigen value

Eigen value and Eigen vectors are computed from the covariance matrix. The new variables known as Principal Components are created by linear combination of the initial variables and they uncorrelated.

4 Sort Eigen values and their corresponding eigenvectors

In order to determine the significant number of PCs, sort the Eigen values from highest to lowest. Additionally, the dimensionality will be reduced without significant data loss. To that end, the individual components can be treated as independent variables.

5 Choose k eigenvalues and form a matrix of corresponding eigenvectors to decide which PCs to keep

The low-significance eigenvectors are discarded, and a "feature vector" matrix is calculated from the high-significance eigenvectors. Eigenvectors of the components we decide to retain are stored in a column matrix called a feature vector.

6 Transform the original matrix/data

Using Eigen vectors of the covariance matrix feature vector is formed. Data are reorient from the original axes to the ones which is known as principal components. Data is transformed using following formula: $Transform = Featurevector * topkEigenvectors$

3.4 k-fold cross validation

Models' ability to accurately generalise or perform on unknown data can be tested with the help of the statistical approach of cross-validation, often known as the re-

sample technique. Here, single k refers the number of groups or folds into which a specific data sample is ti be divided is indicated by single parameter. Initially, data are randomly partitioned with equal size into k fold or groups, say $D1, D2, ..., DK$. Training and testing data are performed k times. The first iteration is trained on groups $D2, D3, ..., Dk$ and tested on $D1$. Then the second iteration is trained on groups $D1, D3, ..., Dk$ and tested on $D2$. Likewise, this procedure will continue for k-folds. Advantage of this method is that all samples in data set are eventually used for training set as well as testing set. The general procedure is as follows:

1. Shuffle the data set randomly.
2. Split the data set into k-fold
3. For each folds:
   (a) Consider one of the fold as a test data set.
   (b) Consider remaining fold as a train data set.
   (c) Fit a model on the training set and evaluate it on the testing set.
   (d) Keep the evaluation score and repeat the process for k- fold.
4. Summarize the evaluation score of each mode by $E = \frac{1}{k} \sum_{k=1}^{10} E_k$.

3.5 Experimental work and results

SVM based Classifier is developed to classify breast cancer data set into Benign (B) and Malignant (M) classes respectively. SVM classifier is designed by employing linear, Gaussian and polynomial kernels. Additionally, PCA is utilized to diminish the dimensions. We first accomplished the mean normalization step, before beginning the PCA process. The feature vector dimensions of the WBC and WDBC data set were significantly reduced up-to 3 without loosing any information of data and retained 99% of discrepancy. By taking $k = 10$ folds. k-fold cross validation technique is utilized to split data into train-test set. Where, all data are shuffle randomly and split into 10 folds. Each set serves as both a training set and a testing set and both are performed 10 times. after a model has been fitted to the training data, then it is tested on the test data. Our investigation involve the following SVM training scenarios for the WBC and WDBC data sets. WBC and WDBC data set are explained in detail in Appendix-A.

1. SVM without PCA and without k–fold CV
2. SVM with PCA only
3. SVM with k–fold CV only
4. SVM with PCA and with k–fold CV

In the first two scenarios, the data set are randomly divided into the training set and the testing set. In the next two scenarios, the k-fold cross-validation method is utilised for the purpose of partitioning data set into training and testing sets. There are three distinct kernels utilised in each each case. Calculations of the confusion matrix and several other accuracy metrics are performed in order to determine the efficiency of the classification. In each scenario, the duration of time required to complete all computations in each case is determined to check efficacy of the classifiers. All experiments are carried out using Python programming language.

3.5.1 Experiment 1: Experiments using WDBC data set

To begin with, WDBC data set would be normalized in order to minimize and remove any redundant data. Experiments are carried out with the different values of the penalty parameter C, learning rate γ and three different kernels. The highest accuracy is obtained for polynomial kernel with degree 3, $C = 100$ and $\gamma = 0.0006$. The accuracy obtained is 96.63% and time required for training was 0.63 second. The comparison of accuracy and time obtained for 3 different kernels are depicted in table 3.1.

Table 3.1: Comparison of Accuracy and Time for different kernels using SVM

Kernels	Measures	Without PCA or k-fold CV	With PCA only	With k-fold CV only	With PCA and k-fold CV
Polynomial	Accuracy(%)	92.98	91.23	96.43	96.63
	Time (sec)	0.35	0.39	0.04	0.63
RBF	Accuracy(%)	59.69	92.98	76.79	96.43
	Time (sec)	0.38	0.50	0.39	0.43
Linear	Accuracy(%)	92.98	92.98	96.43	96.43
	Time (sec)	0.50	0.38	0.39	0.80

Graphically, accuracy and time comparison are plotted which is exhibited in fig. 3.2 and fig. 3.3.

Figure 3.2: Comparison of Accuracy for different kernels using SVM for WDBC dataset

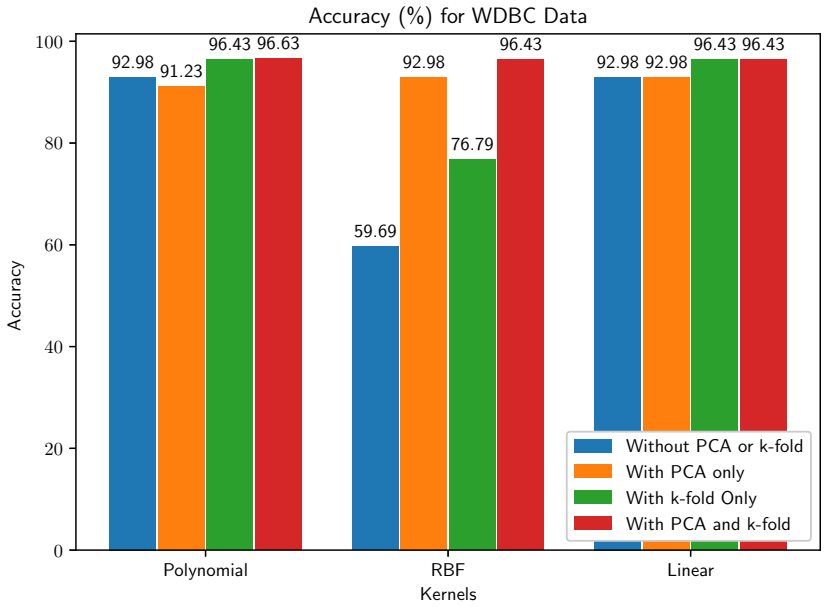


Figure 3.3: Comparison of Time for different kernels using SVM for WDBC dataset

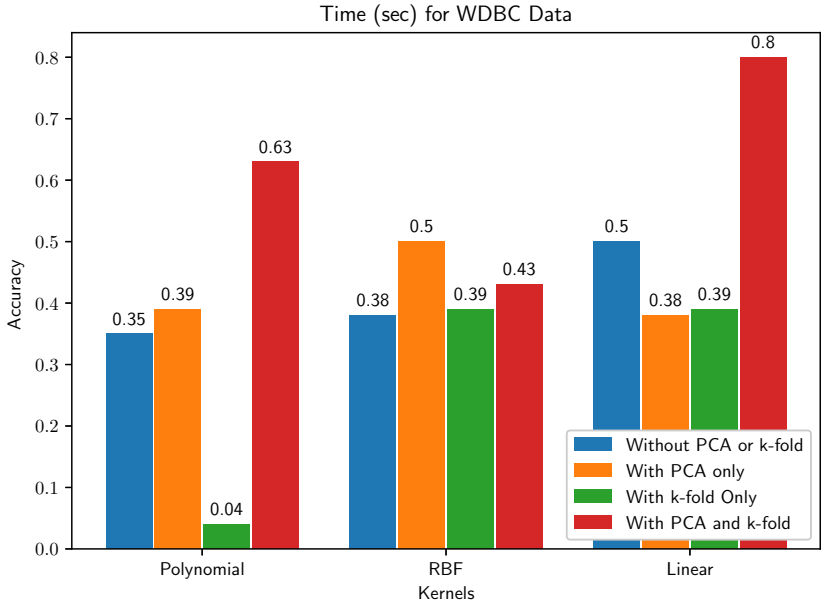


Table 3.2 represents the confusion matrix for the WDBC data set. which was derived through experiments using a case that had SVM along with PCA and k-fold CV. The results of an analysis of the confusion matrix are shown in table 3.3 along with a measure of performance.

We compare the results of our studies on the WDBC data set with the results obtained by other authors as shown in table 3.4. When compared to other authors,

Table 3.2: Confusion matrix for WDBC dataset using SVM for different kernels

Polynomial Kernel			RBF kernel			Linear Kernel		
Actual value	Predicted value		Actual value	Predicted value		Actual value	Predicted value	
	M	B		M	B		M	B
M	40	3	M	41	2	M	43	0
B	2	11	B	2	11	B	3	10

Table 3.3: Performance measures of SVM for WDBC dataset

Measures	Polynomial Kernel	RBF Kernel	Linear Kernel
Sensitivity	0.93	0.95	1.00
Specificity	0.85	0.85	0.77
F-Score	0.91	0.93	0.95

we find that cite Mert2011 has the highest accuracy 94.40%. However, 96.63% classification accuracy was achieved for the same data set in 0.63 second by proposed SVM classifier with polynomial kernel.

Table 3.4: Comparison of classification accuracy of other papers with our experiments for WDBC dataset

Authors	Year	Methods	Classification Accuracy
[76]	2011	SVM (Quad), 25% test data SVM (RBF), 25% test data	94.40% 93.70%
[84]	2010	SVM (Poly.), 40% test data SVM (RBF), 40% test data	92.62% 93.72%
[115]	2010	PSO and SVM QPSO and SVM	93.52% 93.06%
[110]	2019	SVM	93.70%
Present study	2020	SVM (Linear) SVM (Polynomial) SVM (RBF)	96.43% 96.63% 96.43%

3.5.2 Experiment 2: Experiments using WBC data set

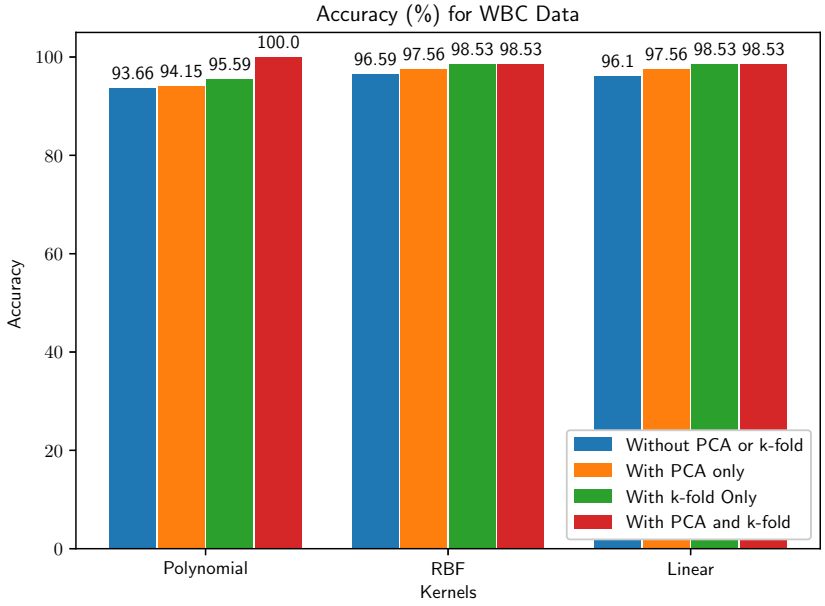
Experiments are carried out in the same manner using the WBC data set. The results of a comparison of the accuracy and time required by three distinct kernels for this data set are presented in table 3.5. Also, graphical representation of accuracy and time are depicted in fig. 3.4 and fig. 3.5. For WBC data set, we tried out a variety of combinations, including a wide range of values for the penalty parameter

C, learning rate γ and using three distinct kernels. The polynomial kernel with 3 degree and $C = 1.0$ yeilds the maximum 100% accuracy in just 0.03 second.

Table 3.5: Comparison of Accuracy and computation time for different kernel using SVM

Kernels	Measures	without PCA or k-fold CV	With PCA only	With k-fold CV only	With PCA and k-fold CV
Polynomial	Accuracy(%)	93.66	94.15	95.59	100.00
	Time (sec)	0.34	0.01	0.78	0.03
RBF	Accuracy(%)	96.59	97.56	98.53	98.53
	Time (sec)	0.42	0.01	0.43	0.03
Linear	Accuracy(%)	96.10	97.56	98.53	98.53
	Time (sec)	0.33	0.06	0.48	0.02

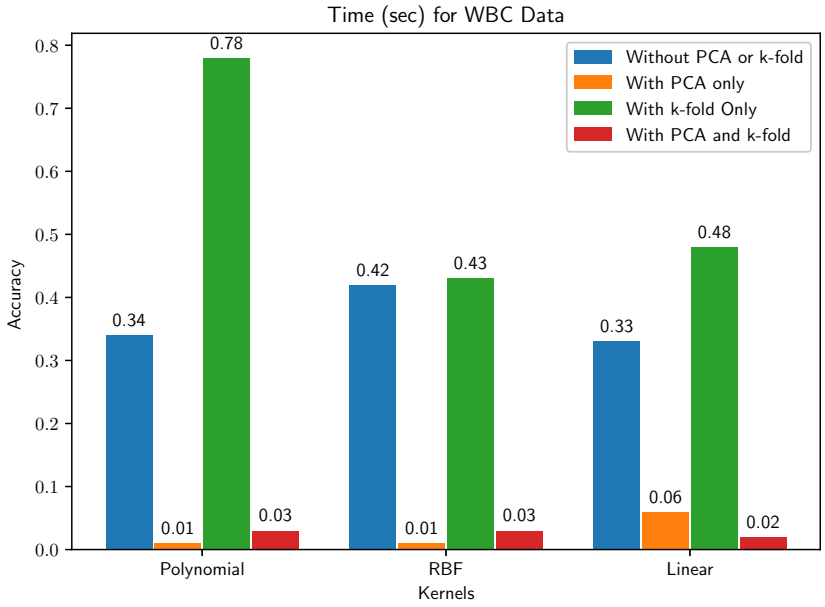
Figure 3.4: Comparison of Accuracy for different kernels using SVM for WBC dataset



The confusion matrix for the various kernels are presented in table 3.6. The performance measures of the SVM for the WBC data set are displayed in table 3.7. Table 3.8 represents a comparison of the classification accuracy achieved by proposed model verses achieved by other authors using WBC data set. Among all the authors, Huang et.al. (2017-add citation) have achieved 98.28% accuracy. However, proposed SVM model achieved 100% classification accuracy in 0.03 second with polynomial kernel.

The detailed information of the both data set are given in the Appendix. Also,

Figure 3.5: Comparison of Time for different kernels using SVM for WBC dataset



above all computation is carried out using Python programming and it is given in the Appendix.

Table 3.6: Confusion matrix for WBC dataset using SVM for different kernels

Polynomial Kernel			RBF kernel			Linear Kernel		
Actual value	Predicted value		Actual value	Predicted value		Actual value	Predicted value	
	M	B		M	B		M	B
M	55	0	M	54	1	M	54	1
B	0	13	B	0	13	B	0	13

Table 3.7: Performance measures of SVM for WBC dataset

Measures	Polynomial Kernel	RBF Kernel	Linear Kernel
Sensitivity	1.00	0.98	0.98
Specificity	1.00	1.00	1.00
F-Score	1.00	0.99	0.99

Table 3.8: Comparison of classification accuracy of other papers with this current experiments for WBC dataset

Authors	Year	Methods	Classification Accuracy
Authors	Year	Methods	Classification Accuracy
[90]	2010	SVM	96.33%
[70]	2003	SVM (Polynomial) SVM (RBF)	96.71% 97.07%
[41]	2005	SVM	97.51%
[49]	2017	SVM ensembles (RBF) + GA SVM (Linear) + GA SVM ensembles (Linear) + GA	98.28% 96.85% 96.57%
Present study	2020	SVM (Polynomial) SVM (RBF) SVM (Linear)	100.00% 98.53% 98.53%

3.6 Conclusion

Different kernels, PCA and k-fold CV are used to conduct a comparative evaluation of SVM-based classifiers. Proposed models are validated using WDBC and WBC data sets. Feature reduction in both data set is done so that 99% of variance is preserved or retained. Evaluation of how various kernels give best classification accuracy, performances and time has been compared.

The classification accuracy is significantly improved after employing PCA for feature reduction and k-fold CV to split data into train-test set. By selecting the suitable values for the penalty parameter C and learning rate γ.

For WDBC data set, Polynomial kernel achieved the maximum accuracy of 96.63%. For WBC data set, Gaussian and Polynomial kernel achieved 98.53% and 100% classification accuracy respectively. It is found that, combining SVM with PCA and k-fold CV considerably decreases training time. The significance and efficiency of proposed model's finding have been established and validated by comparative analysis. According to proposed findings, a computer-aided diagnostic system that has been thoughtfully developed can be of assistance to medical professionals in making quick decisions and reducing the risk of making an incorrect diagnosis.