Design of Primal Soft-margin SVM Leveraging Piecewise Linear Approximation based Linear Programming Technique

[1] Shital Solanki, [2] Dr. Ramesh Prajapati

[1] Research Scholar, Gujarat Technological University, Ahmedabad, Gujarat, India.
[2] Professor, CE/IT Department, Shree Swaminarayan Institute of Technology, Gandhinagar, Gujarat, India.

Abstract: -This research introduces a new approach for training soft-margin Support Vector Machines (SVMs) using the primal formulation. The method, called soft-margin Piecewise Linear Approximation based SVM (Soft-margin PLA-SVM), streamlining the optimization of soft-margin SVM hyperparameters in linear programming framework using well-known GUROBI Optimizer Solver. It eliminates the need for an initial hyperparameter guess and uses an adaptable initial search domain. The study uses the Wisconsin Breast Cancer Original dataset from UCI machine learning to validate the effectiveness of proposed soft-margin PLA-SVM. Comparative analysis shows that proposed PLA-SVM outperforms other classifiers in terms of training speed, accuracy, precision, and ROC-AUC scores. The scalability and computational efficiency of soft-margin PLA-SVM make it suitable for high-dimensional and large-scale datasets. The research demonstrates the effectiveness of the primal perspective in solving the soft-margin SVM design problem.
Keywords: Breast cancer, Classification, GUROBI solver, Piecewise Linear Approximation, Support Vector Machine, Training and testing. Linear Programming.

1. Introduction

Support vector machine (SVM) [1] is a widely used machine learning method for classification and regression problems. By leveraging its ability to handle high-dimensional data and classify complex patterns, SVMs help in identifying and predicting breast cancer in its early stage. The Wisconsin Breast Cancer (WBC) dataset [2] represents a seminal contribution to breast cancer research. Over the years, countless researchers have turned to the WBC dataset as a trusted benchmark to predict breast cancer.

Recently, Shital and Ramesh introduced and validated a novel approach for designing linear Support Vector Machines (SVMs) [3]. This innovative method, termed Piecewise Linear Approximation Support Vector Machine (PLA-SVM), leverages a combination of piecewise linear approximation and separable linear programming optimization techniques to efficiently construct primal SVMs for linearly separable dataset. The linear PLA-SVM is validated on laboratory gas turbine engine giving a global solution through the utilization of mixed integer linear programming and branch and bound algorithms inherent to the GUROBI Optimizer solver [4].

In the present work, the linear PLA-SVM proposed by Shital and Ramesh [3] has been extended to design a soft-margin PLA-SVM that can deal with non-separable datasets. This innovative technique entails the transformation of complex, non-linear, and non-convex optimization problems into approximate linear programming instances through the application of separable programming concepts [5]. This approach has the potential to streamline the optimization process and enhance its feasibility. Nataraj and Makwana have introduced an adept and rapid strategy for formulating a QFT controller through the utilization of separable linear programming principles (referred to as the PLA-LP method) [6 - 9]. This intricate non-convex and non-linear conundrum inherent in QFT controller design is proficiently addressed by the Gurobi optimizer solver [4]. The inherently non-linear character of the soft-margin SVM's objective function has served as inspiration for us to construct an SVM optimization problem employing the concept of separable programming. Inspired by the growing trend of addressing the SVM problem in its primal form, we introduce a novel approach to solving the soft-margin SVM problem in primal formulation.

The proposed SVM is validated using the Wisconsin Breast Cancer (WBC) Original dataset from the UCI Machine Learning Repository [2]. The GUROBI Optimizer's interface with MATLAB [10] is employed to solve the proposed SVM optimization problem. GUROBI optimizer proficiently handles the piecewise linear model with SOS2 (Special Order Set Type 2) variables with free academic license.

The rest of this paper is organized as follows. Section 2 overviews related studies including those on primal support vector machines and breast cancer detection. Section 3 describes basics for developing proposed method. Section 4 presents the proposed methodologies in details. The experimental implementation of the proposed soft-margin PLA-SVM is presented in Section 5. Finally, Section 5 concludes the paper.

2. Related Works

In the SVM research domain, a predominant focus has been directed towards addressing the dual optimization problem, which involves the utilization of LaGrange multipliers associated with individual support vectors. One inherent challenge within the dual SVM formulation lies in the necessity to initialize the LaGrange multipliers with arbitrary values. Consequently, the solution outcome becomes contingent on these initial assignments of the LaGrange multipliers (α's). Despite their remarkable accuracy, SVMs have are preferred due its slow training time. Consequently, the demand for expeditious algorithms to resolve these challenges becomes imperative [11]. The appeal of primal approaches lies in their continuous reduction of the primal objective function. Notably, Suykens&Vandewalle and Fung & Mangasarian have undertaken the development and meticulous detailing of the primary form of LS-SVMs [12][13]. In contrast to the Quadratic Programming (QP) problem, they adeptly addressed a set of linear equations. A pivotal milestone was achieved by Keerthi et al. [15][16], as they were the first researchers to emphasize that the optimization of SVMs in their primal form is more prevalent than the dual form. O. Chapelle, in a significant contribution, exemplified the efficient training of both linear and nonlinear datasets using the primal SVM problem [17]. Zhizheng Li and Li further underscored the practicality of primal optimization in numerous real-world applications, where it directly minimizes the objective function [18]. Additionally, Qing Wu and Wanqing Wang pioneered the transformation of SVM's primal programming problems into smooth, unconstrained minimization problems, making them amenable to rapid resolution via the Newton-Armijo algorithm [19].

In the existing body of literature, we have encountered a limited number of approaches that employ linear programming and piecewise linear approximation methodologies in the design of Support Vector Machines (SVMs). The introduction of Support Vector Machines by Vapnik et al. [20] marked a significant turning point in breast cancer classification. The literature encompasses various statistical and machine learning methodologies employed in crafting breast cancer prediction models, including logistic regression, linear discriminant analysis, naïve Bayes, decision trees, artificial neural networks, k-nearest neighbors, and support vector machine (SVM) techniques. Notably, comparative studies investigating these methodologies have consistently revealed SVM's superior performance when contrasted with many of its counterparts.

Lavanya and Rani [21] analyzed feature selection with classification on three breast cancer datasets, employing the CART classifier for binary classification. Sivakami and Saraswathi [22] worked on breast cancer prediction using a DT–SVM hybrid model of decision tree and support vector machine to improve the accuracy of breast cancer prediction. McKinney et al. [23] proposed an AI-based system that outperformed human experts in breast cancer prediction using mammogram images. Akbulut et al. [24] conducted breast cancer classification using three machine learning models: GBM, XGBoost, and LightGBM. Among these, LightGBM demonstrated superior accuracy, achieving a score of 95.3%.

3. Some Preliminaries

In this section, background theories of piecewise linear approximation (PLA) techniques and primal soft-margin SVM have been discussed in brief.

3.1. Piecewise Linear Approximation using Separable Linear Programming

The proposed design approach is based on the idea that any nonlinear function can be closely approximated by a piecewise linear function with a high degree of precision [25]. Utilizing a piecewise linear approximation transforms a nonlinear problem into a linear one, enabling the application of linear programming techniques that are significantly simpler and more efficient than nonlinear methods [26]. The PLA theory introduced in [5] focuses on transforming non-linear, non-convex optimization problems into approximating linear programming problems through the concept of separable programming. The nonlinear function must be separable (i.e., a function of only one variable) in order to build a piecewise linear model.A function is said to be non-separable if it cannot be broken down into separate functions.

An inequality-constrained problem is said to be separable if the objective functions and/or constraints can be separated in the variables x_i as follows:

$$Min\ f(X) = \sum_{i=1}^{n} f_i(x_i)\ ,\quad n = number\ of\ variables \tag{1}$$

Subject to constraints,

$$C_j:\ \sum_{j=1}^{m} \sum_{i}^{n} g_{ji}(x_i) \leq\ b_j\ ,\ m = number\ of\ constraints \tag{2}$$

where b_j is a constant and $g_{ji}(x_i)$ is a j^th constraint for the ith variable.

A separable piecewise linear function is depicted in Figure 1, where P1, P2..., and Pk are the breakpoints—the places at which the function changes direction. A piecewise linear approximation (PLA) of the nonlinear function f(x) in Figure 1 is shown as by $\widetilde{f(x)}$. Here, $\widetilde{f(x)}$ is defined over the closed intervals $[x_k\ ,x_{k+1}]$, k = 1, …., K – 1, where the co-ordinates ($x_{k,}$ f ($x_{k,}$)), represent breakpoints P1, P2..., and Pk. The figure 1 shows that at each of the line segments' ends, $f_{(x)} = \widetilde{f(x)}$.

$$\widetilde{f(x)} = f(\hat{x}_k) + \frac{f(\hat{x}_{k+1}) - f(\hat{x}_k)}{\hat{x}_{k+1} - \hat{x}_k} (\hat{x} - \hat{x}_k),\ \hat{x}_k \leq \hat{x} \leq\ \hat{x}_{k+1} \tag{3}$$

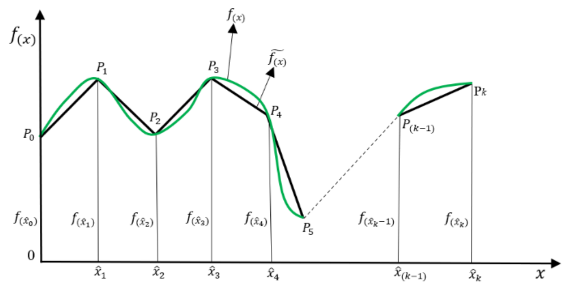


Fig 1: Piecewise Linear Approximation (PLA) $\mathbf{\widetilde{f(x)}}$ of Nonlinear function

In (3), the fraction $\frac{\hat{x} - \hat{x}_k}{\hat{x}_{k+1} - \hat{x}_k}$ is a number between 0 and 1 for any value of x between x_k and x_k+*1*. Let's define,

$$\lambda_{k+1}\ \coloneqq\ \lambda \coloneqq \frac{\hat{x} - \hat{x}_k}{\hat{x}_{k+1} - \hat{x}_k} \tag{4}$$

The (3) can be expressed as

$$\widetilde{f(x)} = f(\hat{x}_k) + \lambda(f(\hat{x}_{k+1}) - f(\hat{x}_k))$$

$$= \lambda f(\hat{x}_{k+1}) - (1 - \lambda) f(\hat{x}_k), \hat{x}_k \leq \hat{x} \leq\ \hat{x}_{k+1} \tag{5}$$

Introducing $\lambda_k = 1 - \lambda$ and $\lambda_{k+1} = \lambda$ we obtain

$$\widetilde{f(x)} = \lambda_k f(\hat{x}_k) + \lambda_{k+1} f(\hat{x}_{k+1}),\ \hat{x}_k \leq \hat{x} \leq\ \hat{x}_{k+1} \tag{6}$$

Where, $\lambda_k\ + \lambda_{k+1}\ = 1\ ,\quad \lambda_{k\ ,} \lambda_{k+1}\ \geq 0.$ (7)

Using (3), we get,

$$x = \hat{x}_k +\ \lambda\ (\hat{x}_{k+1} - \hat{x}_k)\ = (1 - \lambda)\hat{x}_k + \lambda\ \hat{x}_{k+1} = \lambda_k\ \hat{x}_k + \lambda_{k+1}\ \hat{x}_{k+1} \tag{8}$$

So, we can express any point x in the closed interval $[\hat{x}_k, \hat{x}_{k+1}]$ as

$$x = \lambda_k\ \hat{x}_k + \lambda_{k+1}\ \hat{x}_{k+1} \tag{9}$$

Subject to

$$\lambda_k\ + \lambda_{k+1}\ = 1\ ,\quad \lambda_{k\ ,} \lambda_{k+1}\ \geq 0.$$

This leads to the representation of $f(x)$ using a set of weighting variables, $\lambda_{k\ ,}\ \ k = 1, \ldots, K$ by the equality

$$\widetilde{f(x)} = f(\hat{x}_1)\lambda_1\ + f(\hat{x}_2)\lambda_2\ +\ \ldots + f(\hat{x}_k)\lambda_k \tag{10}$$

Where,

$$\hat{x}_1\lambda_1\ +\ \hat{x}_2\lambda_2\ + \cdots + \hat{x}_k\lambda_k\ - x = 0\ ,\quad x\ \geq 0 \tag{11}$$

$$\lambda_1\ + \lambda_2\ + \cdots +\ \lambda_k\ \ = 1\ ,\ \lambda_k\ \ \geq 0\ ,\quad k = 1,2, \ldots, K \tag{12}$$

In (10), $\widetilde{f(x)}$ represents piecewise linear approximation (PLA) of the original nonlinear function f(x). (12) must now include a constraint that no more than two adjacent λ's can ever be non-zero at once in a viable solution in order to express f(x) by $\widetilde{f(x)}$. Here, (10), (11), and (12) are referred to as function rows, reference rows, and convexity rows, respectively and they together defined as "λ - formulation" of the original non-linear function f f(x). (12) is also termed as convexity condition to be included in the PLA optimization problem. The λ - formulation is a technique used to represent the PLA problem in a linear programming (LP) problem which is called approximated LP Problem [5]. The weighting variables λ_k 's are the design variables and known as the special ordered set type two (SOS2) variables [27][28].

When SOS2 variables are included in a branch and bound framework, they intend to find a truly global optima rather than simply local ones [29]. In the proposed work, the SVM optimization problem is transformed into an approximating LP problem and solved using well-known GUROBI Optimizer Solver which uses in-built facility of solving SOS2 variables with branch and bound and mixed integer programming techniques.

3.2 Primal formulation of soft- margin Support Vector Machine

The Figure 2 illustrates the soft - margin SVM for a two-dimensional data set that is not linearly separable. It shows the separating hyperplane which allows some misclassification with the help of slack variables ξi, where i = 1 to n, n = number of observations.The primal formulation of soft-margin SVM as a quadratic optimization problem is given as:

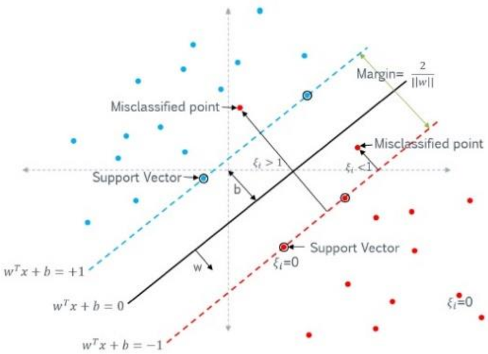


Fig 2: Soft- Margin SVM

$$Objective\ Function : \underset{\xi, W,}{\text{Min}}\, J : \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\xi i)^{p} \quad (13)$$

Subject to:

$$\text{Constraints } c_i = y_i(w^T x_i + b) \geq 1 - \xi i \quad (14)$$

Where ξi ≥ 0 for i = 1, 2, ..., n

In the above equations, we have, x_i as the feature vector of the ith training example, yi as the corresponding class label (+1 or -1), w as the weight vector, b as the bias term, n as the number of training examples, ξi as the slack variable representing the degree of misclassification for the ith example, C as the regularization parameter, controlling the trade-off between maximizing the margin and minimizing the misclassifications. A larger C allows for fewer misclassifications but may result in a narrower margin.

The objective function consists of two terms:

- The first term, $\frac{1}{2} w^T w$, aims to maximize the margin by minimizing the norm of the weight vector.
- The second term, $C \sum_{i=1}^{n} (\xi i)^{k}$ penalizes misclassifications where p=1 is called hinge loss while p=2 is quadratic loss. In the present work, the SVM optimization is solved using hinge loss that is p=1.

4. The Proposed Methodology: Piecewise Linear Approximation based SVM (PLASVM)

In the pursuit of enhancing the efficiency and applicability of Support Vector Machines (SVMs) for classification tasks, this section introduces a novel approach to tackle the primal soft margin SVM optimization problem. In this section, we present our approach to solving the primal soft-margin Support Vector Machine (SVM) optimization problem using a piecewise linear approximation (PLA) technique based on the λ-formulations of the non-linear objective function and linear constraints. We propose a methodology that hinges on the concept of PLA and leverages λ-formulations to render the problem more amenable to efficient optimization techniques.

The proposed methodology addresses these computational challenges by transforming the primal soft-margin SVM problem into a piecewise linear approximation deriving λ-formulations of soft-margin SVM, thus casting it into the realm of linear programming. The crux of our approach lies in the introduction of lambda variables, which enable us to approximate the SVM objective function and constraints with piecewise linear segments.

4.1. The Formulation of Soft-margin PLASVM Problem

To bridge the gap between the non-linear nature of the soft-margin SVM optimization problem and the tractability of linear programming techniques, we propose to reformulate the problem as a PLA problem. To formulate the soft-margin PLASVM, we have derived the $\lambda -$ formualtions of the objective function and constraints of the original soft-margin SVM explained in (13) and (14) based on themethodology developed by Shital and Ramesh [3]. Here, the constraints are linear and the objective function is quadratic and separable. So, the aforementioned optimization problem is perfectly suited to be modeled as a separable linear programming problem. The optimal values for vectors w and b in this optimization problem are those that maximize the distance between classes. The soft-margin PLASVM problem is formulated by deriving the λ -formulation of the objective function J in (13) and constraintsc_i for i $= 1,2,3 \dots .$ n., in (14) as follows:

Let's denote input feature space as x_j, where j= 1, 2, 3……m, m= number of features/predictors in dataset and, label or output as y_i where i=1,2, 3....n, n = amount of data in the dataset. Define x_{ji} as a value of j^{th} feature at i^{th} data point in the dataset. Let's define ξi as the slack variable representing the degree of misclassification for the i^{th} example and C as the regularization parameter, controlling the trade-off between maximizing the margin and minimizing the misclassifications. In the (13), w is a weight vector and the number of elements of w are equal to a number of features (m) in the given dataset.Now, let's define,$w = [w_1\ w_2\ \ w_3 \dots \dots . w_m]$, m = number of features. Let's define the initial search domain for the elements of weight vector w and slack variables ξi as $w_j = [w_j^L, w_j^U]$ and $\xi_i = [\xi_i^L, \xi_i^U]$ respectively. Here, w_j^L = Lower bound on w_j, ξ_i^L = Lower bound on ξ_i, w_j^U= Upper bound on w_j, ξ_i^U= Upper bound onξ_i where j= 1,2,3...m, i=1, 2…n.

The number of breakpoints or intervals required for the initial search domains of w_j and $\xi_i$for carrying out piecewise linear approximation of (13) and (14). Let's denote $I_{w_1}, I_{w_2}, I_{w_3}$......$I_{w_m}$and $I_{\xi_1}, I_{\xi_2}, I_{\xi_3}$......$I_{\xi_n}$as the number of breakpoints or intervals of the initial search domain of $w_1, w_2, w_3 \dots \dots . w_m$ and $\xi_1, \xi_2, \xi_3 \dots \dots \xi_n$ respectively. For λ -formulation of constraints,let's define the number of break points or intervals of SVM parameter b as I_b and initial search domain as,$b = [b^L, b^U]$. Using the concept of separable programming and derivation of the λ-formulation of linear SVM obtained in [3], we can derive the λ-formulation of the objective function $J$and constraints $c_i$of primal soft-margin SVM optimization problem in (13) and (14) as,

$$Min\, J = Min\ \frac{1}{2}\left\{\sum_{j=1}^{m}\sum_{k=0}^{I_{w_j}}\left(w_{jk}\right)^2\lambda_{w_{jk}}\right\} + C\left\{\sum_{i=1}^{n}\sum_{k=0}^{I_{\xi_i}}(\xi_{ik})\lambda_{\xi_{ik}}\right\} \quad (15)$$

$$c_i^{\lambda} = y_i\left\{\sum_{j=1}^{m}\sum_{k=0}^{I_{w_j}} x_{ji}\ (w_{jk}\lambda_{w_{jk}})\right\} + y_i\{\sum_{k=0}^{I_b} b_k\ \lambda_{b_k}\} + \{\sum_{k=0}^{I_{\xi_i}}(\xi_{ik})\lambda_{\xi_{ik}}\} \quad (16)$$

The (15) and (16) together defines the Soft-margin PLA-SVM. To solve this PLA-SVM efficiently and guarantee global optimality, we employ the GUROBI solver's GUROBI-MATLAB interface [4]. GUROBI is renowned for its prowess in tackling complex optimization problems, particularly Mixed Integer Programming (MIP) problems, through the Branch and Bound algorithm. This choice of solver ensures that the solution obtained is globally optimal, offering the best possible values for the decision variables, including w, b, and ξ's.

The optimal values of SVM parameters w, b and ξ_i, is calculated using the obtained design variables λ's as,

$$(w)_{Opt} = \left[(w_1)_{Opt}\ (w_2)_{Opt} \dots \dots ..\ (w_m)_{Opt}\right] \quad (17)$$

here,$(w_m)_{Opt} = \sum_{k=0}^{I_{wm}} \lambda_{mk} w_{mk}$, and$(b)_{Opt} = \sum_{k=0}^{I_b} b_k\ \lambda_{b_k}$

The misclassification cost can be obtained from the slack variables obtained using λ variables $\lambda_{\xi_{ik}}$as

$$(\xi_n)_{Opt} = \sum_{k=0}^{I_{\xi_n}} \xi_{nk}\lambda_{\xi_{nk}}$$

Out of $\xi_1, \xi_2 \ldots \xi_n$, the proposed method will find only non-zero ξ's for misclassified points in the dataset. The Optimal SVM parameters $(w)_{Opt}$and $(b)_{Opt}$ is used to classify an unknown sample x_{new}using decision function as,

$$Sign\{(y)_{new}\} = {(w)_{Opt}}^T x_{new} + (b)_{Opt} \tag{18}$$

The optimal (minimum) values of the objective function J of PLASVM is obtained as

$$(J)_{opt} = \frac{1}{2}(w)_{Opt}^T\,(w)_{Opt} + C\{\sum_{i=1}^{n}(\xi_i)_{Opt}\} \tag{19}$$

The value of regularization parameter C is obtained by performing a grid search over a range of C values. For example, we have considered C values in a logarithmic scale (e.g., 0.01, 0.1, 1, 10, 100) to train and evaluate the SVM for each C value using K-fold cross-validation. We measure the model's performance (e.g., accuracy) on the validation set for each fold to choose the C value that results in the best average performance (e.g., highest accuracy) across all K folds.

4.2. The proposed PLASVM algorithm

In this subsection, we propose the PLASVM algorithms that underlie our approach and highlight the distinctive advantages of PLASVM in terms of scalability, robustness, and adaptability.

Algorithm 1: The soft-margin PLASVM algorithm:

Input:

The training data set(x_{tr}, y_{tr}), The testing dataset (x_{te}, y_{te}), a prespecified tolerance $\mathcal{E}$ (Default tolerance is 1.00e-04).

Output: "Optimal values of SVM hyper parameters"

Initialization: Initialize$w_j = [w_j^L, w_j^U]$,$j = 1$ to m, $b = [b^L, b^U]$, $\xi_i = [0, \xi_i^U]$, $i = 1$ to n.

Define $I_{w_1}, I_{w_2}, I_{w_3} \ldots I_{w_m}, I_{\xi_1}, I_{\xi_2}, I_{\xi_3} \ldots I_{\xi_n}$ and I_b for $w_1, w_2, w_3 \ldots w_m$, $\xi_1, \xi_2, \xi_3 \ldots \xi_n$ and b respectively.

1: Set C =(0.01, 0.1, 1, 10, 100) and choose K for K cross-validation

2:Obtain PLA of soft – margin SVM

For $i = 1$ to n, $j = 1: m$,

$$\text{Min } \frac{1}{2}\left\{\sum_{j=1}^{m}\sum_{k=0}^{I_{w_j}}(w_{jk})^2\lambda_{w_{jk}}\right\} + C\left\{\sum_{i=1}^{n}\sum_{k=0}^{I_{\xi_i}}(\xi_{ik})\lambda_{\xi_{ik}}\right\}$$

Subject t: $C_i^\lambda = y_i\left\{\sum_{j=1}^{m}\sum_{k=0}^{I_{w_j}} x_{ji}\,(w_{jk}\lambda_{w_{jk}})\right\} + y_i\{\sum_{k=0}^{I_b} b_k\,\lambda_{b_k}\} + \{\sum_{k=0}^{I_{\xi_i}} \xi_{ik}\lambda_{\xi_{ik}}\}$

$\lambda_{w_{j0}} + \lambda_{w_{j1}} + \cdots + \lambda_{w_j I_{w_j}} = 1$, $\lambda_{\xi_{i0}} + \lambda_{\xi_{i1}} + \ldots + \lambda_{\xi_i I_{\xi_i}} = 1$, $\lambda_{b_0} + \lambda_{b_1} + \ldots + \lambda_{bI_b} = 1$

3: Impose SOS2 conditions for each λ's, $\sum_{k=0}^{I_{\xi_i}}\lambda_{\xi_{ik}}$,$\sum_{k=0}^{I_{w_j}} w_{jk}$ and $\sum_{k=0}^{I_b}\lambda_{b_k}$

4:Initiate optimization of the PLASVM in Step 2 using GUROBI solver[4].

5: If the solution is infeasible then, go to initialization step, change the values of $w_j, b\xi_i$, $I_{w_1}, I_{w_2}, I_{w_3} \ldots I_{w_m}, I_{\xi_1}, I_{\xi_2}, I_{\xi_3} \ldots I_{\xi_n}$ and I_b and repeat.

6: Obtain the optimal values of design variables $\lambda_{w_{jk}}$ $(k = 0 \text{ to } I_{w_j})$, λ_{b_k} $(k = 0 \text{ to } I_b)$ and $\lambda_{\xi_{ik}}\big(k = 0 \text{ to } I_{\xi_i}\big)$.

7: Obtain an optimal value of PLASVM parameters $(w)_{Opt}$, $(b)_{Opt}$and $(\xi_i)_{Opt}$using design variables of Step

6 as, $(w_j)_{Opt} = \sum_{k=0}^{I_{w_j}}\lambda_{jk}w_{jk}$, $(b)_{Opt} = \sum_{k=0}^{I_b} b_k\,\lambda_{b_k}$ and $(\xi_i)_{Opt} = \sum_{k=0}^{I_{\xi_i}}\xi_{ni}\lambda_{\xi_{ni}}$

8: Test (x_{te}, y_{te})using the decision function, $y_i = \text{sign}\left(\mathbf{w}_{opt}^T x_{te} + (b)_{Opt}\right)$

9: Evaluate the PLASVM and Deploy.

The proposed soft-margin PLASVM offers several salient features as i) Direct solution of primal optimization ii) No initial guess of design variables requirediii) Feasibility and optimality guarantees due to GUROBI Solver (iv) Efficient for large datasets as it leverages linear programming techniques (v) It generalizes well on unseen data due to global optimal solution of SVM parameters.

5. Experimental Implementation

In this section, the proposed PLASVM is validated on Wisconsin Breast Cancer (WBC) original dataset that is acquired from UCI Machine Learning Repository [2].

5.1. The Dataset

The Wisconsin Breast Cancer dataset, was chosen for this study due to its widespread use in breast cancer research and its relevance in clinical diagnostics. This dataset comprises clinical and morphological features extracted from breast cell biopsies. Originally sourced from the University of Wisconsin Hospitals, the dataset contains a total of 699 instances, each associated with a binary class label: malignant (M) or benign (B), representing cancerous and non-cancerous tumors, respectively. The primary objective when working with the WBC dataset is to develop a classification model that can accurately predict whether a tumor is malignant (cancerous) or benign (non-cancerous) based on the provided features. Table 1 shows details of the features, classes, class distribution and number of missing values of the breast cancer dataset.

Data Pre-processing: Before proceeding with model training and evaluation, the WBC dataset was subjected to rigorous preprocessing. This included the treatment of missing values, standardization of feature scales, and the partitioning of data into training and testing sets. The dataset is partitioned as 70% dataset that is 496 observations are used as a training set and 30% of the dataset that is 209 observations are used as a testing set. In the present work, we have substituted 16 missing observations of the feature Bare Nuclei by its median value of 1.

To optimize the performance of our PLASVM model, an extensive hyperparameter tuning process was undertaken. This involved a systematic search over a range of parameters, such as the number of breakpoints for piecewise linear approximation and the regularization parameter C.

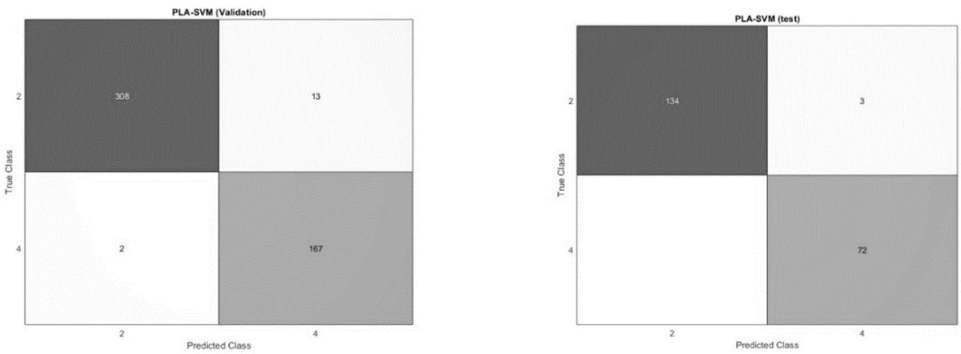
Table 1: Details of the Wisconsin Breast Cancer Dataset

Sr, No.	***Features/Predictors***	***Details***
1	*Sample code number*	*Id number*
2	*Clump Thickness*	*1-10*
3	*Uniformity of Cell Size*	*1-10*
4	*Uniformity of Cell Shape*	*1-10*
5	*Marginal Adhesion*	*1-10*
6	*Single Epithelial Cell Size*	*1-10*
7	*Bare Nuclei*	*1-10*
8	*Bland Chromatin*	*1-10*
9	*Normal Nucleoli*	*1-10*
10	*Mitoses*	*1-10*
11	*Class*	*2 for Benign & 4 for Malignant*
Class Distribution		*Benign: 458(65.5%), Malignant: 241(34.5%)*
Total Number of Observations		*699*
Number of Missing values		*16*

5.2 Implementation of Proposed PLASVM Algorithm

This section discusses numerical experimentation of proposed PLA-SVM on WBC Dataset. To implement the proposed PLA-SVM algorithms on breast cancer dataset, we have set initial search domain of $w_j = [w_j^L, w_j^U] = [-100\ 100]$, $j = 1\ to\ 9, b = [b^L, b^U] = [-100\ 100]\ and\ \xi_i = [0, \xi_i^U] = [0\ 100], i = 1\ to\ 699$. The number of breakpoints or piecewise segments for SVM parameters is set to 200 which initializes $I_{w_1} = I_{w_2} = I_{w_3} ......= I_{w_9} = I_{\xi_1} = I_{\xi_2} = I_{\xi_3} ......I_{\xi_{699}} = I_b$ =200. The hyper parameter C is set as (0.1, 1, 10) and chosen K=10 for K cross-validation. The GUROBI optimizer solver's default tolerance is set at 1.00e-04.

The soft-margin PLASVM problem is set up now. Apply soft-margin PLA-SVM algorithm proposed in Section (4.2) on WBC Dataset to obtain Optimal hyperplanceparamters. The Optimal values of SVM parameters are derived from the design variables λ's as, $(w_1)_{Opt}$ =1.09, $(w_2)_{Opt} = 0.318$, $(w_3)_{Opt} = 0.952$, $(w_4)_{Opt} = 0.273$,$(w_5)_{Opt} = 0.659$,$(w_6)_{Opt} = 1.61$,$(w_7)_{Opt} = 0.813$, $(w_8)_{Opt} = 0.543$, $(w_9)_{Opt} = 0.834$ and $(b)_{Opt} =$



$-0.249$with $C = 1$.

Fig 3: Confusion Matrix of PLASVM (a) Validation (b) Testing

The trained soft-margin PLA-SVM is tested on the test dataset (x_{te}, y_{te}) which is remaining 30% of the breast cancer dataset using the decision function, $y_i = sign\left(w_{opt}^T x_{te} + b_{opt}\right)$.Figure 3 (a) and (b) shows the confusion matrixobtained with proposed soft-margin PLA-SVM on training and testing dataset respectively.Figure 4 shows performance matrices of different classifiers on WBC dataset.

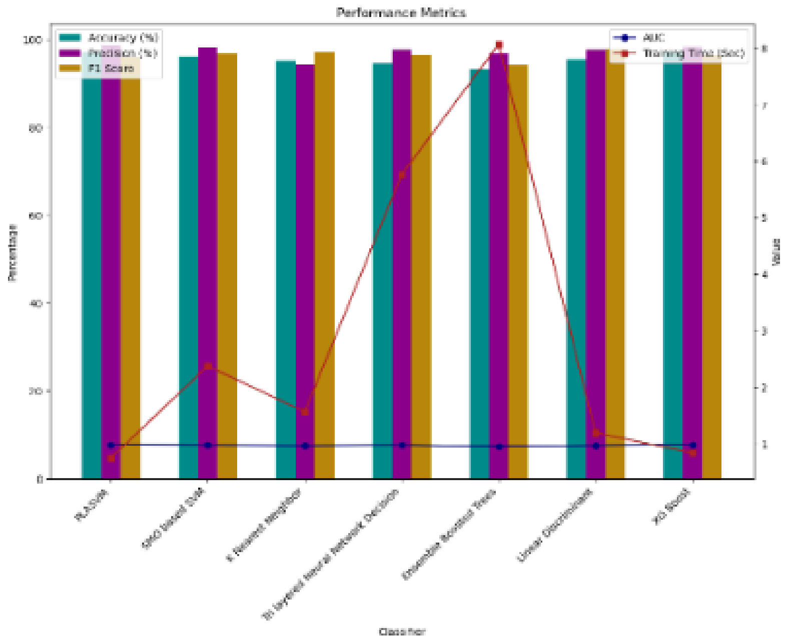


Fig 4: Performance comparison of different Classifiers

5.3 Comparison

To proposed PLA-SVM is evaluated and compared with existing classfiers using metrics like accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic curve (ROC-AUC). MATLAB [10] was chosen as the platform for developing and implementing these classifiers except XGBoost classifier which is modelled using the same pre-processed data in Python.

The proposed PLASVM's performance matrices along with the existing classifiers are shown in Table 2. It is seen that PLA-SVM classifier demonstrated outstanding performance on the WBC dataset, boasting a 98.6% accuracy during testing and excelling in precision, which showcases its proficiency in minimizing false positives. It notched an impressive F1 score of 0.9889 and AUC-ROC of 0.9939, highlighting balanced precision and recall, and distinguished itself with its rapid training time of 0.74 seconds. Though SMO based

SVM and XG Boost marked commendable performances, PLA-SVM's combination of accuracy, efficiency, and speed emerged superior. Its consistency in performance metrics across both validation and testing phases accentuates its robustness and adaptability, particularly for complex, high-dimensional datasets.

In is observed that the blend of parallel and distributed computing, algorithmic optimizations, and efficient resource utilization imbues both XGBoost and PLA-SVM with significant speed. These traits make them especially suited for large datasets and complex machine learning tasks, where both accuracy and computational efficiency are paramount.All the computations were performed on a 4.8 GHz system with an Intel CORE i7 processor and 8 Gb RAM with a 10-fold cross-validation.

Table 2: Comparison of PLASVM with existing classifiers

Sr N o.	*Classifier Name*	*Accuracy (%)*		*Precision (%)*		*F1 Score*		*Area Under the ROC Curve*		*Training Time (Sec)*
		Validation	***Testing***	***Validation***	***Testing***	***Validation***	***Testing***	***Validation***	***Testing***	
1	*PLASVM*	*96.9*	*98.6*	*95.95*	*97.81*	*0.9762*	*0.9889*	*0.9781*	*0.9939*	*0.74*
2	*SMO based SVM*	*96.1*	*98.1*	*96.57*	*98.54*	*0.9703*	*0.9854*	*0.9926*	*0.9990*	*2.37*
3	*K Nearest Neighbor*	*95.1*	*94.3*	*97.2*	*97.08*	*0.9630*	*0.9603*	*0.9416*	*0.9298*	*1.56*
4	*Tri layered NN Decision*	*94.5*	*97.6*	*96.26*	*97.81*	*0.9717*	*0.9817*	*0.9892*	*0.9985*	*5.76*
5	*Ensemble Boosted Trees*	*93.1*	*96.7*	*94.08*	*97.81*	*0.9467*	*0.9745*	*0.9627*	*0.9902*	*8.07*
6	*Linear Discriminant*	*95.5*	*97.6*	*97.82*	*98.54*	*0.9662*	*0.9818*	*0.9917*	*0.9991*	*1.19*
5	*XG Boost*	97.21	98.3	96.26	97.51	0.9793	0.9859	0.9812	0.9909	0.83

6. Conclusion

In the present work, we have introduced a novel and efficient approach for training primal soft-margin Support Vector Machines (SVMs) using PLA based separable programming concept. While the majority of SVM research has traditionally focused on the dual optimization problem, our work has demonstrated the efficacy of the primal perspective in solving the soft-margin SVM design problem. Our proposed method not only efficiently optimizes soft-margin PLASVM parameters using the GUROBI optimizer solver but also introduces a novel way of formulating primal SVM optimization problem using a powerful PLA method. Notably, our method eliminates the need for an initial guess of parameters, requiring only an initial search domain that can be dynamically adjusted based on the solver's outcomes.The validation of our proposed approach on the benchmark WBC dataset reveals its outstanding performancecompared to existing classifiers.

Furthermore, our method's scalability and efficiency make it well-suited for handling challenges associated with high-dimensional and large-scale datasets. As a natural extension of this research, we envision its application in developing kernel-based SVM classifiers for addressing highly non-separable data in multi-class scenarios.

Refrences

[1] C. Cortes and V. Vapnik, "Support-vector networks," Mach. Learn., vol. 20, no. 3, pp. 273-297, 1995. doi:10.1007/BF00994018.

[2]Wolberg, WIlliam. (1992). Breast Cancer Wisconsin (Original). UCI Machine Learning Repository.

[3] Shital Solanki, Dr. Ramesh T. Prajapati "Efficient Primal Support Vector Machine Design Using Piecewise Linear Approximation Based Linear Programming Optimization Techniques" Seybold report, volume 18 (5) 2023, pp 1062-77, doi:10.17605/OSF.IO/2JYMK.

[4] Gurobi Optimization Inc., 2022, Gurobi Optimizer Reference Manual, version 10.0. Houston, Texas.

[5] S. S. Rao, Engineering Optimization: Theory and Practice. John Wiley & Sons, 2019

[6] D. Makwana and P. S. V. Nataraj, "Automated synthesis of fixed structure QFT controller using a piecewise linear approximation based linear programming optimization techniques'' in International

Conference on Industrial Instrumentation and Control. (ICIC), vol. 2015. IEEE, 2015, May, pp. 1597-1602. doi:10.1109/IIC.2015.7151005.

[7] P. S. V. Nataraj and D. Makwana, "Automated synthesis of fixed structure QFT prefilter using a piecewise linear approximation based linear programming optimization techniques," IFAC Papers On-Line, vol. 49, no. 1, pp. 349-354, 2016. doi:10.1016/j.ifacol.2016.03.078.

[8] D. Makwana and P. S. V. Nataraj, "Performance assessment of PLA-LP based QFT control algorithms: An experimental study on industrial plant emulator system," Int. J. Syst. Assur. Eng. Manag., vol. 8, no. S2, pp. 1254-1265, 2017. doi:10.1007/s13198-017-0596-6.

[9] D. Makwana and P. S. V. Nataraj, "Automated Synthesis of Robust QFT-PID Controllers based on Piecewise Linear Approximations" in 14th International Conference on Control. and Automation (ICCA), vol. 2018. IEEE. IEEE, 2018, Jun., pp. 787-793. doi:10.1109/ICCA.2018.8444249.

[10] The MathWorks Incorp., MATLAB Version 9.13.0.2049777, vol. R2022b. Natick, Massachusetts, 2022

[11] S. Chakrabarti et al., "Fast and accurate text classification via multiple linear discriminant projections," VLDB J., vol. 12, no. 2, pp. 170-185, 2003. doi:10.1007/s00778-003-0098-9.

[12] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," Neural Process. Lett., vol. 9, no. 3, pp. 293-300, 1999. doi:10.1023/A:1018628609742.

[13] G. M. Fung and O. L. Mangasarian, "Multicategory proximal Support Vector Machine classifiers," Mach. Learn., vol. 59, no. 1-2, pp. 77-97, 2005. doi:10.1007/s10994-005-0463-6.

[15] S. S. Keerthi et al., "A modified finite Newton method for fast solution of large scale linear SVMs," J. Mach. Learn. Res., vol. 6, no. 3, 2005.

[16] S. S. Keerthi et al., "Building support vector machines with reduced classifier complexity," J. Mach. Learn. Res., vol. 7, no. 7, 2006.

[17] O. Chapelle, "Training a support vector machine in the primal," Neural Comput., vol. 19, no. 5, pp. 1155-1178, 2007. doi:10.1162/neco.2007.19.5.1155.

[18] Zhizheng, Liang & Li, Youfu. (2009). Incremental support vector machine learning in the primal and applications. Neurocomputing. 72. 2249-2258. 10.1016/j.neucom.2009.01.001.

[19] Q. Wu and W. Wang, "Piecewise-smooth support vector machine for classification," Math. Probl. Eng., vol. 2013, 1-7, 2013. doi:10.1155/2013/135149.

[20] Vapnik, V.N. (1995) The Nature of Statistical Learning Theory. Springer, New York.http://dx.doi.org/10.1007/978-1-4757-2440-0.

[21]. D. Lavanya and K. U. Rani, "Analysis of feature selection with classification: Breast cancer datasets," Indian Journal of Computer Science and Engineering (IJCSE), vol. 2, no. 5, pp. 756–763, 2011.

[22] Sivakami, K.; Saraswathi, N. Mining big data: Breast cancer prediction using DT-SVM hybrid model. Int. J. Sci. Eng. Appl. Sci. 2015, 1, 418–429.

[23] McKinney, S.M.; Sieniek, M.; Godbole, V.; Godwin, J.; Antropova, N.; Ashrafian, H.; Back, T.; Chesus, M.; Corrado, G.S.; Darzi, A.; et al. international evaluation of an AI system for breast cancer screening. Nature 2020, 577, 89–94.

[24] Akbulut, S.; Cicek, I.B.; Colak, C. Classification of Breast Cancer on the Strength of Potential Risk Factors with Boosting Models: A Public Health Informatics Application. Med Bull. Haseki/Haseki Tip Bul. 2022, 60, 196–203.

[25] M. Zhao. New developments on polyhedral methods for mixed-integer programming. PhD Thesis, University of Buffalo, New York, U.S.A, 2008

[26] S. P. Bradley, A. C. Hax, and T. L. Magnanti. Applied Mathematical Programming. Addison-Wesley Publishing Company, Boston, U.S.A., 1977.

[27] J. A. Tomlin. A Suggested Extension of Special Ordered Sets to Non-Separable Non-Convex Programming Problems, In: Hansen, P. (Ed.). North-Holland Publishing Company, Amsterdam, Netherland, 1981

[28] Beale, E. and Tomlin, John, Special facilities in a general mathematical programming system for nonconvex problems using ordered sets of variables, Journal of Operation research 1969,447-454.

[29] E. M. L. Beale and J. J. H. Forrest. Global optimization using special ordered sets. Mathematical Programming, 10(1):52–69, 1976.