KL2-BiQPSVM: Kernel-free L2 type bi-quadratic potential support vector machine

Abstract

The concept of a bi-quadratic potential function is introduced in quantum mechanics. Some special form of this function is further included in a well known machine learning technique, support vector machine (SVM), as an attempt to remove the kernel based dependency of SVM algorithm. For non-linear datasets, the kernel dependency takes a huge amount of time to compute the support vectors but the Bi-Quadratic function is already a nonlinear function which can directly identify the non-linear features in a dataset. In previous attempts of adding kernel free approach in SVM caused several problems such that: **repetitive feature space**,**run time complexity** and **low accuracy score**. In KL2-BiQPSVM, the input feature space is first taken into a new higher order feature space by using the monomial expansion of the input features up to 4^{th} order, thus evaluating all the non-linear correlations among the input features. Then, if the final accuracy score is low on traditional L_1 type norm, in order to penalise the function more for each miss-classified data points, optional L_2 norm is used on slack variables. Finally for L_1 type norm, sequential minimal optimization (SMO) is used for fast convergence and for L_2 uses standard quadratic programming to find optimal support vectors. The superiority of KL2-BiQPSVM is demonstrated over eight benchmark datasets (both categorical and continuous) obtained from UCI machine learning repository. Results showed that KL2-BiQPSVM gives better accuracy and faster convergence than some traditional state of the art SVM algorithms used on real life datasets.

Keywords: Support vector machine, Double well potential function, Kernel-free SVM, Binary classification, Sequential minimal optimisation, Monomial expansion, Reinforced SVM.

1. Introduction

Classification and regression based models created major impacts in the field of artificial intelligence and machine learning. Support Vector Machine (SVM), proposed by Vapnik (1995), is a very popular, useful and successful method in those fields. Given a data set, the classical linear SVM model (*Cortes & Vapnik,1995*) [1] separates the data points into two classes utilizing a hyper plane, while the margin between the two classes is maximized and the misclassified data points is minimized. It works effectively for linearly separable data sets but it fails for nonlinear separable data sets. But in real life, maximum datasets are nonlinear. So, in that case various Kernel functions are used (linear, Gaussian, Polynomial), by taking the input feature space into higher nonlinear dimension.

Although this method has achieved great success but it has many drawbacks too. First, there are no general principles to select a suitable kernel function for a given data set. Second, using some kernel functions may be computationally expensive since the inverse of a kernel matrix is needed for solving the dual problem. Third, the singularity issue of kernel matrix may influence the learning process of the models. Although, this issue was resolved by using SMO algorithm *(Platt 1988)* [2]. To overcome those drawbacks of kernel-based SVM models, the kernel-free Quadratic Surface Support Vector Machine *(QSSVM)* [3]model was proposed by directly using quadratic surfaces for nonlinear separation. But QSSVM does not work for all the datasets. It can not handle highly nonlinear separable datasets. So,Double well Potential, a fourth degree polynomial function has been proposed to overcome all the disadvantages of QSSVM and kernel based SVM *(Gao, Fang, Luo, Medhin 2021)* [4].Double well Potential is more nonlinear than a quadratic or a cubic function. This function has strong potential to handle highly non-linearly separable datasets. Therefore DWPSVM was proposed by using a special bi-quadratic function to convert the input feature space into 4^{th} order.

Although this algorithm can handle highly nonlinear datasets but it also has some major run time problem. This model creates repetitive features in the feature space which increases the model complexity and over fits the model. Also for some datasets the accuracy score is low for this model.

The objective of this study is to develop a new model for classification and regression, namely Kernel free L2 Bi-Quadratic SVM (KL2-BiQPSVM) function which can handle the aforementioned issues of Linear SVM, Kernel based SVM and DWPSVM. In this approach first the feature creation process of KL2-BiQPSVM is changed. Then a new cost function is introduced to fix low accuracy score issue. Also SMO algorithm with proper heuristics has been used for fast convergence. A total of 8 benchmark continuous and categorical datasets are used to test the hypothesis. Multi-class classification problems can easily be converted into binary classification using

one vs all method. So our primary focus in this paper is to use this model for binary classification which can be easily generalized for multi class classification problem.

The main contributions in this model towards the field of binary classification are:

1. Based on the paper of (Gao, Fang, Luo, Medhin 2021), kernel free DWPSVM is a special type of bi-quadratic function which handles highly nonlinear dataset and overcomes the drawbacks of kernel based SVM and kernel free QSSVM. But from the previous approach of this model forced it to create some particular features values multiple times thus increasing the feature vector unnecessarily complicated and over-fitting the model. Any function can be expanded by using the Taylor expansion formula. For multi-variable functions total number of unique terms can be calculated using a combination formula. Each unique term in the expansion is called **Monomials**[5]. Use of monomial expansion can reduce the dimension of new individual features by a huge number and decrease the over-fitting problem. It will also increase computational efficiency of the model.
2. Previous DWPSVM model's accuracy was low for some datasets. In order to resolve this issue, L_2 norm has been used as the slack variable to penalize the model whenever the model miss classifies a data-point, thus forcing it to choose the parameter values carefully. Based on the empirical evidences, best norm is used for a dataset.
3. Standard quadratic programming problem causes to slow down the optimization process for large datasets. So, the well-known Sequential Minimal Optimization(SMO) algorithm (John Platt) has been implemented on the proposed model for computational efficiency for L_1 type slack variable. Numerical results indicates its efficiency for as feature space increases. It will help to reduce the time complexity of the model for large datasets. Since SMO is not developed for L2 type norms, standard quadratic programming method is used for this. This norm is mostly suitable on small and medium type datasets.

The structure of the paper is organized as follows. Brief demonstration of SVM for binary classification in Section. Proposed BiQPSVM algorithm is explained in Section. Section discuss the mathematical approach behind various norms for slack variable used in the proposed model. Section explained the SMO algorithm for optimization. Section shows in-depth experimental results of the KL2-BiQPSVM algorithm and a comparison of its performance with some state-of-the-art algorithms. Finally, the conclusions including scope for future works are discussed in Section .

2. Preliminaries

In this section required preliminary math for SVM has been introduced. The classification of binary classes and the algorithm to compute optimal solution is the backbone of SVM. First section explains the math to calculate the cost function of the SVM. Second section explains the heuristics for famous SMO algorithm to calculate the weights analytically.

2.1. ***SVM with Binary Classification***

The goal of binary classification is to find a separation surface which can separate accurately a given dataset with 2 classes. Let D be a dataset consisting of two classes.

$$D = (x_i, y_i),\ i = 1, \ldots, N\ \&\ x_i \in R^N,\ y_i \in \{-1, 1\} \tag{1}$$

where N is the data size, n is the number of features, $x_i = [x_{i1}, \ldots, x_{in}]^T \in \mathrm{R^n}$ is the vector form of n-dimensional feature space. and $\mathrm{y_i}$ is the label of $\mathrm{i^{th}}$ training example. The positive and negative labeled index sets are denoted as $M^+ = \{\mathrm{i} : y_i = 1\}$ and $M^- = \{i :\ y_i = -1\}$; therefore the total index set is $M = M^+ \cup M^-$. **A dataset is linearly separable if there exists a, $\mathbf{u} \in \mathbf{R^n}$ and $\mathbf{d} \in \mathbf{R}$ such that:**

$$u^T x_i + d > 0\ (i \in M^+)\ \&\ u^T x_i + d < 0\ (i \in M^-) \tag{2}$$

Given a linearly separable data set D, the idea of SVM is to separate the data by a hyperplane while the margin of separation is maximized (Cortes & Vapnik, 1995). The separation function is denoted as, $\mathrm{f(x) = u^T x + d}$, then the width of margin equals to $\frac{2}{||u||_2}$.

If the data set D is not linearly separable, the soft-margin concept (Cortes & Vapnik, 1995) is adopted by introducing the slack vector like,

$$\xi = [\xi_1, \ldots, \xi_N] \in R^N \tag{3}$$

To allow the location of points to violate constraints. The soft-margin SVM [6]is formulated as the following model (SSVM):

$$min\ \frac{1}{2}||u||_2^2 + C\sum_{i=1}^{N}\xi_i \tag{4}$$

where $C > 0$ is the penalty parameter for misclassified data points. If the dataset is not linearly separable, kernel function is introduced by Vapnik (2013). The SVM with well-known kernel function can be formulated as follows:

$$\begin{aligned} &min\ \frac{1}{2}||v||_2^2 + C\sum_{i=1}^{N}\xi_i \\ &\text{where, } y_i(v^T\phi(x_i) + d) \geq 1 - \xi_i\ \forall\ i = 1 \dots N \\ &\qquad\qquad \&\ v \in R^l,\ d \in R,\ \xi_i \in R^N; \end{aligned} \tag{5}$$

where ϕ maps data points from R^n to R^l $(n \leq l)$ and $K(\phi(x_i), \phi(x_j)) = \phi(x_i)^T\phi(x_j)$ is a kernel function for any x_i & x_j.
There are several kernel functions including the frequently used Gaussian (RBF) kernel and Quadratic (2^{nd} order polynomial) kernel (Scholkopf & Smola, 2001). The main idea behind using a kernel function with SVM is to first map the data points into a higher dimensional feature space and then separate the mapped data points with a hyperplane in the higher dimensional space. Soft margin SVM model(SSVM) with a kernel function is also a convex quadratic programming (QP) problem, which can be formulated as the following:

$$\begin{aligned} &\min_{\alpha}\ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{N}\alpha_i \\ &\sum_{i=1}^{N}\alpha_i y_i = 0 \text{ where } 0 \leq \alpha_i \leq C\ \forall i = 1, \dots, N \end{aligned} \tag{6}$$

where C is the penalty for each misclassified data point.
One of the most popular approaches proposed in this paper for solving its dual problem is the sequential minimal optimization (SMO) algorithm which will be using for fast optimization for the KL2-BiQPSVM model.
Furthermore, by directly employing quadratic surfaces for separations, Dagher (2008) and Luo et al. (2016) proposed and constructed kernel-free nonlinear SVM models. The aim behind these kernel-free SVM models is to separate the data in the original space rather than transferring it onto a higher dimensional feature space.
According to Luo et al. (2016), a data set D is quadratically separable if there exists $W \in S^n$, $b \in R^n$ and $c \in R$, such that

$$\frac{1}{2}x_i^T W x_i + b^T x_i + c > 0\ (i \in M^+) \tag{7}$$

$$\frac{1}{2}x_i^T W x_i + b^T x_i + c < 0\ (i \in M^-) \tag{8}$$

Given a quadratically separable data set, the quadratic separation hyperplane obtained from the quadratic surface SVM (QSSVM) is represented by: $f(x) = \frac{1}{2}x_i^T W x_i + b^T x_i + c = 0$. A typical model is the following SQSSVM model, which not only minimizes the sum of relative geometrical margins of points but also adopts the soft-margin idea:

$$\begin{aligned} &min\sum_{i=1}^{N}||Wx_i + b||_2^2 + C\sum_{i=1}^{N}\xi_i, \\ &\text{such that } \forall\ i = 1, \dots, N,\ v \in R^l,\ d \in R,\ \xi \in R^{N+} \end{aligned} \tag{9}$$

SSVM model yields a linear separation function $H(x) = u^T x + d$, with $u \in R^n$, $d \in R$, while SQSSVM yields a quadratic separation function $Q(x) = \frac{1}{2}x^T W x + b^T x + c$, with $W \in S^n$, $b \in R^n$, $c \in R$.
The Gaussian kernel function generates a very nonlinear separation margin in the SVM. Quadratic surfaces are more effective in classification than hyper-planes, according to Luo et al. (2016) and Mousavi et al. (2019). In this paper the performance is verified by using a specific quadratic surface for binary classification.
After creating the cost function, in order to optimize the weights, the cost function must be minimized. This process can be done in two ways, **Analytically** or **Numerically**. SMO algorithm uses the numerical approach to optimize the weights, therefore this is much faster than the analytical approach. In the next section the mathematical formulation is described for SMO algorithm.

2.2. *Sequential Minimal Optimization*

SMO solves the SVM QP problem by decomposing it into QP sub-problems and solving the smallest possible optimization problem, involving two Lagrange multipliers, at each step. These small QP problems are solved analytically, which avoids using a time-consuming numerical QP optimization as an inner loop. The amount of memory required for SMO is linear in the training set size, which allows SMO to handle very large training sets. Because matrix computation is avoided, SMO scales somewhere between linear and quadratic in the training set size for various test problems. SMO's computation time is dominated by SVM evaluation; hence SMO is fastest for linear SVMs and sparse data sets. The advantage of SMO lies in the fact that solving for two Lagrange multipliers can be done analytically. Thus, numerical QP optimization is avoided entirely. SMO requires no extra matrix storage at all. Thus, very large SVM training problems can fit inside of the memory of an ordinary personal computer or workstation. Because no matrix algorithms are used in SMO, it is less susceptible to numerical precision problems.

KKT Conditions: The Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient conditions for an optimal point of a positive definite QP problem. The KKT conditions for the QP problem (11) are particularly simple. The QP problem is solved when, for all i:

$$\alpha_i = 0 \iff y_i u_i \geq 1 \tag{10}$$

$$0 < \alpha_i < C \iff y_i u_i = 1 \tag{11}$$

$$\alpha_i = C \iff y_i u_i \leq 1 \tag{12}$$

From equation (6) it can be seen that two Lagrange multipliers (α) can be changed simultaneously while satisfying the KKT conditions. Therefore in order to optimize the weights at least two α_i should be optimized in a single step. In the next section the process to optimize the α is explained.

2.2.1. *Solving for two Lagrangian multipliers*

In order to solve for the two Lagrange multipliers, SMO first computes the constraints on these multipliers and then solves for the constrained minimum. The ends of the diagonal line segment can be expressed quite simply. Without loss of generality, the algorithm first computes the second Lagrange multiplier α_2 and computes the ends of the diagonal line segment in terms of α_2. If the target y_1 does not equal the target y_2, then the following bounds apply to α_2 :

$$L = max(0, \alpha_2 - \alpha_1),\ H = min(C, C + \alpha_2 - \alpha_1).$$

If the target y_1 equals the target y_2, then the following bounds apply to α_2:

$$L = max(0, \alpha_2 + \alpha_1 - C), H = min(C, \alpha_2 + \alpha_1)$$

The second derivative of the objective function along the diagonal line can be expressed as:

$$\eta = K(x_1, x_1) + K(x_2, x_2) - 2K(x_1, x_2).$$

Under normal circumstances, the objective function will be positive definite, there will be a minimum along the direction of the linear equality constraint, and η will be greater than zero. In this case, SMO computes the minimum along the direction of the constraint:

$$\alpha_2^{new} = \alpha_2^{old} + \frac{y_2(E_2 - E_1)}{\eta} \tag{13}$$

where $E_i = u_i - y_i$ is the error on the i^{th} training example. As a next step, the constrained minimum is found by clipping the unconstrained minimum to the ends of the line segment:

$$\begin{aligned} \alpha_2^{new,clipped} &= H \text{ if } \alpha_2^{new} \geq H \\ \alpha_2^{new,clipped} &= \alpha_2^{new} \text{ if } L < \alpha_2^{new} < H \\ \alpha_2^{new,clipped} &= L \text{ if } \alpha_2^{new} \leq L \end{aligned} \tag{14}$$

Now, let $s = y_1 y_2$. The value of α_1 is computed from the new, clipped, $\alpha_2^{new,clipped}$:

$$\alpha_1^{new,clipped} = \alpha_1^{new} + s(\alpha_2 - \alpha_2^{new,clipped}) \tag{15}$$

After optimizing two Lagrange Multipliers, threshold value should updated according to the necessary constrains. In order to update the constraint of equation (5) has been used. The mathematical approach is

explained in the next section.

2.2.2. Computing the Threshold Value

The threshold b is re-computed after each step, so that the KKT conditions are fulfilled for both optimized examples. The following threshold b_1 is valid when the new α_1 is not at the bounds, because it forces the output of the SVM to be y_1 when the input is x_1:

$$b_1 = E_1 + y_1(\alpha_1^{new,clipped} - \alpha_1)K(x_1, x_2) + y_2(\alpha_2^{new,clipped} - \alpha_2)K(x_1, x_2) + b \tag{16}$$

The following threshold b_2 is valid when the new α_2 is not at bounds, because it forces the output of the SVM to be y_2 when the input is x_2:

$$b_2 = E_2 + y_1(\alpha_1^{new,clipped} - \alpha_1)K(x_1, x_2) + y_2(\alpha_2^{new,clipped} - \alpha_2)K(x_1, x_2) + b \tag{17}$$

When both b_1 and b_2 are valid, they are equal. When both new Lagrange multipliers are at bound and if L is not equal to H, then the intervals between b_1 and b_2 are all thresholds that are consistent with the KKT conditions. SMO chooses the threshold to be halfway in between b_1 and b_2. Therefore,

$$b = \begin{cases} b_1, & \text{if } \alpha_1 \text{ is not at the bounds} \\ b_2, & \text{if } \alpha_2 \text{ is not at the bounds} \\ \frac{b_1+b_2}{2} & \text{if both of them are at the bounds} \end{cases} \tag{18}$$

To compute a linear SVM, only a single weight vector $\boldsymbol{w}$ needs to be stored, rather than all of the training examples that correspond to non-zero Lagrange multipliers. If the joint optimization succeeds, the stored weight vector needs to be updated to reflect the new Lagrange multiplier values. The weight vector update is easy, due to the linearity of the SVM:

$$w_1^{new} = w + y_1(\alpha_1^{new} - \alpha_1)x_1 + y_2(\alpha_2^{new} - \alpha_2)x_2 \tag{19}$$

3. Proposed Model

In this section, first BiQPSVM is introduced and the mathematical background is explained then in the second section a new slack variable to calculate the cost function is introduced and finally based on the new norm the final KL2-BiQPSVM is proposed.

3.1. ***Bi-quadratic potential SVM(BiQPSVM)***

The BiQPSVM is a special kind of polynomial function. There are several type of 4^{th} order polynomial function which is used as a kernel free function in SVM; like Double Well Potential function. The mathematical idea of DWPSVM function is explained in the next section and it's drawbacks are discussed.

3.1.1. Double Well Potential Function

The DWP function is a special type of bi-quadratic function which was proposed by Gao, Fang, Luo and Medhin. Let F be a real valued DWP function defined on R^n such that

$$F(x) = \frac{1}{2}\left(\frac{1}{2}||Bx - c||_2^2 - d\right)^2 + \frac{1}{2}x^T Ax + b^T x + q \tag{20}$$

where $B \in R^{m\times n}$, $c \in R^m$, $d \in R$, $A \in S^n$, $b \in R^n$, $q \in R$.
Define $s_i = lvec(x_i)$, $w_B = hvec(B^T, B)$, $w_{Bc} = c^T B$, and $c_d = \frac{1}{2}c^T c - d$, then we have $\xi_i = s_i^T w_B - x_i^T w_{Bc} + c_d$.
Define, $z_i = [s_i^T, x_i^T, 1]$, and $w_\xi = [w_B^T, w_{Bc}^T, c_d]^T$. Therefore,

$$F(x_i) = \frac{1}{2}z_i^T w_\xi w_\xi^T z_i + \frac{1}{2}x_i^T Ax_i + b^T x_i + q \tag{21}$$

Where function F has a quadratic term with respect to z_i on $R^{\frac{n(n+1)}{2}+2n+1} \to R$, and another quadratic term with respect to x_i on R^n. With the similar vectorization procedure, denote $l = \frac{n(n+1)}{2} + n + 1$ and,
$w_W = hvec(w_\xi, w_\xi^T) \in R^{\frac{l(l+1)}{2}}$; $w_A = hvec(A) \in R^{\frac{n(n+1)}{2}}$.

$$v = [\boldsymbol{w}_W, \boldsymbol{w}_B]^T \in R^{\frac{l(l+1)+n(n+1)}{2}+n}$$
$$r_i = [\eta_i, s_i]^T \in R^{\frac{l(l+1)+n(n+1)}{2}+n}$$

Consequently, $F(x_i)$ equals to a linear function F_l with respect to r_i and x_i in $R^{\frac{l(l+1)+n(n+1)}{2}+n}$,

$$F(x_i) = F_l[r_i, x_i]^T = r_i^T v + x_i^T b + q$$

The Bi-quadratic potential function depicts several forms of 4^{th} order polynomial functions, one of them is the DWP function, that are widely found in physics and in some real life datasets as a very nonlinear function. So the proposed model in this paper is more general and covers more ground than the original DWP model. In the DWPSVM the creation of nonlinear feature space is not optimized and creates unnecessary duplicate features which increase the time complexity to calculate the weights and overfit the model. In the proposed model new and more general way to create the feature space is introduced and explained in the next section.

3.1.2. Redefinition of feature space

A double well potential function is a special kind of bi-quadratic function, which transforms the input feature space to a higher dimension in-order to find the non-linear relations between them. In this process of this transformation, the final feature space becomes very huge due to some unnecessary mathematical manipulation and some features becomes repetitive in the feature space which increases the model complexity and increases the overfitting problem.

$$x_i = [x_i^1, x_i^2] \tag{22}$$

$$s_i = lvec(x_i) = [x_i^{1^2}, x_i^1 x_i^2, x_i^{2^2}] \tag{23}$$

$$z_i = [s_i, x_i, 1] = [x_i^{1^2}, x_i^1 x_i^2, x_i^{2^2}, x_i^1, x_i^2, 1] \tag{24}$$

While expressing η_i like above there are some terms repeating, those are $x_1^2 x_2^2$, $x_1^2 x_2$. In order to resolve this issue one new model is proposed in this paper to create the feature space.
KL2-BiQPSVM uses the idea of multinomial theorem with the count of unique monomials to find the number of features in the higher dimension.
Any non-linear function can be expanded using Taylor expansion formula and each term will represent a polynomial of a certain degree. The idea of bi-quadratic function is to use the expansion up to degree 4 and thus treating the non-linear function as 4^{th} degree polynomial. The previous DWP model approach is going to make an individual **n** number of features to:

$$R^{\frac{l(l+1)+n(n+1)}{2}+n} \text{ where } l = \frac{n(n+1)}{2} + n + 1$$

In general, any function $f(x) : [x \in R^N]$ can be expanded into a polynomial using multi-variable Taylor expansion,

$$f(x,y) \approx Q(x,y) = f(a,b) + f_x(a,b)(x-a) + f_y(a,b)(y-b) + \frac{f_{xx}(a,b)}{2}(x-a)^2 + f_{xy}(a,b)(x-a)(y-b) + \frac{f_{yy}(a,b)}{2}(y-b)^2 \tag{25}$$

This function is an example of such an expansion up to 2^{nd} order. Therefore, the error term will be of order 3. Since KL2-BiQPSVM is a 4^{th} order polynomial, therefore the error term will be of order 5. Since all the real-life dataset has a huge amount of training examples and due to this, there will be very little distance between two consecutive data points (h). The error term is directly proportional to this distance h. Since the error term is already of order 5 and the intermediate distance between two consecutive points is very little, our model error will be very few too.
Let $x = [x_1, x_2]$ be a datapoint consisting of two input feature. If f is the original function to classify the data-points, the Taylor expansion of f followed by equation (24) up-to 2^{nd} order term is,

$$\begin{aligned} f([x_1,x_2]) &= f([a,b]) + f_{x_1}([a,b])(x_1-a) + f_{x_2}([a,b])(x_2-b) + \frac{f_{x_1x_1}([a,b])}{2}(x_1-a)^2 \\ &\quad + f_{x_1x_2}([a,b])(x_1-a)(x_2-b) + \frac{f_{x_2x_2}([a,b])}{2}(x_2-b)^2 + \dots \\ &= c + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_1x_2 + w_5x_2^2 + \dots \\ &= c + W_1x + W_2x^2 + \dots \end{aligned} \tag{26}$$

where a, b and c are constants, and W represent the coefficients of the polynomial expansion.
In equation (25) each term of RHS represents the polynomial terms of various degrees. Like the second term of the RHS in equation (25) is a polynomial of order 1; third term represents a second degree polynomial and so

on.
Therefore in order to find the unique terms of each individual polynomial, the idea of multinomial expansion can be used. The multinomial expansion is defined as,

$$(x_1 + x_2 + \ldots + x_m)^n = \sum_{k_1+k_2+\ldots+k_m=n} \binom{n}{k_1, k_2, \ldots, k_m} \prod_{t=1}^{m} x_t^{k_t} \tag{27}$$

The number of individual terms in the above polynomial expansion are,

$$\binom{n+m-1}{m-1} = \frac{(n+m-1)!}{n!(m-1)!} \tag{28}$$

Therefore total number of unique individual terms for bi-quadratic expansion of the entire function $f(x)$ can be calculated using the summation over equation (27) for degree up-to 4.

$$\begin{aligned} \textit{Total number of terms in the KL2-BiQPSVM} &= \sum_{n=0}^{4} \frac{(n+m-1)!}{n!(m-1)!} \\ &= \binom{m+4}{4} \end{aligned} \tag{29}$$

In order to calculate the unique monomial terms of this polynomial expansion python in-build package "Comparison-with-replacement" has been used which is going to perform all the combinations in the feature space up-to order 4.
In the traditional SVM model, datasets are often not perfectly separable due to high overlapping of different classes. In order to make the model more generalize and reduce overfitting, one new variable is introduced by Vapnik. The next section explains the mathematical background of this slack variable.

3.2. REINFORCED SVM WITH VARIOUS NORMS

In previous models of kernel based SVM models, a slack variable ξ_i were proposed for better generalization and reduce overfitting.
In this proposed kernel free SVM model one new version of SVM slack variables is introduced as a new penalization method, L_2 norm, along with the traditional L_1 norm.
Any slack variable method can be generalized in the form of this equation:

$$\Psi = \frac{1}{2} w^T w + C f(\xi_1, \ldots, \xi_N) \tag{30}$$

subject to the constrains

$$y_i(w^T z_i + b) \geq 1 - \xi_i, \qquad i = 1, \ldots, N \tag{31}$$

These are the two different cost functions used in this model:

3.2.1. L_1 norm of slacks

This is the traditional norm which was introduced by Vapnik in his first SVM paper. The function is defined as:

$$f(\xi_1, \ldots, \xi_N) = \sum_{i=1}^{N} \xi_i \ , \qquad \xi_i \geq 0 \tag{32}$$

Therefore the objective function becomes,

$$\Psi(\alpha) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N} \alpha_i \tag{33}$$

subject to these constrains,

$$\sum_{i=1}^{N} y_i \alpha_i = 0, \tag{34}$$

$$0 \leq \alpha_i \leq C \tag{35}$$

Although this version of slack variable has been used and successfully implemented in various models, for some datasets the accuracy score is low from this type of norm. Therefore one new norm has been proposed in this model; the L_2 norm. The mathematical expression has been explained in the next section.

3.2.2. L_2 norm of slacks

This is the new proposed slack version in this model to increase the accuracy of the model. The slack variable function for this type is,

$$f(\xi_1, \ldots, \xi_N) = \frac{1}{2}\sum_{i=1}^{N} \xi_i^2 \tag{36}$$

Therefore the objective function becomes,

$$\Psi = \frac{1}{2} w^T w + \frac{C}{2}\sum_{i=1}^{N} \xi_i^2 \tag{37}$$

subject to equation 26.
Therefore using the Lagrangian method, the dual problem becomes,

$$L = \frac{1}{2} w^T w + \frac{C}{2}\sum_{i=1}^{N} \xi_i^2 + \sum_{i=1}^{N} \alpha_i [1 - \xi_i - y_i(w^T z_i + b)], \qquad \alpha_i \geq 0 \tag{38}$$

From the necessary conditions of minimum,

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{N} \alpha_i y_i z_i \tag{39}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0. \tag{40}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow C\xi_i - \alpha_i = 0 \Rightarrow \xi_i = \frac{\alpha_i}{C}. \tag{41}$$

Putting equation 34, 35, 36 into the objective function Ψ,

$$\Psi(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N} \alpha_i \alpha_j \left[y_i y_j z_j^T z_i + \frac{\delta_{i,j}}{C} \right] \tag{42}$$

where, $\delta_{i,j} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$

In order to find the optimal solution one needs to maximize the objective function Ψ. Therefore the primary dual problem now becomes,

$$\min_{\alpha} \Psi(\alpha) = \min_{\alpha} \left(\frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N} \alpha_i \alpha_j \left[y_i y_j z_j^T z_i + \frac{\delta_{i,j}}{C} \right] - \sum_{i=1}^{N} \alpha_i \right) \tag{43}$$

subject to these constrains,

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{44}$$

$$\alpha_i \geq 0 \tag{45}$$

Since the SMO algorithm was not developed for the L_2 type slack norm, standard quadratic programming method has been used to solve the optimization method. Therefore, the classification is predicted using the final optimized α_i, The computational efficiency of the KL2-BiQPSVM is tested and compared with various state of the art algorithms in the next section.

Data: Input Dataset
Result: Weights of the KL2-BiQPSVM model
Pre-process the Dataset;
X← Feature Vector and y ← Class Labels;
while *Traverse through X Feature Vector* **do**
  read feature row x_i;
  $X_i^{new} = Taylor\ Expansion\ with\ Monomial\ count(deg = 4, x_i)$;
end
if *default norm* $= L_1$ **then**
  W = SMO(X^{new},y);
else
  W = Standard Quadratic Method(X^{new},y);
end
return W

Algorithm 1: Complete algorithm for KL2-BiQPSVM

4. Computational Experiments

In this part, we run computational tests to see how well KL2-BiQPSVM performs on a variety of artificial, public benchmark, and real-world data sets. The proposed KL2-BiQPSVM model and well-known SVM models are first tested with some artificial and public benchmark data sets.By comparing the proposed Reinforced KL2-BiQPSVM model to well-known SVM models, it is extended and used on certain benchmark and real-life credit datasets.

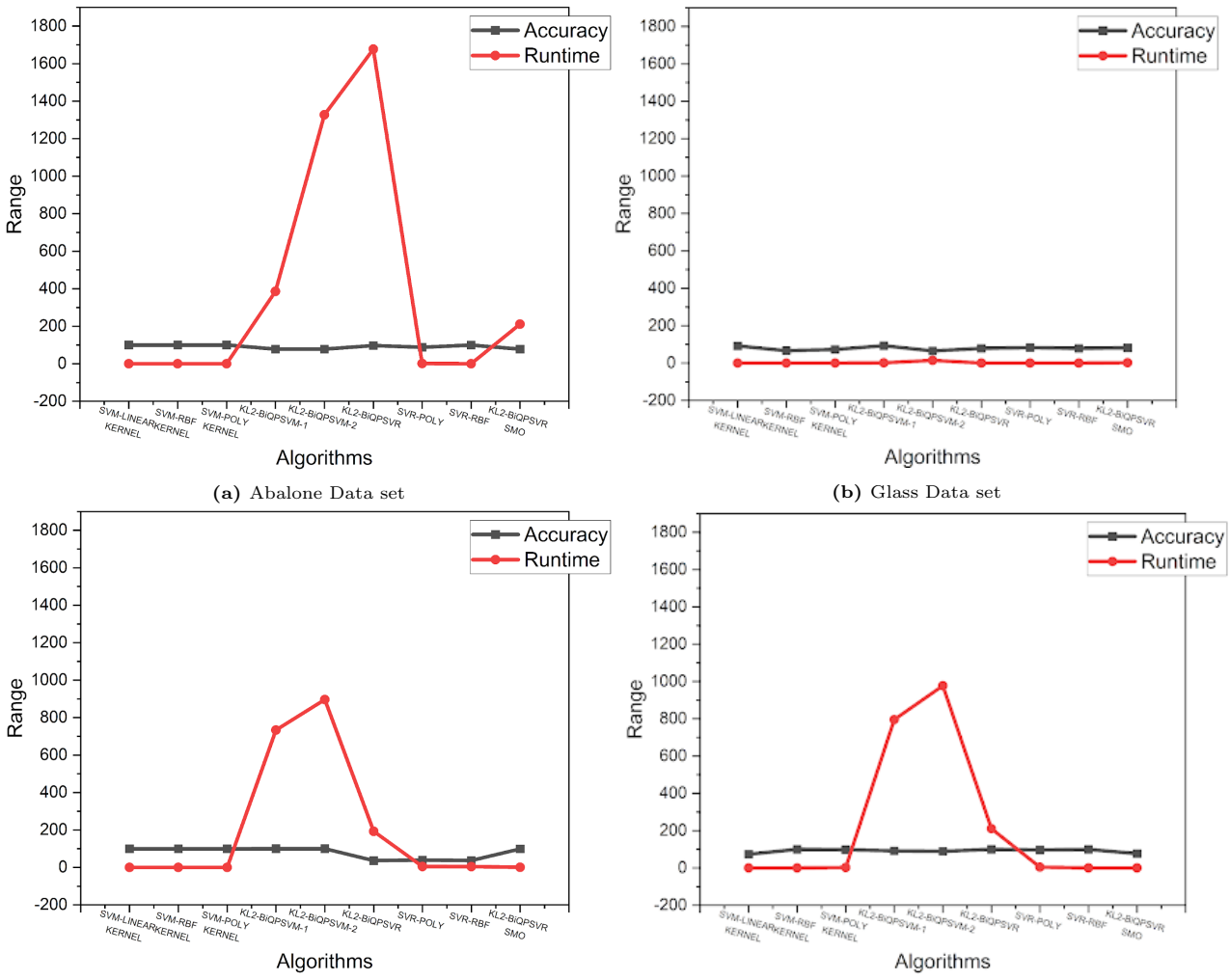
4.1. Experiment Settings

Results for accuracy and the time complexity has been compared among different kinds of algorithms. Several algorithms like kernel free and kernel based algorithm has been used on classification as well as regression type datasets.
Here SVM-LINEAR KERNEL, SVM-RBF KERNEL,SVM-POLY KERNEL,KL2-BiQPSVM-1(Without reinforcement), KL2-BiQPSVM-2(With reinforcement), KL2-BiQPSVR, SVR-POLY, SVR-RBF, KL2-BiQPSVM or SVR-SMO algorithms have been tested and computed for 4 regression and 4 classification based datasets. As the results in figure 4 describes that the run-time reduced significantly. The L_1 norm type model is optimized using numerically(LIBSVM) model while the L_2 norm is optimized using (analytically)quadratic programming method. All the parameters for each algorithms are listed on the table and the parameters are optimized using grid search.

Data set descriptions are given in Table 1.

Table 1: Data-set Description

	Datasets	Types	Instances	Conditional Attributes	Decision Classes
1	Iris	Classification	150	4	3
2	Wholesale	Classification	440	8	3
3	Credit	Classification	690	15	3
4	TaxInfo	Classification	1004	10	3
5	Glass	Regression	214	11	
6	wine	Regression	4898	12	
7	Car Evaluation	Regression	1728	6	
8	abalone	Regression	4177	8	



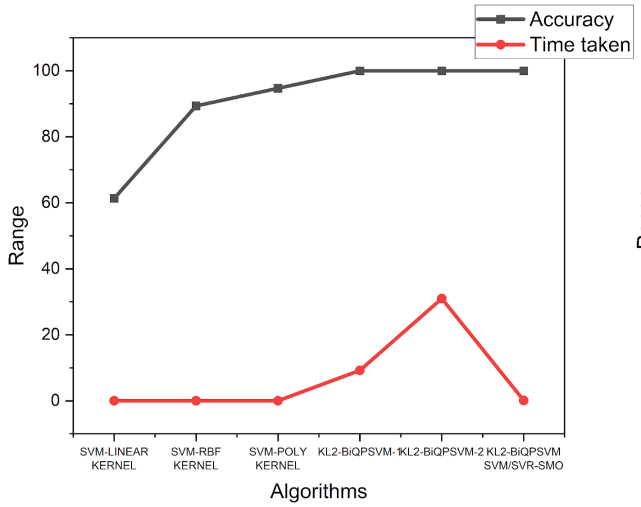
(a) Abalone Data set

(b) Glass Data set

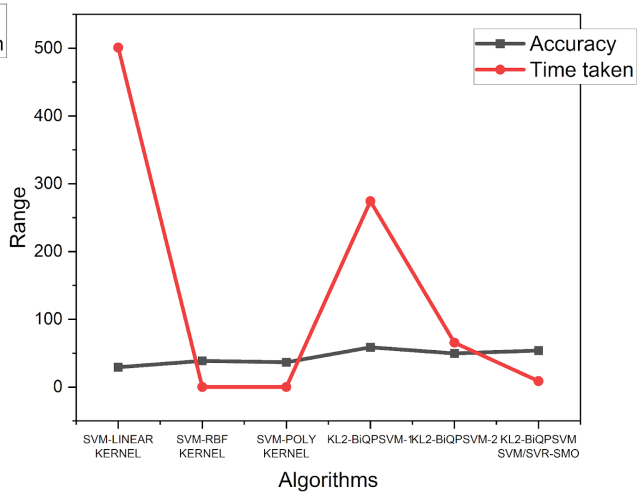
(c) Wine Quality data set

(d) Car Evaluation data set

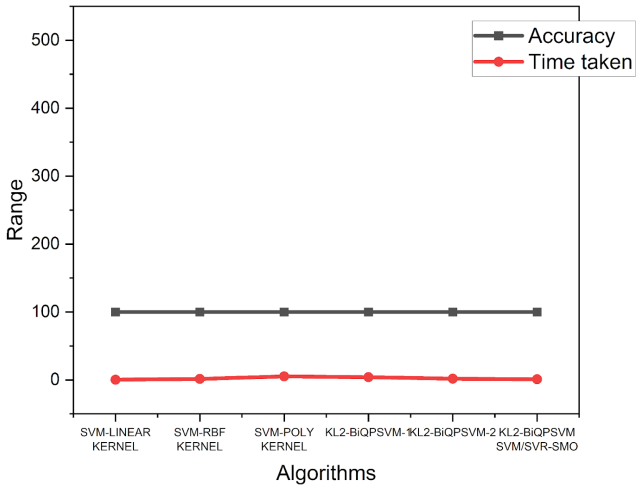
Fig. 1. plots of Regression based results



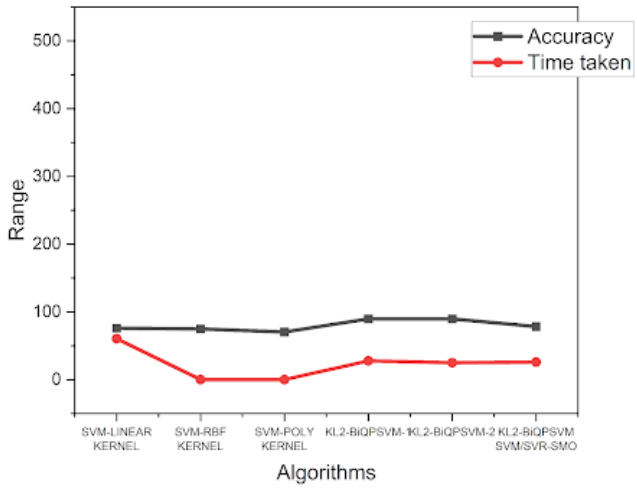
(a) Iris Data set



(b) Tax Info Data set

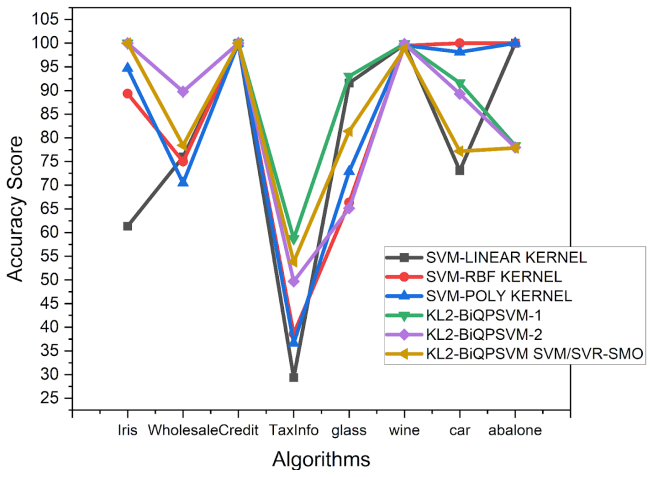


(c) Credit Data set

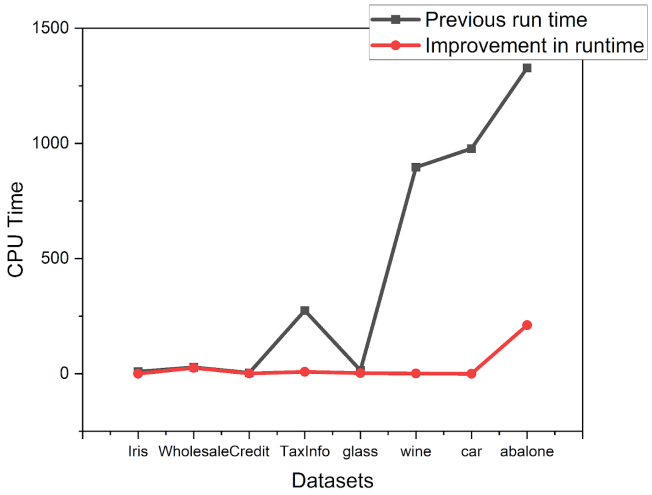


(d) Wholesale Data set

Fig. 2. plots of Classification based results

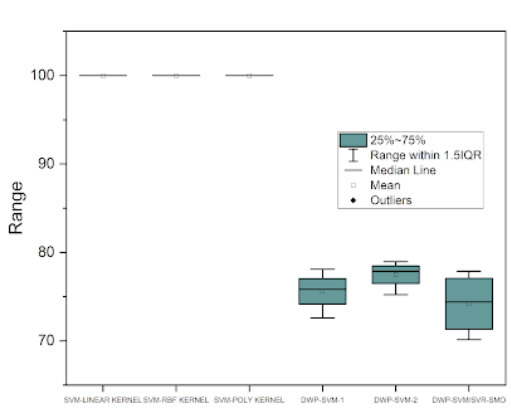


(a) Accuracy comparison on Dataset

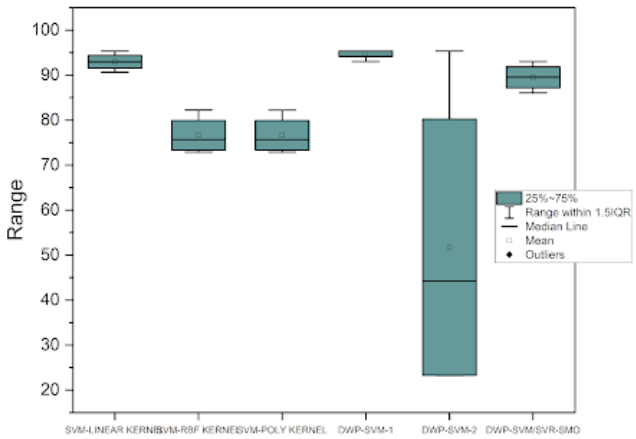


(b) Runtime comparison on Dataset

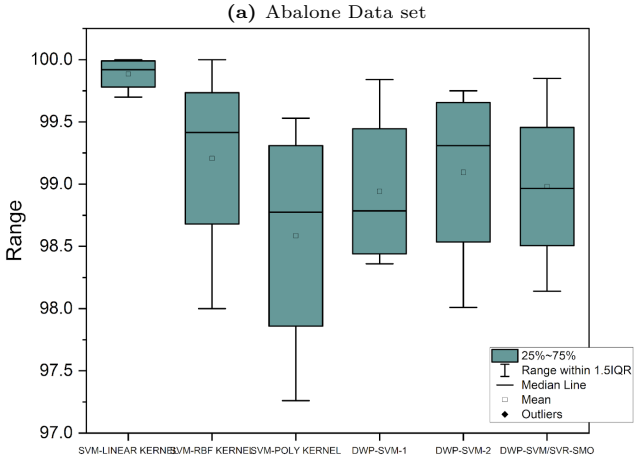
Fig. 3. Result Comparisons on Datasets



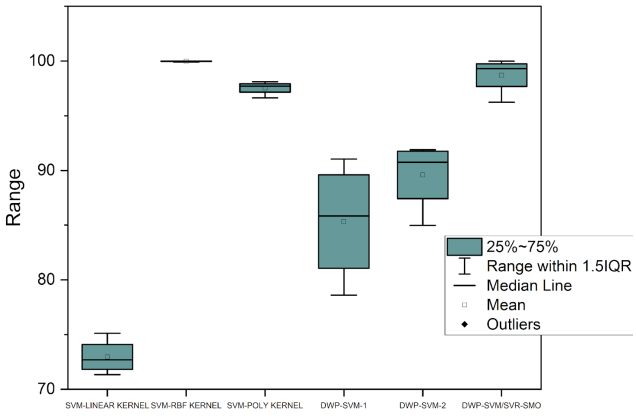
(a) Abalone Data set



(b) Glass Data set

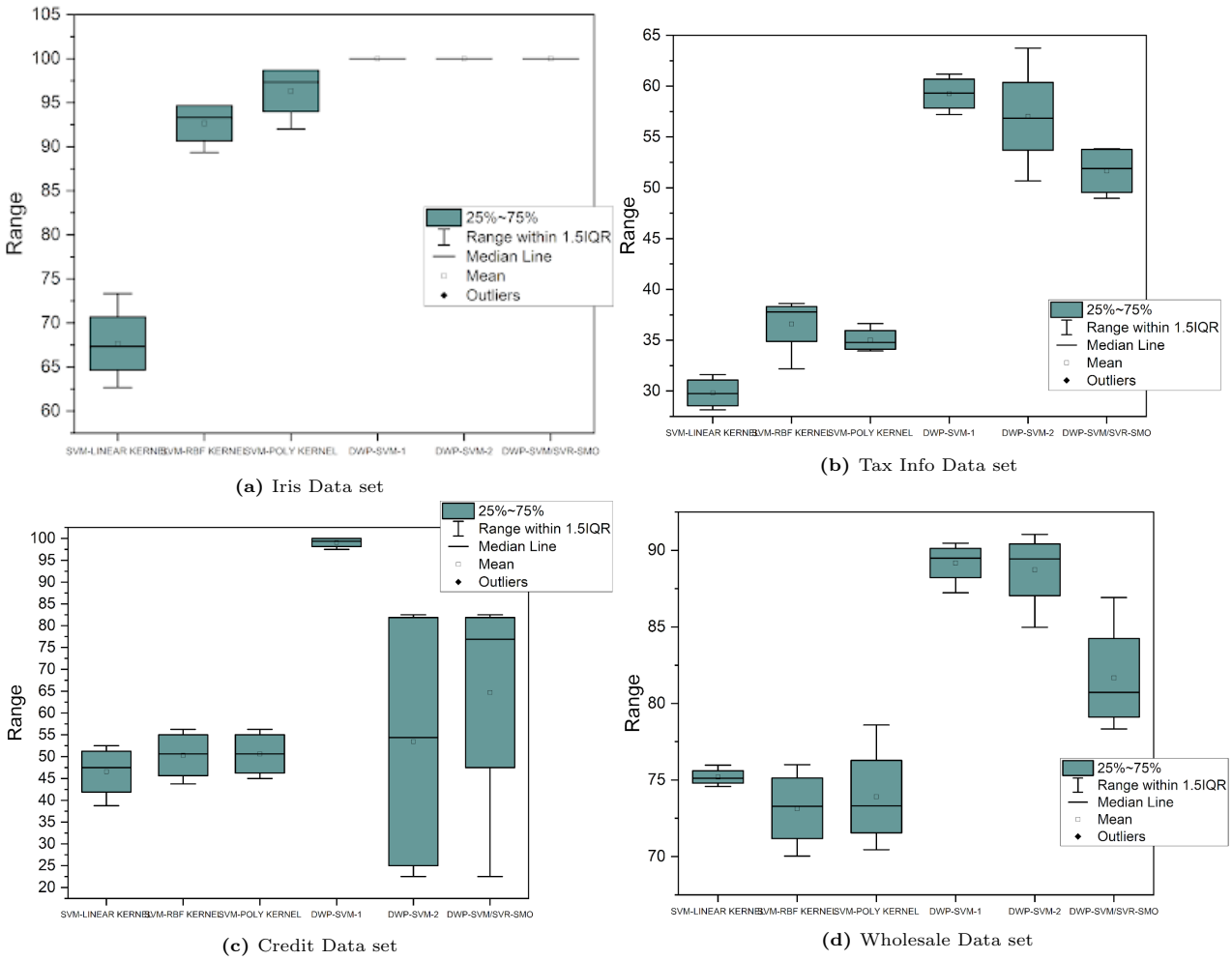


(c) Wine Quality data set



(d) Car Evaluation data set

Fig. 4. Box Plot analysis on the Regression Datasets



(a) Iris Data set

(b) Tax Info Data set

(c) Credit Data set

(d) Wholesale Data set

Fig. 5. Box Plot analysis on the Classification Datasets

Table 2: Algorithm Accuracy Table

	Datasets	SVM-LINEAR KERNEL	SVM-RBF KERNEL	SVM-POLY KERNEL	KL2-BiQPSVM-1	KL2-BiQPSVM-2	KL2-BiQPSVM/SVR SMO
1	Iris	61.34	89.34	94.67	100	100	100
2	Wholesale	75.97	75	70.45	89.78	89.78	78.41
3	Credit	100	100	100	100	100	100
4	TaxInfo	29.37	38.61	36.64	58.7	49.67	53.85
5	glass	91.58	66.35	72.89	93.02	65.11	81.4
6	wine	99.72	99.47	99.53	99.84	99.84	98.85
7	car	73.08	100	98.11	91.61	89.3	77.16
8	abalone	100	100	100	78.53	77.99	77.87

Table 3: Runtime Comparison Table

	Datasets	Previous Runtime	Improved Runtime
1	Iris	9.216	0.131
2	Wholesale	27.75	25.96
3	Credit	4.042	0.985
4	TaxInfo	274.36	8.7
5	glass	14.713	2.323
6	wine	896.259	0.98
7	car	977.35	0.259
8	abalone	1327.45	210.98

We can see from the tables that KL2-BiQPSVM gives better results than other kernel fee or kernel SVM algorithms and our model takes a very less run time because of application of SMO Optimization algorithms.

Table 4: Observation table for statistical tests

	Datasets	KL2-BiQPSVM(Observation A)	KL2-BiQPSVM(Observation B)
1	Iris	89.30	77.16
2	Wholesale	77.99	77.87
3	Credit	89.78	78.41
4	TaxInfo	49.67	53.85
5	glass	65.11	81.40
6	wine	99.84	98.85
7	car	89.3	77.16
8	abalone	77.99	77.87

Table 5: Mann-Whitney Test Table

	Test Statistic(W)	p-value	Null Hypothesis	Confidence Interval
1	34	0.873903	0.0	95

Table 6: Wilcoxon Signed-Rank Test Table

	W-value	Mean Difference	Sum of Positive ranks	Sum of Negative ranks	Z-Value
1	9	7.4	12	9	-0.3145

5. Conclusions

7 In terms of classification accuracy, the proposed reinforced KL2-BiQPSVM model performs better than other well-known SVM models. The proposed model showed dominant performance on most of the public

benchmark data sets.
The numerical results on public data also indicated the increasing dominance of reinforced KL2-BiQPSVM over other tested models as the number of data features increases.The proposed model showed its stable effectiveness and acceptable efficiency in credit scoring. This shows the potential of KL2-BiQPSVM in handling real-life classification problems.
Unlike other kernel based nonlinear SVM models, this model does not require any kernel functions or tuning their relative parameters. It saves considerable effort in the training process.Our investigation of the proposed reinforced KL2-BiQPSVM model for binary classification indicates some additional research works as follows.First, compared with kernel-based SVM models,the training CPU time of the reinforced KL2-BiQPSVM models bigger for large-sized data sets.The kernel-based SVM models were implemented by using LIBSVM[7],but the codes for implementing the reinforced KL2-BiQPSVM with SMO algorithm were written from scratch.So an immediate future work is to optimize the code for rapid computations.Moreover the proposed model can be extended for other real-world applications including image processing applications.

Acknowledgment

All the benchmark data sets come from three sources: UCI machine learning repository (Dua Graff, 2017), Kaggle, and Hsu, Chang, Lin et al. (2003).Acquired from the University of California, Irvine (UCI) Machine Learning Repository `https://archive.ics.uci.edu/ml/index.php`.All the results plotted in Origin 2021b software.

References

[1] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (3) (1995) 273–297.

[2] J. Platt, Sequential minimal optimization: A fast algorithm for training support vector machines (1998).

[3] J. Luo, X. Yan, Y. Tian, Unsupervised quadratic surface support vector machine with application to credit risk assessment, European Journal of Operational Research 280 (3) (2020) 1008–1017.

[4] Z. Gao, S.-C. Fang, J. Luo, N. Medhin, A kernel-free double well potential support vector machine with applications, European Journal of Operational Research 290 (1) (2021) 248–262.

[5] G. Bolondi, F. Ferretti, A. Maffia, Monomials and polynomials: the long march towards a definition, Teaching Mathematics and its Applications: An International Journal of the IMA 39 (1) (2020) 1–12.

[6] X. Peng, Y. Wang, D. Xu, Structural twin parametric-margin support vector machine for binary classification, Knowledge-Based Systems 49 (2013) 63–72.

[7] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM transactions on intelligent systems and technology (TIST) 2 (3) (2011) 1–27.