Support Vector Regression via a Combined Reward Cum Penalty Loss Function

Pritam Anand, Reshma Rastogi and Suresh Chandra.

Abstract—In this paper, we introduce a novel combined reward cum penalty loss function to handle the regression problem. The proposed combined reward cum penalty loss function penalizes the data points which lie outside the $\epsilon$-tube of the regressor and also assigns reward for the data points which lie inside of the $\epsilon$-tube of the regressor. The combined reward cum penalty loss function based regression (RP-$\epsilon$-SVR) model has several interesting properties which are investigated in this paper and are also supported with the experimental results.

Index Terms—Regression, loss function, Support Vector Regression, sparsity, robustness, noise distribution.

I. INTRODUCTION

Past few decades have witnessed the evolution of the Support Vector Regression (SVR) models (Vapnik et al. [1], Drucker et al. [2], Smola and Scholkopf [3], Gunn [4], Vapnik [5]) as a promising tool for handling the problem of function approximation. It has been successfully used in a wide variety of applications, e.g. [6] to [7]. SVR models have also been extended in non-parallel framework e.g. [8] to [9].

Given a training set $T = \{(x_i, y_i) : x_i \in \mathbb{R}^n, y_i \in \mathbb{R}, i = 1, 2, \ldots, l\}$, a typical SVR model determines a regressor $f(x) = w^T \phi(x) + b, w \in \mathbb{R}^n, b \in \mathbb{R}$ in feature space for predicting the response of an unseen test point. It uses the training set to minimize the empirical risk. In addition to this, it also minimizes a regularization term in its optimization problem for minimizing the structural risk.

There exist several SVR models in the literature. These models commonly use different types of loss functions to measure their empirical risk along with different types of regularizations. Some of SVR models which uses the convex loss function are as follows.

(i) The standard $\epsilon$-SVR model (Drucker et al. [2]) uses the $\epsilon$-insensitive loss function to measure the empirical risk with the regularization term $\frac{1}{2} w^T w$.

(ii) The standard Least Squares Support Vector Regression (LS-SVR) model (Suykens and Vandewalle [10]) uses the quadratic loss function to measure the empirical risk along with the regularization term $\frac{1}{2} w^T w$.

(iii) Maximum Likelihood Optimal and Robust Support Vector Regression model (Karal [11]) uses the Incosh loss function to measure the empirical risk with the regularization term $\frac{1}{2} w^T w$.

(iv) Huber loss function based SVR (Gunn [4]) uses the Huber loss function to measure the empirical risk along with the regularization term $\frac{1}{2} w^T w$.

(v) $L_1$-norm SVR (Tanveer et al. [12]) uses the $\epsilon$-insensitive loss function for measuring the empirical risk with the $L_1$-norm regularization term $\frac{1}{2} ||w||_1$.

(vi) Large-margin Distribution Machine based Regression (LDMR) model (Rastogi et al. [13]) uses a linear combination of the $\epsilon$-insensitive loss function and the quadratic loss function for measuring the empirical risk with the $L_2$-norm regularization.

(vii) Penalizing- $\epsilon$-generalized SVR (Anand et al., [14]) uses the generalized $\epsilon$-loss function to measure the empirical risk along with the regularization term $\frac{1}{2} w^T w$.

Apart from the above convex loss functions, some non-convex loss functions have also been used by researchers in SVR models. Some important of them are smooth Ramp loss function (Zhao and Sun, [15]) non convex least square loss function (Wang and Zhong, [16]), non convex generalized loss function (Wang et al., [17]) generalized quantile loss (Yang et al., [18]) and rescaled expectile loss (Yang et al., [19]). However, the optimization problem of non convex loss function based SVR models are algorithmically complex and computationally expensive.

A particular choice of loss function in a SVR model enables it to obtain optimal estimate for a particular type of noise model. For example, the use of quadratic loss function in LS-SVR model enables it to perform optimal for the normal noise. But, an ideal SVR model is expected to be robust i.e. it should perform well without bothering the nature of noise present in data. Apart from this, it should perform optimal for a wide family of noise distributions and manage to obtain sparse solution as well.

In existing SVR models, the standard $\epsilon$-SVR model is the most popular one. It is because of the fact that, it is a robust SVR model and can also manage to obtain the sparse solution.

The standard $\epsilon$-SVR model minimizes the regularization $\frac{1}{2} w^T w$ to make the estimated regressor as flat as possible along with $\epsilon$-insensitive loss function to minimize the empirical risk. The $\epsilon$-insensitive loss function is given as follows

$$L_\epsilon(y_i, x_i, f(x_i)) = \begin{cases} |y_i - f(x_i)| - \epsilon, & \text{if } |y_i - f(x_i)| \geq \epsilon, \\ 0, & \text{otherwise}, \end{cases}$$

(1)

where $\epsilon \geq 0$ is a parameter. The use of $\epsilon$-insensitive loss function in standard $\epsilon$-SVR model makes it to ignore those data
points which lie inside the \( \epsilon \)-tube of the regressor \( f(x) \). The data points which lie outside the \( \epsilon \)-tube are penalized in the optimization problem to bring them close to the \( \epsilon \)-tube. These data points along with the data points lying on the boundary of the \( \epsilon \)-tube of \( f(x) \) constitute ‘support vectors’ which only decide the orientation and position of the regressor \( f(x) \).

The use of the \( \epsilon \)-insensitive loss function in the \( \epsilon \)-SVR model enables it to obtain a robust and sparse solution. But, it also causes it to lose most of the information contained in the training set in the sense that data points lying inside of the \( \epsilon \)-tube are ignored in the construction of regressor. Further, the performance of the \( \epsilon \)-SVR model is subjected to having a right choice of the value of the \( \epsilon \). A wrong choice of \( \epsilon \) may result in the loss of significant part of the information contained in the training set and can lead to poor generalization ability.

We require a SVR model which can properly use the training set and can also preserve the elegance of the \( \epsilon \)-SVR model simultaneously. Taking motivation from this, we propose a new convex loss function termed as ‘reward cum penalty loss function’. Unlike the existing loss function, the proposed reward cum penalty loss function can take both positive and negative values. Here, a positive value represents ‘penalty’ and a negative value represent ‘reward’. It penalizes data points which lie outside the \( \epsilon \)-tube and can lead to poor generalization ability. A wrong choice of \( \epsilon \) may result in the loss of significant part of the information contained in the training set.

The proposed reward cum penalty loss function is given by

\[
RP_{\tau_1, \tau_2, \epsilon}(u) = \max(\tau_2(|u| - \epsilon), \tau_1(|u| - \epsilon)),
\]

where \( \tau_2, \tau_1 \) and \( \epsilon \geq 0 \) are parameters. For the regression training set \( T = \{(x_i, y_i) : x_i \in \mathbb{R}^n, y_i \in \mathbb{R}, i = 1, 2, ..., l\} \), the above proposed loss function can be used to measure the empirical error as follow

\[
RP_{\tau_1, \tau_2, \epsilon}(y_i, x_i, f(x_i)) = \begin{cases} 
\tau_2(|y_i - f(x_i)| - \epsilon), & \text{if } |y_i - f(x_i)| \geq \epsilon, \\
\tau_1(|y_i - f(x_i)| - \epsilon), & \text{otherwise,}
\end{cases}
\]

where \( \tau_2 \geq \tau_1 \) and \( \epsilon > 0 \) are parameters. Figure 1 shows the graph of a typical reward cum penalty loss function for different values of \( \tau_2 \geq \tau_1 \geq 0 \). The proposed reward cum penalty loss function reduces to the popular \( \epsilon \)-insensitive loss function for \( \tau_2 = 1 \) and \( \tau_1 = 0 \). Figure 2 shows the graph of reward cum penalty loss for different values of \( \tau_2 \geq \tau_1 \leq 0 \). It can also be observed that the proposed reward cum penalty loss function is a convex function for \( \tau_2 \geq \tau_1 \geq 0 \) but, for \( \tau_1 \leq 0 \), it loses its convexity. Therefore, in our subsequent discussion we shall always assume \( \tau_2 \geq \tau_1 \geq 0 \).

To build the regression model based on the proposed reward cum penalty loss function, we use the same for measuring the empirical risk of the training set which is minimized in the proposed optimization problem along with the regularization term \( \frac{1}{2}w^Tw \). We term the resulting regression model as ‘Reward cum Penalty loss function based \( \epsilon \)-Support Vector Regression (RP-\( \epsilon \)-SVR)’ model. Following are some salient features of the proposed reward cum penalty loss function and resulting RP-\( \epsilon \)-SVR model.
We now describe notations used in the rest of this paper. All vectors are taken as column vectors unless it has been specified otherwise. For any vector $x \in \mathbb{R}^n$, $|x|$ denotes the $L_2$ norm. A vector of ones of arbitrary dimension is denoted by $e$. $(A, Y)$ denotes the training set where $A = [A_1, A_2, \ldots, A_l]$ contains the $l$ points in $\mathbb{R}^n$ represented by $l$ rows of the matrix $A$ and $Y = [y_1; y_2; \ldots; y_l] \in \mathbb{R}^{1 \times l}$ contains the corresponding label or response value of the row of matrix $A$. Further, $\xi_1 = (\xi_1^1; \xi_1^2; \ldots; \xi_1^l)$, $\xi_2 = (\xi_2^1; \xi_2^2; \ldots; \xi_2^l)$ and $\xi = (\xi^1; \xi^2; \ldots; \xi_l)$ are $l$ dimensional column vectors which will be used to denote the errors.

The rest of this paper has been organized as follows. Section III briefly describes existing $\epsilon$-SVR model. In Section IV, the proposed RP-$\epsilon$-SVR model has been formulated for its linear and non-linear cases. In Section V, we have theoretically established the robustness, sparsity and general nature of the proposed RP-$\epsilon$-SVR model. Section VI evaluates the proposed RP-$\epsilon$-SVR model using the numerical results which is obtained by the experiments carried on several artificial and UCI benchmark datasets. Section VII concludes this paper.
II. \( \varepsilon \)-SUPPORT VECTOR REGRESSION

The standard \( \varepsilon \)-SVR minimizes
\[
\frac{1}{2} ||w||^2 + C \sum_{i=1}^{l} L_{\varepsilon}(y_i, x_i, f(x_i)),
\]
which can be equivalently converted to the following Quadratic Programming Problem (QPP)
\[
\min_{w, b, \xi_1, \xi_2} \frac{1}{2} ||w||^2 + Ce^T(\xi_1 + \xi_2)
\]
subject to,
\[
\sum_{i=1}^{l} x_i \xi_i - \langle w, x \rangle \leq \varepsilon + \xi_1, \quad (Aw + c) - \langle w, x \rangle \leq \varepsilon + \xi_2, \\
\xi_1 \geq 0, \quad \xi_2 \geq 0.
\]

Here \( C > 0 \) is the user specified positive parameter that balances the trade off between the training error and the flatness of the approximating function. To solve the primal problem (4) efficiently, we write the corresponding Wolfe dual (Mangasarian, [21]) using Karush-Kuhn-Tucker (KKT) conditions. The Wolfe dual of the primal problem (4) has been obtained as follows.
\[
\begin{align*}
\min_{(\beta_1, \beta_2)} & \frac{1}{2} (\beta_1 - \beta_2) A A^T (\beta_1 - \beta_2) - (\beta_1 - \beta_2)^T Y \\
\text{subject to,} & \quad (\beta_1 - \beta_2)^T e = 0, \\
& \quad 0 \leq \beta_1, \beta_2 \leq C.
\end{align*}
\]

After finding the optimal values of \( \beta_1 \) and \( \beta_2 \), the estimated value for the test point \( x \) is given by \( f(x) = (\beta_1 - \beta_2)^T A x + b \).

III. REWARD cum PENALTY \( \varepsilon \)-SUPPORT VECTOR REGRESSION

The RP-\( \varepsilon \)-SVR model minimizes
\[
\frac{1}{2} ||w||^2 + C \sum_{i=1}^{l} RP_\tau (\tau_1, \tau_2, \varepsilon) (y_i, x_i, f(x_i)) = \frac{1}{2} ||w||^2 + \\
C \sum_{i=1}^{l} \max(\tau_2(|y_i - f(x_i)| - \varepsilon), \tau_1(|y_i - f(x_i)| - \varepsilon)),
\]
where \( \tau_2 \geq \tau_1 \geq 0 \) and \( \varepsilon > 0 \) are parameters. Let us introduce a \( l \)-dimensional column error vector \( \xi \) where \( \xi^i = \max(\tau_2(|y_i - f(x_i)| - \varepsilon), \tau_1(|y_i - f(x_i)| - \varepsilon)) \) for \( i = 1, 2, \ldots, l \). Then problem (6) can be written as follows
\[
\min_{(w, b, \xi)} \frac{1}{2} ||w||^2 + Ce^T \xi
\]
subject to,
\[
\xi_i \geq \tau_2(|y_i - f(x_i)| - \varepsilon), \quad i = 1, 2, \ldots, l, \\
\xi_i \geq \tau_1(|y_i - f(x_i)| - \varepsilon), \quad i = 1, 2, \ldots, l.
\]

A. Linear Reward cum Penalty-\( \varepsilon \) SVR

The optimization problem (7) can be converted to the following standard QPP
\[
\min_{(w, b, \xi_1, \xi_2)} \frac{1}{2} ||w||^2 + Ce^T(\xi_1 + \xi_2)
\]
subject to,
\[
Y - (Aw + \varepsilon_2) \leq \varepsilon + \frac{1}{\tau_1} \xi_1, \\
(Aw + \varepsilon_2) - Y \leq \varepsilon + \frac{1}{\tau_2} \xi_2, \\
Y - (Aw + \varepsilon_1) \leq \varepsilon + \frac{1}{\tau_1} \xi_1, \\
(Aw + \varepsilon_1) - Y \leq \varepsilon + \frac{1}{\tau_2} \xi_2,
\]
where \( \xi_1 \) and \( \xi_2 \) are \( l \)-dimensional slack variables. The QPP (8) reduces to QPP (4) of the standard \( \varepsilon \)-SVR model with the particular choice of parameters \( \tau_2 = 1 \) and \( \tau_1 = 0 \). It makes the standard \( \varepsilon \)-SVR model a particular case of the proposed RP-\( \varepsilon \)-SVR formulation.

In order to find a solution of primal problem (8), we need to derive its Wolfe dual (Mangasarian, [21]). For this, we write the Lagrangian function for primal problem (8) as follows
\[
L(w, b, \xi_1, \xi_2, \alpha_1, \alpha_2, \beta_1, \beta_2) = \frac{1}{2} ||w||^2 + Ce^T(\xi_1 + \xi_2) + \\
\alpha_1^T(Y - (Aw + \varepsilon_2) - \varepsilon - \frac{1}{\tau_1} \xi_1) + \beta_1^T(Aw + \varepsilon_2 - Y - \varepsilon - \frac{1}{\tau_2} \xi_2),
\]
where \( \alpha_1 = (\alpha_1^1, \alpha_1^2, \ldots, \alpha_1^l), \beta_2 = (\beta_2^1, \beta_2^2, \ldots, \beta_2^l) \) and \( \beta_2 = (\beta_2^1, \beta_2^2, \ldots, \beta_2^l) \) are vectors of Lagrangian multipliers.

The KKT optimality conditions for the optimization problem (8) are given by
\[
\begin{align*}
\frac{\partial L}{\partial w} &= -A^T(\alpha_1 - \alpha_2 + \beta_1 - \beta_2) = 0, \\
\frac{\partial L}{\partial b} &= e^T(\alpha_1 - \alpha_2 + \beta_1 - \beta_2) = 0, \\
\frac{\partial L}{\partial \xi_1} &= C - \frac{1}{\tau_1} \alpha_1 - \frac{1}{\tau_2} \beta_1 = 0, \\
\frac{\partial L}{\partial \xi_2} &= C - \frac{1}{\tau_1} \alpha_2 - \frac{1}{\tau_2} \beta_2 = 0, \\
\alpha_1^T(Y - (Aw + \varepsilon_2) - \varepsilon - \frac{1}{\tau_1} \xi_1) &= 0, \\
\beta_1^T(Y - (Aw + \varepsilon_2) - \varepsilon - \frac{1}{\tau_2} \xi_2) &= 0, \\
\beta_2^T(Aw + \varepsilon_2 - Y - \varepsilon - \frac{1}{\tau_2} \xi_2) &= 0, \\
Y - (Aw + \varepsilon_1) &\leq \varepsilon + \frac{1}{\tau_1} \xi_1, \\
(Aw + \varepsilon_2) - Y &\leq \varepsilon + \frac{1}{\tau_2} \xi_2, \\
Y - (Aw + \varepsilon_1) &\leq \varepsilon + \frac{1}{\tau_1} \xi_1, \\
(Aw + \varepsilon_1) - Y &\leq \varepsilon + \frac{1}{\tau_2} \xi_2,
\end{align*}
\]
Using the above KKT conditions, the Wolfe dual (Mangasar-
ian, (21) of primal problem (8) can be obtained as follows
\[
\min_{(\alpha_1, \alpha_2, \beta_1, \beta_2)} \frac{1}{2} (\alpha_1 - \alpha_2 + \beta_1 - \beta_2)^T A A^T (\alpha_1 - \alpha_2 + \beta_1 - \beta_2) - (\alpha_1 - \alpha_2 + \beta_1 - \beta_2)^T Y + (\alpha_1 + \alpha_2 + \beta_1 + \beta_2)^T \varepsilon
\]
subject to,
\[
(\alpha_1 - \alpha_2 + \beta_1 - \beta_2)^T \varepsilon = 0,
\]
\[
C - \frac{1}{\tau_1} \alpha_1 - \frac{1}{\tau_2} \beta_1 = 0,
\]
\[
C - \frac{1}{\tau_1} \alpha_2 - \frac{1}{\tau_2} \beta_2 = 0,
\]
\[
\alpha_1, \alpha_2, \beta_1, \beta_2 \geq 0.
\]

After obtaining the solution of the dual problem (22), the value of \( w \) can be obtained from the KKT condition (9) as follows
\[
w = A^T (\alpha_1 - \alpha_2 + \beta_1 - \beta_2).
\] (23)

Let us now define the following sets
\[S_1 = \{ i : \alpha_1^i > 0, \beta_1^i > 0 \} \]
and
\[S_2 = \{ j : \alpha_2^j > 0, \beta_2^j > 0 \} \]

Then taking \( i \in S_1 \) and making use of the KKT conditions (13) and (15), we get
\[
y_i - (A_i w + b) - \varepsilon - \frac{1}{\tau_1} \xi_1^i = 0,
\] (24)
and
\[
y_i - (A_i w + b) + \varepsilon - \frac{1}{\tau_2} \xi_2^i = 0.
\] (25)

But (24) and (25) give \( \xi_1^i (\frac{1}{\tau_1} - \frac{1}{\tau_2}) = 0 \). Therefore for \( \tau_1 \neq \tau_2 \), we obtain
\[
b = y_i - A_i w - \varepsilon.
\] (26)

On similar lines, taking \( j \in S_2 \) and \( \tau_1 \neq \tau_2 \), we obtain
\[
b = y_j - A_j w + \varepsilon.
\] (27)

In practice, for each \( i \in S_1 \) and each \( j \in S_2 \), we calculate the values of \( b \) from (26) and (27) respectively and take their average value as the final value of \( b \). For the given test point \( x \in R^n \), the estimated response is obtained
\[
f(x) = w^T x + b = (\alpha_1 - \alpha_2 + \beta_1 - \beta_2)^T A x + b.
\] (28)

**B. Non-linear Reward cum Penalty-\( \epsilon \) SVR**

The non-linear RP-\( \epsilon \)-SVR model seeks to determine the regressor
\[
f(x) = w^T \phi(x) + b,
\]
where \( \phi : R^n \rightarrow H \) is a non-linear mapping and \( H \) is an appropriate higher dimensional feature space.

The non-linear RP-\( \epsilon \)-SVR model solves the following optimization problem
\[
\min_{w, \lambda} \frac{1}{2} ||w||^2 + C \varepsilon \sum_{i=1}^{l} \max(0, 1 - y_i f(x_i))
\]
subject to,
\[
Y - \phi(A)w - \varepsilon Y \leq \lambda,
\]
\[
\phi(A)w + \varepsilon Y \leq \lambda Y,
\]
\[
\phi(A)w = \lambda \varepsilon.
\] (29)

Similar to the linear RP-\( \epsilon \)-SVR model, the corresponding Wolfe dual (Mangasarian, 21) problem of the primal problem (29) is obtained as
\[
\min_{(\gamma_1, \gamma_2, \lambda_1, \lambda_2)} \frac{1}{2} (\gamma_1 - \gamma_2 + \lambda_1 - \lambda_2)^T \phi(A) \phi(A)^T (\gamma_1 - \gamma_2 + \lambda_1 - \lambda_2)
\] - (\gamma_1 - \gamma_2 + \lambda_1 - \lambda_2)^T Y + (\gamma_1 + \gamma_2 + \lambda_1 + \lambda_2)^T \varepsilon
subject to,
\[
(\gamma_1 - \gamma_2 + \lambda_1 - \lambda_2)^T \varepsilon = 0,
\]
\[
C - \frac{1}{\tau_1} \gamma_1 - \frac{1}{\tau_2} \lambda_1 = 0,
\]
\[
C - \frac{1}{\tau_1} \gamma_2 - \frac{1}{\tau_2} \lambda_2 = 0,
\]
\[
\gamma_1, \gamma_2, \lambda_1, \lambda_2 \geq 0.
\] (30)

A positive definite kernel \( K(A, A^T) \), satisfying the Mercer condition (Scholkopf and Smola, 24), is used to obtain \( \phi(A) \phi(A)^T \) without explicit knowledge of mapping \( \phi \). Thus problem (30) reduces to
\[
\min_{(\gamma_1, \gamma_2, \lambda_1, \lambda_2)} \frac{1}{2} (\gamma_1 - \gamma_2 + \lambda_1 - \lambda_2)^T K(A, A^T) (\gamma_1 - \gamma_2 + \lambda_1 - \lambda_2)
\] - (\gamma_1 - \gamma_2 + \lambda_1 - \lambda_2)^T Y + (\gamma_1 + \gamma_2 + \lambda_1 + \lambda_2)^T \varepsilon
subject to,
\[
(\gamma_1 - \gamma_2 + \lambda_1 - \lambda_2)^T \varepsilon = 0,
\]
\[
C - \frac{1}{\tau_1} \gamma_1 - \frac{1}{\tau_2} \lambda_1 = 0,
\]
\[
C - \frac{1}{\tau_1} \gamma_2 - \frac{1}{\tau_2} \lambda_2 = 0,
\]
\[
\gamma_1, \gamma_2, \lambda_1, \lambda_2 \geq 0.
\] (31)

For the given test point \( x \in R^n \), the determined regressor gives the value
\[
f(x) = w^T \phi(x) + b = (\gamma_1 - \gamma_2 + \lambda_1 - \lambda_2) K(A, x) + b.
\] (32)

**IV. PROPERTIES OF PROPOSED RP-\( \epsilon \)-SVR MODEL**

**A. Maximal likelihood approach and loss functions**

Let \( T = \{ (x_i, y_i), x_i \in R^n, y_i \in R, i = 1, 2, ..., l \} \) be the given training set. It is assumed that values \( (x_i, y_i) \) are related by unknown function \( f \) such that
\[
y_i = f(x_i) + \xi_i,
\] (33)
where \( \xi_i \) are independent and identically distributed random variables form an unknown distribution \( p(\xi) \). The celebrated Statistical Learning Theory (Vapnik, 5) employs the maximal likelihood principle to derive the ‘optimal’ loss function for a given distribution function \( p(\xi) \). This ‘optimal’ loss function is used to determine the regressor \( f \) for the estimation of the response \( y_j \) for a given test data point \( x_j \). Here the ‘optimal’ is understood in terms of maximizing the ‘likelihood function’ for the given training set \( T \), which is given by
\[
p[T / f] = \prod_{i=1}^{l} p(y_i - f(x_i)) = \prod_{i=1}^{l} p(\xi_i).
\] (34)

Since \( p(\xi_i) \geq 0 \) for all \( i \), the maximization of the likelihood function (34) is equivalent to the maximization of the log of the likelihood function. Therefore (34) is equivalent to
\[
\min \sum_{i=1}^{l} -log(p(\xi_i)).
\] (35)
Now the specific assumption about the density of noise model will specify the computed loss function which should be used for measuring the empirical error for finding the estimator function \( f \). We describe following robust densities of noise which lead to different popular loss functions.

(i) Laplace noise distribution:
This noise model is given by
\[
p(\xi) \propto \frac{1}{2}e^{-|\xi|}, \quad \xi \in \mathbb{R}.
\] (36)

On substituting (36) into (35), we get
\[
\min \sum_{i=1}^{l} |\xi_i|,
\] (37)
which is equivalent to the minimization of the Laplace loss \( L(\xi) = |\xi| \) for the training set \( T \). Fig 4 shows the Laplace loss function and its corresponding density function \( p(\xi) \).

(ii) Vapnik distribution:
It is one of the popular noise models used in the standard SVR formulation and is defined as
\[
p(\xi) \propto \frac{1}{2(1+\epsilon)}e^{-|\xi|\epsilon},
\] (38)

On substituting (38) into (35), we get
\[
\min \sum_{i=1}^{l} |\xi_i|\epsilon,
\] (39)

where
\[
|\xi_i|\epsilon = \begin{cases} 
0, & |\xi_i| < \epsilon, \\
|\xi_i| - \epsilon, & otherwise,
\end{cases}
\]
is the \( \epsilon \)-insensitive loss function used in the standard SVR formulation. Fig 5 shows the \( \epsilon \)-insensitive loss function and its corresponding density function.

(iii) Huber distribution:
It is a mixed noise model which is described as
\[
p(\xi) \propto \begin{cases} 
e^{-\frac{1}{2}\epsilon^2}, & if \ |\xi| < c, \\
e^{\frac{1}{2}(|\xi| - \epsilon)}, & otherwise.
\end{cases}
\] (40)

On substituting (40) in (35), we get
\[
\min L_{\epsilon}^{Huber} \sum_{i=1}^{l} \max(\tau_2(|\xi| - \epsilon), \tau_1(|\xi| - \epsilon)),
\] (41)

where
\[
L_{\epsilon}^{Huber} = \begin{cases} 
\sum_{i=1}^{l} \frac{1}{2\epsilon^2}(\xi_i)^2, & if \ |\xi| < c, \\
\sum_{i=1}^{l}(|\xi_i| - \frac{\epsilon}{2}), & otherwise,
\end{cases}
\] (42)
is Huber loss function. Fig 6 shows the Huber loss function and its corresponding density function.

(iv) Distribution of noise for the proposed reward cum penalty loss function:
We now present an analysis of above nature for our proposed reward cum penalty loss function \( RP_{\tau_1,\tau_2}(u) \).

Let us consider a noise model which follows the density function
\[
p(\xi) \propto \frac{1}{2((\tau_2 - \tau_1) + \epsilon)}e^{-\max(\tau_2(|u| - \epsilon), \tau_1(|u| - \epsilon))}.
\] (43)

Substituting (43) in (35) we get
\[
\min \sum_{i=1}^{l} \max(\tau_2(|u| - \epsilon), \tau_1(|u| - \epsilon)),
\] (44)

which is equivalent to the minimization of the proposed reward cum penalty loss function. Fig 7 shows the proposed reward cum penalty loss function and its corresponding density function.

Here, it is interesting to note that equation (43) represents a family of noise densities for different choices of \( \tau_1 \) and \( \tau_2 \). Therefore, as a consequence, (44) represents a family of loss function for different value of \( \tau_1 \) and \( \tau_2 \). In particular, the density function of Laplace distribution and Vapnik distribution belongs to the family of densities (43) with the particular choice of the parameters \( \tau_2 = \)}
1, \( \tau_1 = 0 \) and \( \tau_2 = 1, \tau_1 = 1 \) respectively. Hence, we can argue that the proposed loss function is a more general loss function in the sense that it is optimal to a wide range of noise models which also include the Vapnik and Laplace noise models.

### B. Sparsity of proposed RP-\( \epsilon \)-SVR model

**Proposition 1** For a given \( \tau_2 > \tau_1 \) and data point \((x_i, y_i)\), the \( \alpha^i_1 \beta^i_1 \neq 0 \) or \( \alpha^i_2 \beta^i_2 \neq 0 \) is possible only when it is lying on the boundary of the \( \epsilon \)-tube.

**Proof:** Let us consider first that \( \alpha^i_1 \beta^i_1 \neq 0 \). It is possible only when \( \alpha^i_1 \) and \( \beta^i_1 \) > 0. For \( \alpha^i_1 \) and \( \beta^i_1 > 0 \), we can obtain as follows

\[
(Y_i - (A_i w + b) - \epsilon - \frac{1}{\tau_1} \xi^i_1) = 0 \quad (45)
\]

\[
(Y_i - (A_i w + b) - \epsilon - \frac{1}{\tau_2} \xi^i_2) = 0 \quad (46)
\]

from KKT condition \([13]\) and \([15]\) respectively. After solving the equation \([45] \) and \([46]\), we get \( \xi^i_1 = 0 \) as \( \tau_2 \neq 1 \). It implies that \( Y_i - (A_i w + b) = \epsilon \) which means that the response point \( y_i \) for data point \((x_i, y_i)\) is lying on the upper boundary of the \( \epsilon \)-tube.

On the similar line, we can consider \( \alpha^i_2 \beta^i_2 \neq 0 \) and can obtain \( \xi^i_2 = 0 \) from the KKT condition \([14]\) and \([16]\). It means that the response point \( y_i \) for data point \((x_i, y_i)\) is lying on the lower boundary of the \( \epsilon \)-tube.

The contra-positive statement equivalent to the Proposition-1 is as follow. **For any data point \((x_i, y_i)\), which is not lying on the boundary of the \( \epsilon \)-tube, i.e. lying inside or outside of the \( \epsilon \)-tube, the \( \alpha^i_1 \beta^i_1 = 0 \) and \( \alpha^i_2 \beta^i_2 = 0 \) will hold true.**

**Proposition 2** For a given \( \tau_2 > \tau_1 \), any data point \((x_i, y_i)\) lying inside of the \( \epsilon \)-tube must satisfy \( \alpha^i_1 \beta^i_2 = 0 \) and \( \alpha^i_2 \beta^i_1 = 0 \).

**Proof** Since data point \((x_i, y_i)\) is lying inside of the \( \epsilon \)-tube, so it will satisfy

\[
Y_i - (A_i w + b) - \epsilon < 0 \quad (47)
\]

\[
(A_i w + b) - Y_i - \epsilon < 0. \quad (48)
\]

If possible, let us suppose that \( \alpha^i_1 \beta^i_2 = 0 \). It means that \( \alpha^i_1 > 0 \) and \( \beta^i_2 > 0 \) from which we can obtain

\[
\xi^i_1 = \tau_1 (Y_i - (A_i w + b) - \epsilon) \quad (49)
\]

\[
\xi^i_2 = \tau_2 ((A_i w + b) - Y_i - \epsilon). \quad (50)
\]

But, the KKT conditions \([18]\) is

\[
(A_i w + b) - Y_i \leq \epsilon + \frac{1}{\tau_1} \xi^i_1. \quad (51)
\]

After putting the value of the \( \xi^i_2 \) from \([50]\), we get

\[
(A_i w + b) - Y_i - \epsilon \leq \frac{\tau_2}{\tau_1}((A_i w + b) - Y_i - \epsilon) \quad (52)
\]

which is not possible as \( (A_i w + b) - Y_i - \epsilon < 0 \) and \( \tau_2 \geq 1 \). On the similar line, we can show that \( \alpha^i_2 \beta^i_1 = 0 \) as \( \alpha^i_2 > 0 \) and \( \beta^i_1 > 0 \) contradicts the KKT condition \([17]\).

**Proposition 3** For \( \tau_2 > \tau_1 \), all data points \((x_i, y_i)\), which lie inside of the \( \epsilon \)-tube, must satisfy \( \alpha^i_1 - \alpha^i_2 + \beta^i_1 - \beta^i_2 = 0 \).

**Proof** From the KKT condition \([11]\) and \([12]\), we can obtain

\[
\frac{1}{\tau_1} \alpha_1 + \frac{1}{\tau_2} \beta_1 = \frac{1}{\tau_1} \alpha_2 + \frac{1}{\tau_2} \beta_2 = C \quad (53)
\]

Also, from Proposition-1 and Proposition-2, we have

\[
\alpha^i_1 \beta^i_1 = 0, \quad \alpha^i_2 \beta^i_2 = 0. \quad (54)
\]

\[
\alpha^i_1 \beta^i_2 = 0, \quad \alpha^i_2 \beta^i_1 = 0. \quad (55)
\]

respectively. From which, we can infer that there will exist only one of possible following cases when a data point \((x_i, y_i)\) is lying inside of the \( \epsilon \)-tube for a given \( \tau_2 \geq 1 \).

(a) Only \( \alpha^i_1 \) and \( \alpha^i_2 \) takes non-zero values.

(b) Only \( \beta^i_1 \) and \( \beta^i_2 \) takes non-zero values.

But, in all of cases, we can get \((\alpha^i_1 - \alpha^i_2 + \beta^i_1 - \beta^i_2 = 0)\) from \([53]\). It completes the proof.

Though, the proposed RP-\( \epsilon \)-SVR model assigns a non-zero empirical risk with every training data point but, it can still obtain the sparse solution as we can obtain \((\alpha^i_1 - \alpha^i_2 + \beta^i_1 - \beta^i_2 = 0)\) for all training data points which lie inside of the \( \epsilon \)-tube.

### C. Robustness of the proposed \( \epsilon \)-penalty loss function

In this subsection, we shall show the robustness of proposed \( \epsilon \)-penalty loss function against outliers. For this, we shall be using the approach based on the influence function. This approach has been used to measure the robustness of loss functions in (Karal, [11]). It has also been shown in (Karal, [11]) that the influence function for the quadratic loss function is not bounded. However, the influence function of the popular \( \epsilon \)-insensitive loss function is bounded. A loss function which has bounded influence function is a desirable loss function for a regression model as it makes the regression model a robust regression model.

For a given training set, the loss function \( L(u) \) is used to measure the empirical risk as follows.

\[
E = \frac{1}{N} \sum_{i=1}^{N} L(u_i) \quad (56)
\]

Taking the gradient of \([56]\) with respect to the model parameter \( W \) will give

\[
\frac{\partial E}{\partial W} = \frac{1}{N} \sum_{i=1}^{N} \tilde{\phi}(u_i) \frac{\partial u_i}{\partial W} \quad (57)
\]
Here \( \phi(u) = \frac{\partial L(u)}{\partial u} \) is the influence function of loss function \( L(u) \) which determines the performance of the loss function. We can observe from [57] that the rate of change in the empirical risk is the weighted sum of \( \frac{\partial u_i}{\partial u} \) and \( \phi(u_i) \) stands for the weights for the data points.

The influence function of the proposed reward cum penalty loss function is given as follow

\[
\phi(u_i) = \begin{cases} 
\tau_1 \text{sign}(u_i), & \text{for } |u_i| \leq \epsilon, \\
\tau_2 \text{sign}(u_i), & \text{for } |u_i| > \epsilon,
\end{cases} \quad (58)
\]

where

\[
\text{sign}(u) = \begin{cases} 
-1, & \text{for } u \leq 0, \\
1, & \text{for } u > 0.
\end{cases} \quad (59)
\]

It means that, the influence function of the proposed \( \epsilon \)-penalty loss function is bounded in the interval \([-\tau_2, \tau_2]\) and the effect of any sample point is limited in the range \([-\tau_2, \tau_2]\).

V. Experimental Results

To study the behavior of the proposed RP-\( \epsilon \)-SVR model, we have tested it on eight artificial and ten real world UCI benchmark (Blake CL and Merz CJ [23]) datasets. The proposed RP-\( \epsilon \)-SVR is basically an improvement over the standard \( \epsilon \)-SVR model. Therefore, we have also compared the performance of the RP-\( \epsilon \)-SVR model with existing \( \epsilon \)-SVR model on these datasets. The numerical results on these datasets illustrate that irrespective of the nature of the noise present in these datasets, the proposed RP-\( \epsilon \)-SVR model always obtains better generalization ability than existing \( \epsilon \)-SVR model.

All regression methods presented here were simulated in MATLAB 16.0 environment (http://in.mathworks.com/) on Intel XEON processor with 16.0 GB RAM. The respective primal problems of the proposed RP-\( \epsilon \)-SVR and existing \( \epsilon \)-SVR models have same number of constraints and variables. However, the dual problem of the proposed RP-\( \epsilon \)-SVR model has \( 4l \) variables, \( 2l + 1 \) equality constraints and \( 4l \) inequality constraints, whereas the dual problem of the \( \epsilon \)-SVR model has \( 2l \) variables, \( 1 \) equality constraints and \( 2l \) inequality constraints. The dual QPPs of the proposed RP-\( \epsilon \)-SVR model and \( \epsilon \)-SVR model have been solved by using the 'quadprog' function of MATLAB (http://in.mathworks.com/) with its default algorithm in this paper. The development of an efficient algorithm for the solution of the QPP of the proposed RP-\( \epsilon \)-SVR model has been left as future work. Throughout the experiments, we have used RBF kernel \( \exp\left(-\frac{|x-y|^2}{q}\right) \) where \( q \) is the kernel parameter.

The optimal values of the parameters have been obtained using the exhaustive search method (Hsu and Lin [24]) by using cross-validation. The values of the parameter \( C \) and RBF kernel parameter \( q \) of \( \epsilon \)-SVR model have been tuned by searching in the set \( \{2^i, i = -10, -9, ..., 12\} \). The value of the parameter \( \epsilon \) of \( \epsilon \)-SVR model has been tuned by searching in the set \( \{0.05, 0.1, 0.2, 0.3, ..., 1.15, 2, 2.5, 3, 3.5, 4, 4.5, 5\} \). To have a fair comparisons with \( \epsilon \)-SVR model, we have not explicitly tuned the parameters \( q, C \) and \( \epsilon \) of the RP-\( \epsilon \)-SVR model rather, we have used the same values of these parameters which was obtained form the \( \epsilon \)-SVR model. We have only tuned the value of parameters \( \tau_1 \) and \( \tau_2 \) of the proposed RP-\( \epsilon \)-SVR model by searching in the set \( \{0.5, 0.6, ..., 2.5\} \) and \( \{0.1, 0.2, 0.3, ..., 1\} \), respectively.

A. Performance Criteria

For evaluating the performance of the regression methods, we introduce some commonly used evaluation criteria. Without loss of generality, let \( l \) and \( k \) be the number of the training samples and testing samples respectively. Furthermore, for \( i = 1, 2, ..., k, \) let \( y_i^l \) be the predicted value for the response value \( y_i \) and \( \bar{y} = \frac{1}{k} \sum_{i=1}^{k} y_i \) is the average of \( y_1, y_2, ..., y_k \). The definition and significance of the some evaluation criteria has been listed as follows.

(i) \( \text{SSE} \): Sum of squared error of testing, which is defined as \( \text{SSE} = \sum_{i=1}^{k} (y_i - y_i^l)^2 \). \( \text{SSE} \) represents the fitting precision.

(ii) \( \text{SST} \): Sum of squared deviation of testing samples, which is defined as \( \text{SST} = \sum_{i=1}^{k} (y_i - \bar{y})^2 \). \( \text{SST} \) shows the underlying variance of the testing samples.

(iii) \( \text{SSR} \): Sum of square deviation of the testing samples which can be explained by the estimated regressor. It is defined as \( \text{SSR} = \sum_{i=1}^{k} (y_i^l - \bar{y})^2 \).

(iv) \( \text{RMSE} \): Root mean square of the testing error, which is defined as \( \text{RMSE} = \sqrt{\frac{1}{k} \sum_{i=1}^{k} (y_i - y_i^l)^2} \).

(v) \( \text{MAE} \): Mean absolute error of testing, which is defined as \( \frac{1}{k} \sum_{i=1}^{k} |y_i - y_i^l| \).

(vi) \( \text{SSE/SST} \): \( \text{SSE/SST} \) is the ratio between the sum of the square of the testing error and sum of the square of the deviation of testing samples. In most cases, small \( \text{SSE/SST} \) means good agreement between estimations and real values.

(vii) \( \text{SSR/SST} \): It is the ratio between the variance obtained by the estimated regressor on testing samples and actual underlying variance of the testing samples.

(viii) \( \text{Sparsity} \% \): The sparsity of a vector \( u \) is defined as \( \text{Sparsity} \% \left(u\right) = \frac{\text{\#(r)}}{\text{\#(u)}} \times 100 \), where \( \text{\#(r)} \) determines the number of the component of the vector \( r \).

B. Artificial Datasets

We have synthesized some artificial datasets to show the efficacy of the proposed method over the other existing methods. To compare the noise-insensitivity of the regression methods, only training sets were added with different types of noises in these artificial datasets. For the training samples \( (x_i, y_i) \) for \( i = 1, 2, ..., l \), following types of datasets have been generated.

**TYPE 1**-

\[
y_i = \frac{\sin(x_i)}{x_i} + \xi_i, \quad \xi_i \sim U[-0.2, 0.2]
\]

and \( x_i \) is from \( U[-4\pi, 4\pi] \).

**TYPE 2**-

\[
y_i = \frac{\sin(x_i)}{x_i} + \xi_i, \quad \xi_i \sim U[-0.3, 0.3]
\]
and \( x_i \) is from \( U[-4\pi, 4\pi] \).

TYPE 3:-
\[
y_i = \frac{\sin(x_i)}{x_i} + \xi_i, \quad \xi_i \sim U[-0.4, 0.4]
\]
and \( x_i \) is from \( U[-4\pi, 4\pi] \).

TYPE 4:-
\[
y_i = \frac{\sin(x_i)}{x_i} + \xi_i, \quad \xi_i \sim N[0, 0.1]
\]
and \( x_i \) is from \( U[-4\pi, 4\pi] \).

TYPE 5:-
\[
y_i = \frac{\sin(x_i)}{x_i} + \xi_i, \quad \xi_i \sim N[0, 0.3]
\]
and \( x_i \) is from \( U[-4\pi, 4\pi] \).

TYPE 6:-
\[
y_i = \frac{\sin(x_i)}{x_i} + \xi_i, \quad \xi_i \sim N[0, 0.4]
\]
and \( x_i \) is from \( U[-4\pi, 4\pi] \).

TYPE 7:-
\[
y_i = \frac{|x_i - 1/4|}{4} + \sin(\pi(1 + x_i - 1/4)) + 1 + \xi_i,
\]
\( \xi_i \sim U[-0.4, 0.4] \) and \( x_i \) is from \( U[-4\pi, 4\pi] \).

TYPE 8:-
\[
y_i = \frac{|x_i - 1/4|}{4} + \sin(\pi(1 + x_i - 1/4)) + 1 + \xi_i,
\]
\( \xi_i \sim U[-0.6, 0.6] \) and \( x_i \) is from \( U[-4\pi, 4\pi] \).

All datasets contain 100 training samples with noise and 500 non-noise testing samples. To avoid the biased comparison, ten independent groups of noisy samples were generated randomly in MATLAB (http://in.mathworks.com/) for all type of datasets.

Table I lists the numerical results obtained from the experiments carried on the artificial datasets. We analyze the numerical results listed in Table I as follows.

(a) The numerical results show that irrespective of the evaluation criteria and nature of noise present in the artificial datasets, the proposed RP-\( \epsilon \)-SVR model always owns better generalization ability than existing \( \epsilon \)-SVR model. It also empirically verifies that proposed RP-\( \epsilon \)-SVR model is an improvement over the standard \( \epsilon \)-SVR model.

(b) To realize this improvement, we have also computed the percentage of decrease in SSE/SST, RMSE and MAE values obtained by RP-\( \epsilon \)-SVR model over \( \epsilon \)-SVR model on artificial datasets as
\[
\text{percentage of the decrease in value} = \frac{(\text{RP-}\epsilon-\text{SVR value} - \epsilon-\text{SVR value}) \times 100}{\epsilon-\text{SVR value}}
\]

Fig. 8: Performance of the RP-\( \epsilon \) SVR model over \( \epsilon \)-SVR model using different evaluation criteria on eight different artificial datasets listed in subsection (V-B) (represented by 1 to 8 on x-axis).
Figure 8 shows the comparison and obtained improvement of the RP-ε-SVR over the ε-SVR using different evaluation criteria on eight artificial datasets. The use of RP-ε-SVR model over the ε-SVR model always results significant improvement in the values of the SSE/SST, RMSE and MAE on artificial datasets. It is because of the fact that RP-ε-SVR model can properly use the information of training set. Figure 8(d) compares the sparsity of the solution vector of the RP-ε-SVR and ε-SVR model. We can realize that the sparsity of the solution vector of the RP-ε-SVR is still comparable with the existing ε-SVR model in the Table I, though it can properly utilize the full information of the training set.

(c) Table I also lists the tuned parameters of the ε-SVR model. The values of the parameter C, q and ϵ of RP-ε-SVR model have not been tuned explicitly. The tuned values of the ε-SVR model has been only supplied to the RP-ε-SVR model. It is noteworthy that, irrespective of the parameters values C, ϵ and q, tuned by the ε-SVR model, the proposed RP-ε-SVR model can find several τ1 and τ2 values on which it can outperform the ε-SVR model. Figure 9 shows the plot of the SSE/SST values obtained using the proposed RP-ε-SVR model against different τ1 values for a fixed value of the parameter τ2 on artificial datasets. It can be visualized that there exists several τ1 values for which the proposed RP-ε-SVR model obtains better SSE/SST values than ε-SVR model.

C. Benchmark datasets

For further evaluation, we have checked the performance of the proposed RP-ε-SVR model on UCI datasets namely, Yatch Hydro Dyanamics, Concrete Slump, Chwirut, Servo, Machine CPU, NO2, ENSO, HahnI and and AutoMpg. Yatch Hydro Dyanamics, Concrete Slump, Servo, Machine CPU, NO2, Automp and Nelson datasets were downloaded from UCI repository [23] (archive.ics.uci.edu/ml). ENSO, HahnI and Nelson datasets were downloaded from www.itl.nist.gov/div898/strd/nls/nls_main.shtml. For all the datasets, only feature vectors are normalized in the range of [0, 1]. Ten-fold cross validation (Duda and Hart [25]) method has been used to report the numerical results for these datasets.

Table II lists the numerical results obtained from the experiments carried on real-world benchmark datasets. The proposed RP-ε-SVR always performs better than ε-SVR model on several τ1 values on given datasets. The tuned parameters of the ε-SVR method is also listed for different datasets. Similar to the line of the numerical results for artificial datasets, we can also analyze the numerical results listed in Table II for benchmark datasets. Figure 10 shows the plot of the RMSE values obtained by the proposed RP-ε-SVR model against different τ1 values for the fixed value of the τ2 listed in the Table II on UCI datasets. The proposed RP-ε-SVR model can perform better than ε-SVR model on several τ1 values as the RP-ε-SVR model is more general model than ε-SVR model. The best value of the τ1 is different with datasets.

The proposed RP-ε-SVR model is basically an improvement
over popular and widely used \(\epsilon\)-SVR model. Therefore the numerical results presented in the Table I compares the proposed \(\rho\)-\(\epsilon\)-SVR model with the \(\epsilon\)-SVR model and are enough to empirically show that the proposed model is a better substitute of the \(\epsilon\)-SVR model. These numerical results also establishes the efficacy of the proposed reward cum penalty loss function over existing \(\epsilon\)-insensitive loss functions.

We have also compared the performance of proposed \(\rho\)-\(\epsilon\)-SVR model with some other existing traditional SVR models namely Huber SVR [4] and LS-SVR[10]. Further, we have also compared the proposed \(\rho\)-\(\epsilon\)-SVR model with some recent SVM models namely \(L_1\)-Norm SVM model[12] and LDMR model. The parameters of these models has also been tuned using Exhaustive search method[24] in their appropriate range.

For the comparison, we have picked up three more UCI datasets namely Boston Housing, Motorcycle and Wine quality (Red). Datasets were partitioned into the training set and testing set randomly ten times and numerical results were reported by taking the mean and variance of the obtained numbers. The cardinality of training set and testing set has been listed in the Table III. Table III also lists the comparison of the performance of the proposed \(\rho\)-\(\epsilon\)-SVR model and other traditional and recent SVM models along with the CPU time.
Fig. 9: Plot of the SSE/SST values obtained by the RP-\(\epsilon\)-SVR model against the \(\tau_1\) values on artificial datasets (a) TYPE 1 (b) TYPE 4 (c) TYPE 5 and (d) TYPE 6.

Fig. 10: Plot of the RMSE values obtained by the RP-\(\epsilon\)-SVR model using different \(\tau_1\) values on (a) Concrete Slump (b) Chwirut (c) Servo and (d) Hanh1 datasets.
It can be observed that the performance of the proposed RP-\(\epsilon\)-SVR model is not only better than standard \(\epsilon\)-SVR model but, it also outperforms the other existing SVR models.

VI. CONCLUSIONS

This paper proposes a novel reward cum penalty loss function for handling the regression problem. Unlike the other existing loss functions, it can also take negative values. Like \(\epsilon\)-insensitive loss function, the reward cum penalty loss function not only penalizes data points which lie outside the \(\epsilon\)-tube of the regressor \(f(x)\) but, it also assigns reward for the data points lying inside the \(\epsilon\)-tube. The trade-off between the reward and penalty can be controlled by the parameters \(\tau_1\) and \(\tau_2\). The reward cum penalty loss function has been judiciously used in the proposed RP-\(\epsilon\)-SVR model in such a way that it can always obtain the sparse solution. The proposed RP-\(\epsilon\)-SVR model is a direct improvement over the standard \(\epsilon\)-SVR model as it can properly use the full information of training set while preserving the robustness and sparsity of the solution. The standard \(\epsilon\)-SVR model is a particular case of the proposed RP-\(\epsilon\)-SVR model with choice of the parameters \(\tau_2 = 1\) and \(\tau_1 = 0\). Experimental results on several artificial and real world datasets show that the proposed RP-\(\epsilon\)-SVR model always owns better generalization ability than existing \(\epsilon\)-SVR model.

As compared to the standard \(\epsilon\)-SVR model, the RP-\(\epsilon\)-SVR model will be requiring to tune at least one extra parameter \(\tau_1\). The parameter \(\tau_2\) can be kept as constant and parameter \(C\) can be tuned appropriately instead. However, this extra tuning of parameter \(\tau_1\) in RP-\(\epsilon\)-SVR model makes its model selection time longer than \(\epsilon\)-SVR model.

There are some potential problems for future studies. We need a development of the fast algorithm to solve the optimization problem of the proposed RP-\(\epsilon\)-SVR model. It will make the RP-\(\epsilon\)-SVR model suitable for the large scale datasets. A traversal algorithm for finding the best \(\tau_1\) value in RP-\(\epsilon\)-SVR model is also required.

ACKNOWLEDGEMENT

This work was supported by the Ministry of Electronics and Information Technology Government of India under Visvesvaraya PhD Scheme for Electronics and IT Order No. Phd-MLA/4(42)/2015-16. We are also thankful to the Editor and the learned referee for their valuable comments.

REFERENCES

[1] V. Vapnik, S. E. Golowich, and A. J. Smola, “Support vector method for function approximation, regression estimation and signal processing,” in Advances in neural information processing systems, 1997, pp. 281–287.
[2] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, “Support vector regression machines,” in Advances in neural information processing systems, 1997, pp. 155–161.
[3] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” Statistics and computing, vol. 14, no. 3, pp. 199–222, 2004.
[4] S. Gunn, “Support vector machines for classification and regression,” ISIS technical report, 1998.
[5] V. Vapnik, The nature of statistical learning theory. Springer science & business media, 2013.
[6] M. A. Mohandes, T. O. Halawani, S. Rehman, and A. A. Hussain, “Support vector machines for wind speed prediction,” Renewable Energy, vol. 29, no. 6, pp. 939–947, 2004.
[7] C.-H. Wu, J.-M. Ho, and D.-T. Lee, “Travel-time prediction with support vector regression,” IEEE transactions on intelligent transportation systems, vol. 5, no. 4, pp. 276–281, 2004.
[8] X. Peng, “Tsvr: an efficient twin support vector machine for regression,” Neural Networks, vol. 23, no. 3, pp. 365–372, 2010.
[9] R. Khemchandani, K. Goyal, and S. Chandra, “Twsvr: regression via twin support vector machine,” Neural Networks, vol. 74, pp. 14–21, 2016.
[10] J. A. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” Neural processing letters, vol. 9, no. 3, pp. 293–300, 1999.
[11] O. Karal, “Maximum likelihood optimal and robust support vector regression with lncosh loss function,” Neural networks, vol. 94, pp. 1–12, 2017.
[12] M. Tanveer, M. Mangal, I. Ahmad, and Y.-H. Shao, “One norm linear programming support vector regression,” Neurocomputing, vol. 173, pp. 1508–1518, 2016.
[13] R. Rastogi, P. Anand, and S. Chandra, “Large-margin distribution machine-based regression,” Neural Computing and Applications, pp. 1–16, 2018.
[14] P. Anand, R. Rastogi, and S. Chandra, “Generalized loss function-based regression,” in Machine Intelligence and Signal Analysis. Springer, 2019, pp. 395–409.
[15] K. Wang and P. Zhong, “Robust support vector regression with flexible loss function,” International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 7, no. 4, pp. 211–220, 2014.
[16] ——, “Robust non-convex least squares loss function for regression with outliers,” Knowledge-Based Systems, vol. 71, pp. 290–302, 2014.
[17] K. Wang, W. Zhu, and P. Zhong, “Robust support vector regression with generalized loss function and applications,” Neural Processing Letters, vol. 41, no. 1, pp. 89–106, 2015.
[18] L. Yang and H. Dong, “Robust support vector machine with generalized quantile loss for classification and regression,” Applied Soft Computing, vol. 81, p. 105483, 2019.
[19] L. Yang, G. Ding, C. Yuan, and M. Zhang, “Robust regression framework with asymmetrical analogous to correntropy-induced loss,” Knowledge-Based Systems, vol. 191, p. 105211, 2020.
[20] X. Huang, L. Shi, and J. A. Suykens, “Solution path for pin-svm classifiers with positive and negative \(\tau\) values,” IEEE transactions on neural networks and learning systems, vol. 28, no. 7, pp. 1584–1593, 2017.
[21] M. O.L., Nonlinear Programming. SIAM, 1994.
[22] S. A. Schikopf B, “Learning with kernels: support vector machines, regularization, optimization, and beyond.” MIT press, 2002.
[23] C. Blake and C. Merz, “Uci repository of machine learning databases. Irvine, university of california, 1998.”
[24] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” IEEE transactions on Neural Networks, vol. 13, no. 2, pp. 415–425, 2002.
[25] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern classification. John Wiley & Sons, 2012.