Chapter 4

Res-TSVR: Robust Twin Support Vector Regression based on Rescaled Hinge Loss Function

In this chapter, robust twin support vector regression (TSVR) is proposed using rescaled hinge loss function. A rigorous analysis is provided along with several experimental results to show the efficacy of the model.

4.1 Introduction

In this chapter, a regression variant of SVM is discussed. In 2007, Khemchandani and Chandra [32] came up with an idea of splitting the SVM formulation into two simpler formulations that generate two non-parallel hyperplanes. This model was called Twin SVM (TWSVM). This variant of SVM was analytically and experimentally proved to be four times faster than SVM [32]. Peng [86] also tried to enhance its properties by adding sparseness to it. In 2012, Peng and Xu [87] proposed to generate two hyperspheres, each of them being smaller than the standard SVM hyperplane. As compared with TWSVM, that work avoided matrix inversions in its two dual quadratic formulations [87]. Such

non-parallel hyperplanes were also generated using the clustering mechanism. In that method, the similarity between the training points was considered to form clusters [88]. That technique was also different from the TWSVM in a manner that it solved a single quadratic problem in contrast to TWSVM, which solves two quadratic problems. As this variant also used hinge loss, the problem of sensitivity towards noise was still there. Similarly, various approaches were proposed to make it robust, e.g., the weighted version of TWSVM [89], TWSVM with ramp loss function [90], TWSVM with pinball loss function [65], TWSVM with rescaled hinge loss [91] and TWSVM with general pinball loss [92], etc. Recent research in the field of TWSVM for classification includes the use of stochastic gradient descent for TWSVM, which is useful to solve large-scale problems [93]. Using the concept of TWSVM in neural networks, the problem of imbalance in the data sets was solved [94].

The above-discussed formulations were proposed for the classification problems. Researchers also proposed similar formulations for regression. Like the TWSVM, twin support vector *regression* (TSVR), a variant of SVR, was also proposed in [33]. After its proposal, it was observed that the model suffers from the problem of overfitting. To solve this problem, a weighted version of TSVR was proposed [95]. Similarly, the least-square variant of TSVR was also proposed [96]. It was observed that although the TSVR is four times faster than the SVR method [96], it suffers from the following limitations (see [96]):

(i) It is highly sensitive to noise and outliers.

(ii) It has the problem of overfitting.

(iii) It is not sparse.

Because of these facts, Chen et al. [97] proposed an improved TSVR, which was robust to noise and had sparseness. The authors used the L_1 norm instead of L_2 norm to make the model robust. Also, to avoid the problem of overfitting, they added a regularization

term in the objective function. Furthermore, their proposed approach was sparse, which was missing in TSVR. The critical point of their work was that they derived the linear programming (LP) model, which was comparatively easier to solve than TSVR. They used the Newton-Armjio method to solve these LP problems [97].

Other approaches that are robust towards noise include the one proposed by [98], in which a weighted ε-TSVR (W-ETSVR) was introduced using a quadratic loss function. The experimental results [98] demonstrated that their algorithm could reduce the impact of outliers to a certain extent. That work was inspired by the work of [99] in which an unconstrained Lagrangian TSVR was proposed so that the computational speed can be further improved. Furthermore, the pinball loss function was also used for the regression problems (pin-TSVR) to make the TSVR robust to outliers [100]. That work is amongst the most recent contribution in making the TSVR robust. Anagha et al. [100] used the squared pinball loss function and solved the primal problem itself. The pinball loss function makes the problem strongly convex, which can be solved through an iterative algorithm [100].

4.1.1 Motivation Behind This Work

Several machine learning researchers have focused their attention on imparting robustness to the models. Since machine learning is essentially a data-driven approach, the resulting model should be robust against noisy data. Specifically, concerning SVMs, it is observed that many robust variants [25, 36, 37] of SVM are proposed to make the model robust to noise and outliers. A recently proposed approach [78] of using correntropy with hinge loss, i.e., rescaled hinge loss, has made the SVM model robust for a classification problem. However, the question of using it for regression problems not been addressed.

The contribution of correntropy in making the loss functions of a classification problem robust motivated us to try it for regression problems. The correntropy-based loss

functions are not only exploited in machine learning but also in deep learning [101]. Furthermore, the hinge loss is an unbounded and non-smooth function. However, it was observed that the composition of the correntropy-based loss function (C-loss [81]) with hinge loss makes the overall function bounded which is desirable to deal with the outliers. This function is monotonic, smooth, and non-convex [78]. Inspired by these properties and the results obtained over the classification tasks, its application is extended to regression, using the TSVR model.

4.1.2 Contribution of This Work

Inspired by the above-discussed research and the identified gaps, the rescaled hinge loss function was proposed for the regression problem of TSVR. The following contributions had added novelty and quality to work.

(i) In this work, rescaled hinge loss was proposed for TSVR, which had not been proposed earlier.

(ii) The performance metrics were compared before and after adding noise to the data sets. Synthetic data sets were used in some experiments. This facilitates a clear visual understanding of changes in the performance of various approaches.

(iii) In this work, the dual of the resultant non-convex problem was formulated rigorously after incorporating rescaled hinge loss into the conventional TSVR. Through this approach, the robustness of the standard TSVR was improved. It also reduced the problem of overfitting.

4.1.3 Outline

This chapter is organized as follows. In Section 4.2, a brief introduction is proposed to the approaches with which our method is being compared. Section 4.3 presents the new approach in detail, followed by the analysis of Res-TSVR in Section 4.4. In Section 4.5,

numerical experiments are performed on the proposed approach Res-TSVR. Section 4.6 discusses the effect of rescaling parameter, $\hat{\eta}$ on the performance of Res-TSVR. Section 4.7 concludes this work by highlighting the main contributions and advantages of the proposed work.

4.2 Brief Introduction to ε-SVR and TSVR

In what follows, a brief introduction of the methods which are closely related to our work is provided.

4.2.1 Support Vector Regression

SVR [10] follows the same principle as SVM does with some differences. In this case, a level of tolerance, ε is predefined as this model uses ε-loss function, which is in Figure 4.1. In Figure 4.1, $-\varepsilon$ and $+\varepsilon$ define the error bounds. SVR finds a regression function

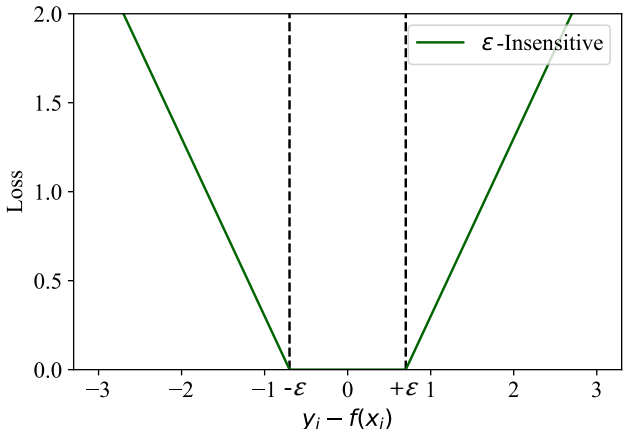


Figure 4.1: ε-Insensitive Loss Function

at the end which is defined as

$$f(x) = w^T x + b, \tag{4.1}$$

where $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$. Through this regression function, the model predicts the target values corresponding to each instance of the test set. Using the regularization

term $\frac{1}{2}\|w\|^2$ and the slack variables $\xi = (\xi_1, \xi_2, \cdots, \xi_m)^T$ and $\hat{\xi} = (\hat{\xi}_1, \hat{\xi}_2, \cdots, \hat{\xi}_m)^T$ corresponding to upper and lower bound functions, the primal form of SVR is defined as follows:

$$\begin{aligned} \min_{w,b,\xi,\hat{\xi}} \quad & \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \hat{\xi}_i) \\ \text{subject to} \quad & y_i - (X_i w + b) \leq \varepsilon + \xi_i,\ \xi_i \geq 0, i = 1, 2, \cdots, m, \\ & (X_i w + b) - y_i \leq \varepsilon + \hat{\xi}_i,\ \hat{\xi}_i \geq 0, i = 1, 2, \cdots, m, \end{aligned} \tag{4.2}$$

where $C \geq 0$ is a regularization parameter that signifies the trade-off between the loss function and the regularization term. Note that the objective function in (4.2) corresponds to the linear SVR. It can be adapted easily for a non-linear kernel. Although SVR has been proved to be an efficient regressor compared to many other regressors, the main challenge for using this SVR is the high computational complexity, which raises to $O(m^3)$.

4.2.2 Twin Support Vector Regression

A limitation of ε-SVR is that it takes high computational time to make the predictions [33]. This limitation is handled by TSVR, in which the complex quadratic function of SVR is divided into two simpler quadratic equations. As the name indicates, TSVR provides two hyperplanes which are given by

$$f_1(x) = w_1^T x + b_1 \quad \text{and} \quad f_2(x) = w_2^T x + b_2, \tag{4.3}$$

where $w_1, w_2 \in \mathbb{R}^n$ and $b_1, b_2 \in \mathbb{R}$, and these are obtained by solving the following optimization problems (see [33]):

$$\min_{w_1,b_1,\xi} \quad \tfrac{1}{2}\|Y - \varepsilon_1 e - (Xw_1 + b_1 e)\|^2 + C_1 e^T \xi$$

$$\text{subject to} \quad Y - (Xw_1 + b_1 e) \geq \varepsilon_1 e - \xi, \quad \xi \geq 0, \tag{4.4}$$

and

$$\begin{aligned} \min_{w_2, b_2, \eta} \quad & \tfrac{1}{2} \|Y + \varepsilon_2 e - (Xw_2 + b_2 e)\|^2 + C_2 e^T \eta \\ \text{subject to} \quad & (Xw_2 + b_2 e) - Y \geq \varepsilon_2 e - \eta, \quad \eta \geq 0, \end{aligned} \tag{4.5}$$

where C_1, C_2, ε_1 and ε_2 are the positive parameters, and e is the m-tuple column vector $(1, 1, \cdots, 1)^T$. The two regressors in (4.3) find the ε-insensitive lower and upper bound regressors, respectively. The final regressor is computed by the mean of these two regressors. It can be observed that the first terms in (4.4) and (4.5) represent the sum of squared distances from $y = w_1^T x + b_1 + \varepsilon_1$ or $y = w_2^T x + b_2 - \varepsilon_2$ to the training points [33]. The final loss function (which is the combination of ε-insensitive loss function and squared error loss function) is shown in Figure 4.2 (represented by 'Total Error').

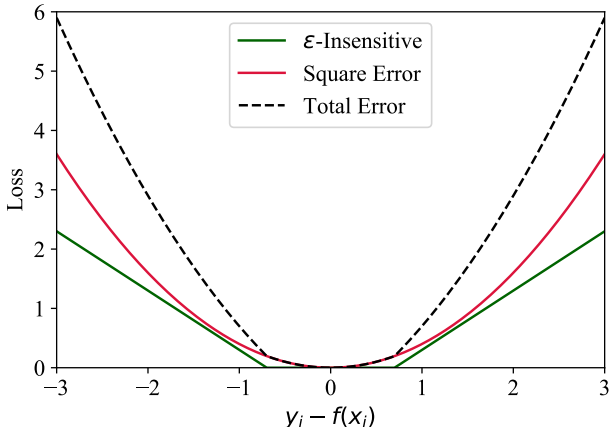


Figure 4.2: Loss Functions used in TSVR

To solve (4.4) and (4.5), their dual formulations can be derived as follows (see [33]):

$$\max_{\alpha \in \mathbb{R}^m} \quad -\tfrac{1}{2} \alpha^T G (G^T G)^{-1} G^T \alpha + f^T G (G^T G)^{-1} \alpha - f^T \alpha$$

$$\text{subject to} \quad 0 \leq \alpha \leq C_1 e, \tag{4.6}$$

where $G = [X\ e]$ and $f = Y - \varepsilon_1 e$. Similarly, for the second TSVR, the dual form is

$$\begin{aligned} \max_{\gamma \in \mathbb{R}^m} \quad & -\tfrac{1}{2}\gamma^T G(G^T G)^{-1} G^T \gamma - h^T G(G^T G)^{-1}\gamma + h^T \gamma \\ \text{subject to} \quad & 0 \leq \gamma \leq C_2 e, \end{aligned} \tag{4.7}$$

where $h = Y + \varepsilon_2 e$. From the solutions of (4.6) and (4.7), we can obtain the augmented vectors $\mu_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix}$ and $\mu_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix}$ by computing

$$\mu_1 = (G^T G + \sigma I)^{-1} G^T (f - \alpha) \quad \text{and} \quad \mu_2 = (G^T G + \sigma I)^{-1} G^T (h + \gamma), \text{ respectively,} \tag{4.8}$$

where $\sigma > 0$ is a constant so that $G^T G + \sigma I$ is invertible. After obtaining w_1, w_2 and b_1, b_2, the final regressor can be expressed as follows

$$f(x) = \tfrac{1}{2}(f_1(x) + f_2(x)) = \tfrac{1}{2}(w_1 + w_2)^T x + \tfrac{1}{2}(b_1 + b_2). \tag{4.9}$$

In (4.4), if the loss function is replaced by the pinball loss function, one can obtain the pin-TSVR [100]. If the weight vector is assigned to every training data point, which is multiplied by the hinge loss function, W-ETSVR can be obtained [98].

4.3 Res-TSVR

In this section, the formulation of the proposed approach, Res-TSVR, is described. This technique extended the use of rescaled hinge loss from classification to a regression problem. This method aimed to formulate a robust TSVR, which is less sensitive to noise than the existing approaches.

Like TSVR, the proposed regressor, TSVR, using rescaled hinge loss (Res-TSVR),

generates a pair of non-parallel lines for the training samples. In TSVR, the use of a square loss function with a hinge loss function is already explained in Subsection 4.2.1. In the proposed scheme, the hinge loss function is replaced with the rescaled hinge loss function, and a regularizer term is added to the TSVR problem. This regularizer term is added to address the overfitting problem with the conventional TSVR problem [97].

4.3.1 Problem Formulation

The optimization problem of the conventional TSVR (4.4) can be rewritten as

$$\min_{w_1,b_1} \quad \tfrac{1}{2}\left\|Y-(X_iw_1+b_1e)\right\|^2 + C_1\sum_{i=1}^{m} P_{\text{Hinge}}(y_i-(X_iw_1+b_1)), \tag{4.10}$$

for TSVR1 and

$$\min_{w_2,b_2} \quad \tfrac{1}{2}\left\|(Xw_2+b_2e)-Y\right\|^2 + C_2\sum_{i=1}^{m} P_{\text{Hinge}}((X_iw_2+b_2)-y_i) \tag{4.11}$$

for TSVR2, where $P_{\text{Hinge}}(\cdot)$ represents the penalty function. From Figure 4.3, it can be easily observed that the hinge loss function is not bounded, and the value of this loss function is very substantial in the case of outliers. In [68], it is shown that the use of correntropy with hinge loss function (rescaled hinge loss function) can make the revised loss function bounded, monotonic, and non-convex. The rescaled hinge loss function is given by

$$P_{\text{rhinge}}(\xi) = \beta\left[1-\exp(-\hat{\eta}P_{\text{Hinge}}(\xi))\right], \tag{4.12}$$

where $\beta = \frac{1}{1-\exp(-\hat{\eta})}$ is a normalization parameter and $\hat{\eta} > 0$ is a constant.

The rescaled hinge loss function is shown in Figure 4.3 with different values of $\hat{\eta}$. This figure indicates that the rescaling has changed the properties of hinge loss function from unbounded to bounded and convex to non-convex. One can observe the behavior of the graph with the variation in the degree of rescaling. Also, it is

noteworthy that as $\hat{\eta} \to 0$, the rescaled hinge loss behaves like the conventional hinge loss, i.e., $P_{\text{rhinge}}(\xi) \to P_{\text{Hinge}}(\xi)$.

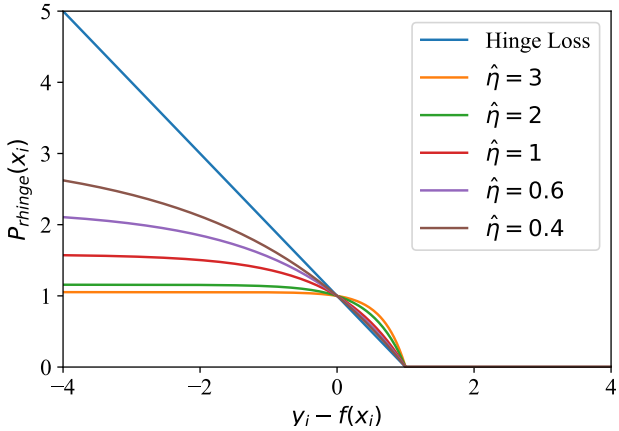


Figure 4.3: Hinge Loss and Rescaled Hinge Loss Function with Different $\hat{\eta}$ Values

The hinge loss function in (4.10) and (4.11) is replaced by the rescaled hinge loss function to make the objective function robust to noise. Introducing rescaled hinge loss in (4.10) and (4.11) leads to

$$\min_{w_1,b_1} \quad \tfrac{1}{2}\left\|Y - (Xw_1 + b_1 e)\right\|^2 + \frac{1}{2}\left\|\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}\right\|^2 + C_1 \sum_{i=1}^{m} P_{\text{rhinge}}(y_i - (X_i w_1 + b_1)) \quad (4.13)$$

for TSVR1 and

$$\min_{w_2,b_2} \quad \tfrac{1}{2}\left\|(Xw_2 + b_2 e) - Y\right\|^2 + \frac{1}{2}\left\|\begin{bmatrix} w_2 \\ b_2 \end{bmatrix}\right\|^2 + C_2 \sum_{i=1}^{m} P_{\text{rhinge}}((X_i w_2 + b_2) - y_i) \quad (4.14)$$

for TSVR2. It is to be noted that a regularizer term added in the above equations prevented the model from overfitting. Next, this problem was converted to its dual form to implement it efficiently. The conversion of the primal form of TSVR1 to its dual form is shown in Section 4.3.2. The same steps were followed for TSVR2.

By conjugate function theory,

$$P_{\text{rhinge}}(z) = \beta(1 - \exp(-\hat{\eta}P_{\text{Hinge}}(z))) = \inf_{t<0} \quad \beta(1 + \psi(t) - \hat{\eta}P_{\text{Hinge}}(z)t), \tag{4.15}$$

where $\psi(t) = t(1 - \log(-t)), \quad t < 0.$

Thus, TSVR1 was recasted as

$$\min_{w_1,b_1,t} \quad \tfrac{1}{2}\left\|Y - (Xw_1 + b_1e)\right\|^2 + \frac{1}{2}\left\|\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}\right\|^2 - \frac{C_1\beta\hat{\eta}}{2}\sum_{i=1}^{m} P_{\text{Hinge}}(y_i - (X_iw_1 + b_1))t_i, \tag{4.16}$$

where $t = (t_1, t_2, \cdots, t_m)$. Defining $p_i = C_1\beta\hat{\eta}(-t_i) > 0$, (6.17) can be written as

$$\min_{w_1,b_1} \quad \tfrac{1}{2}\left\|Y - (Xw_1 + b_1e)\right\|^2 + \frac{1}{2}\left\|\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}\right\|^2 + \sum_{i=1}^{m} p_i\xi_i, \tag{4.17}$$

where $\xi_i = P_{\text{Hinge}}(y_i - (X_iw_1 + b_1))$. So, the final primal form (4.13) can be rewritten as

$$\begin{aligned} \min_{w_1,b_1} \quad & \tfrac{1}{2}\left\|Y - (Xw_1 + b_1e)\right\|^2 + \frac{1}{2}\left\|\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}\right\|^2 + \sum_{i=1}^{m} p_i\xi_i \\ \text{subject to} \quad & Y - (Xw_1 + b_1e) \geq \varepsilon_1 e - \xi_i, \quad \xi_i \geq 0, \ i = 1, 2, \cdots, m. \end{aligned} \tag{4.18}$$

- *Explanation of the Objective Function of (4.18)*

In (4.18), the first term represents the *squared error loss function.* It tries to minimize the squared distance from the function $Y = Xw_1 + b_1 - \varepsilon_1$. This term fits the function to the ε-insensitive upbound regressor. The same is the case with TSVR2, whose first term fits the function to the ε-insensitive lower bound regressor.

The second term of the objective function in (4.18) represents the *regularization term.* The minimization of this term attempts to avoid overfitting the training samples.

The third term of the objective is the *rescaled hinge loss function*, which keeps the loss bounded and makes the overall objective function robust. However, it makes the corresponding optimization problem non-convex. It was therefore decided to convert the optimization problem to its dual form which is convex and can be solved efficiently.

4.3.2 Linear Res-TSVR

In this subsection, the dual of (4.18) is formulated. To derive the dual form of (4.18), first the Lagrangian for (4.18) was introduced as follows:

$$L(w_1, b_1, \xi, \alpha, \mu) = \tfrac{1}{2}\left\|Y - (Xw_1 + b_1 e)\right\|^2 + \tfrac{1}{2}\left\|\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}\right\|^2 + p^T\xi - \alpha^T((Y - (Xw_1 + b_1 e)) + \xi) - \mu^T\xi \quad (4.19)$$

where α and μ are Lagrange multiplier vectors.
The Karush-Kuhn-Tucker (KKT) necessary optimality conditions [102] for (4.19) are

$$\frac{\partial L}{\partial w_1} = -X^T(Y - (Xw_1 + b_1 e)) + w_1 + X^T\alpha = 0, \quad (4.20)$$

$$\frac{\partial L}{\partial b_1} = -e^T(Y - (Xw_1 + b_1 e)) + b_1 + e^T\alpha = 0, \quad (4.21)$$

$$\frac{\partial L}{\partial \xi} = p - \alpha - \mu = 0, \quad (4.22)$$

$$Y - (Xw_1 + b_1 e) \geq -\varepsilon_1 e - \xi, \quad \xi \geq 0, \quad (4.23)$$

$$\alpha^T(Y - (Xw_1 + b_1 e) + \xi) = 0, \quad \alpha \geq 0, \quad \mu \geq 0. \quad (4.24)$$

The dual objective function for (4.18) is

$$D(\alpha, \mu) = \inf\{L(w_1, b_1, \xi, \alpha, \mu) \ : \ Y - (Xw_1 + b_1 e) \geq -\varepsilon_1 e - \xi\}. \quad (4.25)$$

From (4.20) and (4.21), we get

$$-\begin{bmatrix} X^T \\ e^T \end{bmatrix}\left(Y - [X \quad e]\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}\right) + \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} X^T \\ e^T \end{bmatrix}\alpha = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{4.26}$$

This can be written as

$$-G^T(Y - Gu_1) + u_1 + G^T\alpha = 0, \tag{4.27}$$

where $G = [X \quad e]$ and $u_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix}$. Equation (4.27) implies

$$u_1 = (G^TG + I)^{-1}G^T(Y - \alpha). \tag{4.28}$$

Therefore,

$$\begin{aligned} D(\alpha, \mu) &= \tfrac{1}{2}(Y - Gu_1)^T(Y - Gu_1) + \tfrac{1}{2}u_1^Tu_1 + (p - \mu)^T\xi \\ &= \tfrac{1}{2}(Y - Gu_1)^T(Y - Gu_1) + \tfrac{1}{2}u_1^Tu_1 + \alpha^T\xi, \quad \text{by (4.22)} \\ &= \tfrac{1}{2}(Y - Gu_1)^T(Y - Gu_1) + \tfrac{1}{2}u_1^Tu_1 - \alpha^T(Y - Gu_1 + \varepsilon_1 e)\xi \\ &= \tfrac{1}{2}(Y - Gu_1)^T(Y - Gu_1) + \tfrac{1}{2}u_1^Tu_1 + \alpha^TGu_1 - (\varepsilon_1 e^T + Y^T)\alpha. \end{aligned} \tag{4.29}$$

From (4.27), we further get $(u_1 + G^T\alpha) = G^T(Y - Gu_1)$. Thus,

$$\begin{aligned} (u_1 + G^T\alpha)^Tu_1 &= (Y - Gu_1)^TGu_1 \\ &= \tfrac{1}{2}(Y - Gu_1)^T(Y - Gu_1) - \tfrac{1}{2}u_1^Tu_1 + (u_1 + G^T\alpha)^Tu_1 - (\varepsilon_1 e^T + Y^T)\alpha \\ &= \tfrac{1}{2}(Y - Gu_1)^T(Y - Gu_1) - \tfrac{1}{2}(Y - Gu_1)^TGu_1 - \tfrac{1}{2}u_1^Tu_1 \\ &\quad + C_1(Y - Gu_1)^TGu_1 - (\varepsilon_1 e^T + Y^T)\alpha \\ &= \tfrac{1}{2}(Y - Gu_1)^T(Y + Gu_1) - \tfrac{1}{2}u_1^Tu_1 - (\varepsilon_1 e^T + Y^T)\alpha \end{aligned}$$

$$= \tfrac{1}{2}(Y - Gu_1)^T Y + \tfrac{1}{2}(Y - Gu_1)^T Gu_1 - \tfrac{1}{2}u_1^T u_1 - (\varepsilon_1 e^T + Y^T)\alpha. \tag{4.30}$$

Since, from (4.27), after taking transpose, we obtained

$$(Y - Gu_1)^T G = \alpha^T G + u_1^T, \tag{4.31}$$

and the expression in (4.30) leads to

$$\begin{aligned}
&\tfrac{1}{2}(Y - Gu_1)^T Y + \tfrac{1}{2}(\alpha^T G + u_1^T)u_1 - \tfrac{1}{2}u_1^T u_1 - (\varepsilon_1 e^T + Y^T)\alpha \\
&= \tfrac{1}{2}(Y - Gu_1)^T Y + \tfrac{1}{2}\alpha^T Gu_1 - (\varepsilon_1 e^T + Y^T)\alpha \\
&= \tfrac{1}{2}Y^T Y - \tfrac{1}{2}((Gu_1)^T Y - (Gu_1)^T \alpha) - (\varepsilon_1 e^T + Y^T)\alpha \\
&= \tfrac{1}{2}Y^T Y - \tfrac{1}{2}(Gu_1)^T (Y - \alpha) - (\varepsilon_1 e^T + Y^T)\alpha \\
&= \tfrac{1}{2}Y^T Y - \tfrac{1}{2}u_1^T G^T (Y - \alpha) - (\varepsilon_1 e^T + Y^T)\alpha.
\end{aligned} \tag{4.32}$$

From (4.27), again we have $G^T(Y - \alpha) = G^T Gu_1 + u_1$. Thus, (4.32) is

$$\tfrac{1}{2}Y^T Y - \tfrac{1}{2}u_1^T (G^T Gu_1 + u_1) - (\varepsilon_1 e^T + Y^T)\alpha. \tag{4.33}$$

As the first term in (4.33) is constant with respect to α and μ, the dual function can be taken as

$$\begin{aligned}
D(\alpha, \mu) &= -\tfrac{1}{2}u_1^T G^T Gu_1 - \tfrac{1}{2}u_1^T u_1 - (\varepsilon_1 e^T + Y^T)\alpha \\
&= -\tfrac{1}{2}(Gu_1)^T (Gu_1) - \tfrac{1}{2}u_1^T u_1 - (\varepsilon_1 e^T + Y^T)\alpha \\
&= -\tfrac{1}{2}\Big(G(G^T G + I)^{-1} G^T (Y - \alpha)\Big)^T \Big(G(G^T G + I)^{-1} G^T (Y - \alpha)\Big) \\
&\quad - \tfrac{1}{2}\Big((G^T G + I)^{-1} G^T (C_1 Y - \alpha)\Big)^T \Big((G^T G + I)^{-1} G^T (Y - \alpha)\Big) - (\varepsilon_1 e^T + Y^T)\alpha.
\end{aligned} \tag{4.34}$$

On simplification, (4.33) leads to the dual formulation of (4.18) as below:

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2}\alpha^T G(G^T G + I)^{-1} G^T \alpha + Y^T G(G^T G + I)^{-1} G^T \alpha - (\varepsilon_1 e^T + Y^T)\alpha \\ \text{subject to} \quad & 0 \leq \alpha_i \leq -C_1 \beta \hat{\eta} t_i, \ i = 1, 2, \cdots, m. \end{aligned} \tag{4.35}$$

Similarly, the dual formulation for Res-TSVR2 is given by

$$\begin{aligned} \max_{\gamma} \quad & -\frac{1}{2}\gamma^T G(G^T G + I)^{-1} G^T \gamma - Y^T G(G^T G + I)^{-1} G^T \gamma (Y^T - \varepsilon_2 e^T)\gamma \\ \text{subject to} \quad & 0 \leq \gamma_i \leq -C_2 \beta \hat{\eta} t_i, \ i = 1, 2, \cdots, m. \end{aligned} \tag{4.36}$$

It should be noted for Res-TSVR2 that

$$u_2 = (G^T G + I)^{-1} G^T (Y + \gamma). \tag{4.37}$$

4.3.3 Non-Linear Res-TSVR

In this subsection, the formulation for non-linear Res-TSVR is proposed. After careful selection of the kernel, $K(X, X^T)$, the primal forms of the TSVR optimization problem became

$$\begin{aligned} \min_{w_1, b_1, \xi} \quad & \tfrac{1}{2} \left\| Y - (K(X, X^T)w_1 + b_1 e) \right\|^2 + \frac{1}{2} \left\| \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} \right\|^2 + \sum_{i=1}^{m} p_i \xi_i \\ \text{subject to} \quad & Y - (K(X, X^T)w_1 + b_1 e) \geq \varepsilon_1 e - \xi, \quad \xi \geq 0. \end{aligned} \tag{4.38}$$

For TSVR2, the primal form is

$$\min_{w_2, b_2, \eta} \quad \tfrac{1}{2} \left\| Y - (K(X, X^T)w_2 + b_2 e) \right\|^2 + \frac{1}{2} \left\| \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} \right\|^2 + \sum_{i=1}^{m} g_i \eta_i$$

$$\text{subject to } \; (K(X, X^T)w_2 + b_2 e) - Y \geq \varepsilon_2 e - \eta, \quad \eta \geq 0, \tag{4.39}$$

where $p_i = C_1\beta\hat{\eta}(-t_i) > 0$ for TSVR1 and $g_i = C_2\beta\hat{\eta}(-t_i) > 0$ for TSVR2.

Following the same steps as described earlier for the linear case, the dual formulations for the non-linear TSVRs are given by

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2}\alpha^T H(H^T H + I)^{-1}H^T\alpha + Y^T H(H^T H + I)^{-1}H^T\alpha - (\varepsilon_1 e^T + Y^T)\alpha \\ \text{subject to} \quad & 0 \leq \alpha_i \leq -C_1\beta\hat{\eta}t_i, \; i = 1, 2, \cdots, m \end{aligned} \tag{4.40}$$

and

$$\begin{aligned} \max_{\gamma} \quad & -\frac{1}{2}\gamma^T H(H^T H + I)^{-1}H^T\gamma - Y^T H(H^T H + I)^{-1}H^T\gamma + (Y^T - \varepsilon_1 e^T)\gamma \\ \text{subject to} \quad & 0 \leq \gamma_i \leq -C_2\beta\hat{\eta}t_i, \; i = 1, 2, \ldots, m. \end{aligned} \tag{4.41}$$

where $H = [K(X, X^T) \;\; e]$. For computing the vectors of w_1, b_1 and w_2, b_2,

$$u_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = (H^T H + I)^{-1}H^T(Y - \alpha) \;\; \text{for TSVR1} \tag{4.42}$$

and

$$u_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (H^T H + I)^{-1}H^T(Y + \gamma) \;\; \text{for TSVR2.} \tag{4.43}$$

are computed.

4.3.4 Algorithm Res-TSVR

The complete training procedure is described in this subsection. Algorithm 2 gives the pseudo code for Res-TSVR. The weight vectors and the bias term of the two hyperplanes were obtained by Algorithm 2. The final prediction function for linear Res-TSVR is

Algorithm 2 Training Procedure of Res-TSVR

Input: Training set $\{X_i, y_i\}_{i=1}^m$;

$\hat{\eta}$: a rescaled hinge loss constant;

C_1, C_2: the regularization parameters;

$k_{\max}$: number of iterations;

$\varepsilon_1, \varepsilon_2$: error bounds;

tol: the tolerance value;

Output: $\hat{w_1}, \hat{b_1}$ and $\hat{w_2}, \hat{b_2}$, the solutions of Res-TSVR1 (4.18) and Res-TSVR2.

1: Initialization : $k = 1$, $t_1 = [-1, -1, \ldots, -1]^T \in \mathbb{R}^m$, $t_2 = [-1, -1, \ldots, -1]^T \in \mathbb{R}^m$, $G = [X \quad e]$.

2: Compute $\beta = \frac{1}{1-\exp(-\hat{\eta})}$.

3: Compute α and γ using (4.35) and (4.36), respectively for Linear Res-TSVR or (4.40) and (4.41) for Non-Linear Res-TSVR.

4: **while** $k < k_{\max}$ **do**

5: $\alpha_{\text{old}} = \alpha$ and $\gamma_{\text{old}} = \gamma$.

6: Compute w_1, b_1 and w_2, b_2 using (4.28) and (4.37), respectively (or (4.42) and (4.43), if considering Non-Linear Res-TSVR).

7: Compute $z_{1i} = y_i - (X_i w_1 + b_1 e)$ and $z_{2i} = y_i - (X_i w_2 + b_2 e), \quad i = 1, 2, \cdots, m$.

8: Compute t_1 and t_2 by $t_1 = -\exp\{-\hat{\eta} P_{\text{Hinge}}(z_{1i})\}$, $t_2 = -\exp\{-\hat{\eta} P_{\text{Hinge}}(z_{2i})\}$, $i = 1, 2 \ldots, m$.

9: Compute α and γ as described in Step 2.

10: $k = k + 1$.

11: **end while**

12: Return $\hat{w_1} = w_1$, $\hat{b_1} = b_1$ and $\hat{w_2} = w_2$, $\hat{b_2} = b_2$.

given by

$$f(x) = \tfrac{1}{2}(f_1(x) + f_2(x)) = \frac{1}{2}\left[\left(\hat{w_1}^T x + \hat{b_1}\right) + \left(\hat{w_2}^T x + \hat{b_2}\right)\right]. \tag{4.44}$$

For non-linear Res-TSVR, the final prediction is:

$$f(x) = \tfrac{1}{2}(f_1(x) + f_2(x)) = \frac{1}{2}\Big[\left((K(x, x^T)\hat{w_1} + \hat{b_1}\right) + \left((K(x, x^T)\hat{w_2} + \hat{b_2})\right)\Big]. \tag{4.45}$$

Note 1 In Algorithm 2, if an appropriate guess of $k_{\max}$ is difficult, one can use the stopping criteria as $\|\alpha_{\text{old}} - \alpha\| > \text{tol}$ and $\|\gamma_{\text{old}} - \gamma\| > \text{tol}$ instead of $k < k_{\max}$. The

stopping criteria $\|\alpha_{\text{old}} - \alpha\| > \text{tol}$ and $\|\gamma_{\text{old}} - \gamma\| > \text{tol}$ along with $k < k_{\max}$ can be used to avoid unnecessary computations if there is no significant improvement of γ and α values in consecutive iterations. In Subsection 4.4.2, an estimate on the number of iterations is given to obtain an ε-precision optimum solution to (4.18).

4.4 Analysis of Res-TSVR

Optimization Process for Res-TSVR

For each iteration, i.e., for a value of k, in Res-TSVR, it is observed that the updation of t_1, t_2 as well as α and γ is required. As the conjugate function theory (see equations (4.15) and (4.16)) applied to the formulation of TSVR1 (4.13) and TSVR2 (4.14), *Alternating Optimization Technique* [103] was applied on the Step 6 of Res-TSVR. The main advantages of using this technique are that

(i) for the proposed TSVR1 and TSVR2, the technique can never diverge (see Subsection 4.4.1),

(ii) the maximum number of steps required to reach at an ε-precision solution can be estimated (see Subsection 4.4.2), and

(iii) this technique makes a parallel computation possible because one dimensional searches across different dimensions are independent of each other.

However, the proposed method have user-dependent regularizers C_1 and C_2. An incorrect selection of these regularizers may lead to poorer accuracy on the test set.

At the k-th iteration, for a given t_1^k and t_2^k, (4.35) and (4.36) are solved to obtain α^k and γ^k for linear Res-TSVR and similarly (4.40) and (4.41) for non-linear Res-TSVR. Consequently, u_1^k and u_2^k are obtained to calculate $\begin{bmatrix} w_1^k \\ b_1^k \end{bmatrix}$ and $\begin{bmatrix} w_2^k \\ b_2^k \end{bmatrix}$, respectively.

Using the obtained α^k and γ^k, t_1 and t_2 are updated by solving (see (4.15))

$$\max_{t_1^k<0} \sum_{i=1}^{m} \left\{ \hat{\eta} P_{\text{Hinge}}(z_{1i}^k) t_{1i}^k - g(t_{1i}^k) \right\} \tag{4.46}$$

and

$$\max_{t_2^k<0} \sum_{i=1}^{m} \left\{ \hat{\eta} P_{\text{Hinge}}(z_{2i}^k) t_{2i}^k - g(t_{2i}^k) \right\}, \tag{4.47}$$

respectively, where $z_{1i}^k = y_i - (X_i w_1^k + b_1^k e)$ and $z_{2i}^k = y_i - (X_i w_2^k + b_2^k e), \quad i = 1, 2, \cdots, m.$

The analytic solutions of (4.46) and (4.47) give

$$t_{1i}^{k+1} = -\exp(-\hat{\eta} P_{\text{Hinge}}(z_{1i}^k)) \tag{4.48}$$

$$\text{and } t_{2i}^{k+1} = -\exp(-\hat{\eta} P_{\text{Hinge}}(z_{2i}^k)). \tag{4.49}$$

Updating the iteration number from k to $k+1$, again the calculations of u_1 and u_2 are followed to calculate $\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}$ and $\begin{bmatrix} w_2 \\ b_2 \end{bmatrix}$, respectively, for the $(k+1)$-th iteration.

4.4.1 Convergence Proof of Algorithm 2

In this subsection, the convergence proof of Algorithm 2 is given for the problem (4.18). Let the objective function of (4.18) be $P_1(w_1, b_1, t)$, i.e.,

$$P_1(w_1, b_1, t) = \tfrac{1}{2} \|Y - (Xw_1 + b_1 e)\|^2 + \tfrac{1}{2} \left\| \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} \right\|^2 - \tfrac{C_1 \beta}{2} \sum_{i=1}^{m} (\psi(t_i) - \hat{\eta} P_{\text{Hinge}}(z_i) t_i).$$

Suppose

$$Q_1(w_1, b_1) = \inf_{t<0} P_1(w_1, b_1, t).$$

Then, by conjugate function theory,

$$\begin{aligned} Q_1(w_1, b_1) &= \tfrac{1}{2}\left\|Y - (Xw_1 + b_1 e)\right\|^2 + \tfrac{1}{2}\left\|\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}\right\|^2 - \frac{C_1\beta}{2}\sum_{i=1}^{m}\exp\left(-\hat{\eta}P_{\text{Hinge}}(z_i)\right) \\ &\geq -\frac{C_1\beta}{2}\sum_{i=1}^{m}\exp\left(-\hat{\eta}P_{\text{Hinge}}(z_i)\right) \\ &\geq -\frac{C_1\beta m}{2} \end{aligned}$$

since

$$\frac{1}{m}\sum_{i=1}^{m}\exp\left(-\hat{\eta}P_{\text{Hinge}}(z_i)\right) \leq \frac{1}{m}\sum_{i=1}^{m}1 = 1.$$

Hence,

$$P_1(w_1, b_1, t) \geq Q_1(w_1, b_1) \geq -\frac{C_1\beta\hat{\eta}}{2}.$$

Thus, the sequence $\{P_1(w_1^k, b_1^k, t^k)\}$ is bounded below. Also, for a given (w_1^k, b_1^k),

$$P_1(w_1^k, b_1^k, t^k) \leq P_1(w_1^k, b_1^k, t^{k-1}).$$

Hence,

$$P_1(w_1^k, b_1^k, t^k) \geq P_1(w_1^{k+1}, b_1^{k+1}, t^k) \geq P_1(w_1^{k+1}, b_1^{k+1}, t^{k+1}).$$

Therefore, the sequence $\{P_1(w_1^k, b_1^k, t^k)\}$ is convergent and it converges to the greatest lower bound.

So, the Algorithm 2 converges to a point (w_1, b_1, t) which is an optimum solution to (4.18).

4.4.2 An Estimate of $k_{\max}$ in Algorithm 2 for an ε-Precision Solution

Let

$$P_1^{\min} = \lim_{k\to\infty}\Big\{P_1(w_1^k, b_1^k, t^k) \quad : \quad Y - (Xw_1^k + b_1^k e) \geq \varepsilon_1 e - \xi^k, \xi^k \geq 0\Big\}.$$

We say a point (w_1^k, b_1^k, t^k) is an ε-precision solution to (4.18) if $\left\|P_1(w_1^k, b_1^k, t_1^k) - P_1^{\min}\right\| < \varepsilon$. In this subsection, we endeavour to estimate $\bar{k} = k_{\max}$ so that $\left\|P_1\left(w_1^{\bar{k}}, b_1^{\bar{k}}, t_1^{\bar{k}}\right) - P_1^{\min}\right\| < \varepsilon$.

Theorem 4.1 *If the Newton's method with inexact line search is applied in the Alternating Optimization for Res-TSVR, then to find ε-precision solution to (4.18), the maximum number of iterations $k_{\max}$ is $O(\log(\frac{1}{\varepsilon}))$.*

Proof: We note that

$$
\begin{aligned}
& P_1(w_1^{k+1}, b_1^{k+1}, t^{k+1}) - P_1(w_1^k, b_1^k, t^k) \\
= & \left\{P_1(w_1^{k+1}, b_1^{k+1}, t^{k+1}) - P_1(w_1^{k+1}, b_1^{k+1}, t^k)\right\} + \left\{P_1(w_1^{k+1}, b_1^{k+1}, t^k) - P_1(w_1^k, b_1^k, t^k)\right\} \\
= & \left\{\frac{C_1\beta}{2}\sum_{i=1}^{m}\left[\left(\psi(t_1^{k+1}) - \psi(t_1^k)\right) - \hat{\eta}P_{\text{Hinge}}(z_i^k)(t_i^{k+1} - t_i^k)\right]\right\} \\
& + \tfrac{1}{2}\left(\left\|Y - (Xw_1^{k+1} + b_1^{k+1}e)\right\|^2 - \left\|Y - (Xw_1^k + b_1^k e)\right\|^2\right) \\
& + \tfrac{1}{2}\Big\{\left\|w_1^{k+1}\right\|^2 + b_1^{k+1^2} - \left\|w_1^k\right\|^2 - b_1^{k^2}\Big\}.
\end{aligned}
$$

Out of the three terms in the last expression, for the last two terms, a computational cost of $O(\log\log(\frac{1}{\varepsilon}))$ is required to update w_1^k, w_2^k, b_1^k and b_2^k to the ε-precision point.

The expression in the first term for the update of t_1 and t_2 shows that one requires a computational cost of $O(\log(\frac{1}{\varepsilon}))$ to reach an ε-precision t_1 and t_2.

Therefore, the total computational cost to achieve an ε-precision solution to (4.18) a computational cost of $O(\log\log(\frac{1}{\varepsilon})) + O(\log(\frac{1}{\varepsilon})) \simeq O(\log\frac{1}{\varepsilon})$ is required. □

4.5 Numerical Experiments and Results

This section presents the experiments and their results to illustrate the superiority (in terms of robustness towards different types of noises) of the proposed method over

existing methods, SVR, TSVR, W-ETSVR, and pin-TSVR. Towards this direction, ten data sets were used which included three synthetic data sets and seven real-world data sets. All these experiments were performed over MATLAB 2013a on a system with an Intel i7 processor with 10GB RAM. To make the process a bit simpler, the same regularization parameter and error bounds were used for both the TSVRs. The error bound was given a value of 0.05, which was fixed for all the experiments. This value can be changed. The values of optimal $\hat{\eta}$ and C_1, C_2 were obtained by using 10-fold cross-validation. All the data sets were normalized in the range of $[0, 1]$.

4.5.1 Performance Criteria

In the experiments, the proposed method is compared with the existing methods based on three performance criteria: normalized mean square error (NMSE), root mean square error (RMSE), and R^2. In addition to these, the computational time is also reported for all the methods. The performance metrics used are the sum of a squared estimate of error (SSE), total sum squares (TSS), the sum of squares of residuals (SSR), RMSE, NMSE, and coefficient of determination (R^2). These are defined in Table 4.1. In this table, the actual target value is represented by Y, and the predicted target value is represented by $\hat{Y}$. Also, $\bar{Y}$ represents the mean of Y values.

Table 4.1: Performance Metrics

SSE = $\sum_{i=1}^{m}(Y_i - \hat{Y}_i)^2$	SST = $\sum_{i=1}^{m}(Y_i - \bar{Y}_i)^2$
SSR = SSE	RMSE = $\sqrt{\text{mean}(SSE)}$
NMSE = SSE/SST	R^2 = 1-SSR/SST

4.5.2 Performance on Synthetic Data Sets

In this subsection, the performance of the proposed Res-TSVR is demonstrated on three synthetic data sets that have been used in [98] and [100]. These synthetic data sets are defined in Table 4.2.

Table 4.2: Synthetic Data Sets Used For Experimentation Purposes

Synthetic Data Set 1	$y = \frac{\sin(x)}{x}, \quad x \in [-4\pi, 4\pi]$
Synthetic Data Set 2	$y = 0.2\sin(2\pi x) + 0.2x^3 + 0.3, \ x \in [0, 2]$
Synthetic Data Set 3	$y_i = x_i + 2(\exp(-16x_i^2)), \quad x_i = 0.01(i-1)$

In all the data sets, 500 samples were generated and divided into a 70:30 ratio for training and testing samples. First, the performance of the proposed method was compared with the existing techniques over clean data (results are shown in Table 4.3). After that, the randomly selected 10% of the training data was corrupted by two types of noises, Gaussian noise, and uniform noise. The Gaussian noise was added with mean = 0 and a standard deviation of 0.2. The results are shown in Table 4.4. Similarly, uniform noise was added in the training set, and the results are tabulated in Table 4.5.

It should be noted here that all the experiments were performed using the radial basis function (RBF) kernel. In Tables 4.4 and 4.5, one more column (last column) is added which consists of the optimal parameters used in these algorithms. The parameters which are not used in the respective algorithms are marked by a dash '-'. Note that p used in the last column is the parameter of squared pinball loss function such that $0 \leq p \leq 1$ [100]. The table entries with the least RMSE and NMSE while the highest R^2 are marked in bold.

Table 4.3: Comparison of Various Techniques over Synthetic Data Sets using RBF Kernel (Without Noise)

Data Sets	Techniques	RMSE	NMSE	R^2	Time (in sec)	(C, γ, $\hat{\eta}$, p)
Synthetic Data Set 1	SVR	0.0428	0.5245	0.9790	0.00624	(1, 1, - , -)
	TSVR	0.0071	0.0005	0.9744	0.6459	(10^{-5}, 5, -, -)
	W-ETSVR	0.0044	0.0002	0.9815	0.0038	(10, 1, -, -)
	pin-TSVR	0.0206	0.0047	0.8948	3.5714	(10^{-2}, 3, -, 0.4)
	Res-TSVR	**0.0027**	**0.00002**	**0.9999**	2.6659	(1, 0.125, 11, -)
Synthetic Data Set 2	SVR	0.0370	0.7911	0.9911	0.0027	(4, 3, -, -)
	TSVR	0.0292	0.0049	0.9837	0.8269	(10^{-5}, 3, -, -)
	W-ETSVR	0.0450	0.0115	0.9711	1.6723	(10, 3, -, -)
	pin-TSVR	0.0537	0.0250	0.9513	1.9162	(10^{-1}, 3, -, 0.2)
	Res-TSVR	**0.0084**	**0.00004**	**0.9996**	2.9290	(16, 16, 16, -)
Synthetic Data Set 3	SVR	0.0358	0.0023	0.9616	0.0539	(5, 5, -, -)
	TSVR	0.0111	0.0007	0.9896	1.2110	(10^{-1}, 16, -, -)
	W-ETSVR	0.0269	0.0045	0.9736	1.4264	(10, 16, -, -)
	pin-TSVR	0.0452	0.0127	0.9245	1.8280	(10^{-1}, 16, -, 0.2)
	Res-TSVR	**0.0078**	**0.00003**	**0.9996**	2.0003	(2, 16, 16, -)

Table 4.4: Comparison of Various Techniques over Synthetic Data Sets using RBF Kernel (With Gaussian Noise $\mathcal{N}(0, 0.2)$)

Data Sets	Techniques	RMSE	NMSE	R^2	Time (in sec)	(C, γ, $\hat{\eta}$, p)
Synthetic Data Set 1	SVR	0.0381	0.0135	0.9864	0.00157	(4, 3, -, -)
	TSVR	**0.0082**	0.000629	**0.9994**	4.3318	(10^{1}, 0.125, -, -)
	W-ETSVR	0.0102	0.000957	0.9990	2.5862	(10, 1, -, -)
	pin-TSVR	0.0104	0.0010	0.9990	3.6212	(10^{-1}, 0.125, -, 0.4)
	Res-TSVR	**0.0082**	**0.000628**	**0.9994**	4.9137	(1, 0.125, 11, -)
Synthetic Data Set 2	SVR	0.0358	0.6987	0.9920	0.0051	(5, 3, -, -)
	TSVR	0.0214	0.0028	0.9972	0.8283	(10^{-10}, 16, -, -)
	W-ETSVR	0.0429	0.0113	0.9887	1.5140	(10, 16, -, -)
	pin-TSVR	0.0435	0.0116	0.9884	2.8339	(10^{-3}, 16, -, 0.4)
	Res-TSVR	**0.0160**	**0.0016**	**0.9984**	2.5745	(16, 16, 16, -)
Synthetic Data Set 3	SVR	0.2422	0.3179	0.6820	0.0043	(1, 5, -, -)
	TSVR	0.3708	0.7453	0.2547	1.1886	(10^{-1}, 12, -, -)
	W-ETSVR	0.2458	0.3275	0.6725	1.4564	(1, 12, -, -)
	pin-TSVR	0.2467	0.3297	0.6703	2.6995	(10^{-1}, 12, -, 0.2)
	Res-TSVR	**0.2234**	**0.2704**	**0.7296**	2.7006	(1, 8, 1, -)

4.5.2.1 Discussion and Comparison

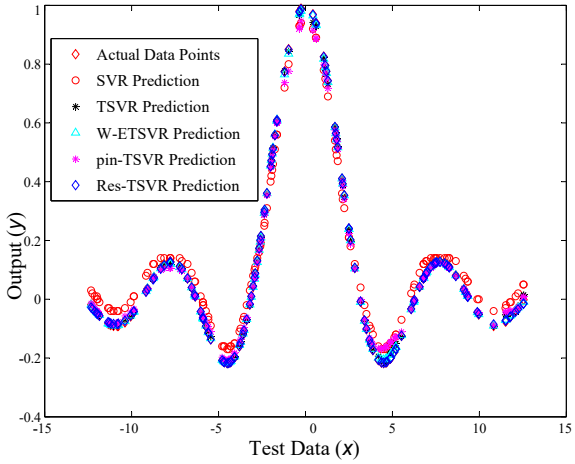
In this part of the study, the results of the approaches mentioned above are discussed over synthetic data sets. First, the performance over uncorrupted data sets (without any noise) is discussed. Table 4.3 shows that the proposed approach performed better than the rest of the techniques considering all the performance metrics. For synthetic data set 1, Res-TSVR showed its best performance at the least γ value of 0.125, while

Table 4.5: Comparison of Various Techniques over Synthetic Data Sets using RBF Kernel (With Uniform Noise $\mathcal{U}(0, 0.2)$)

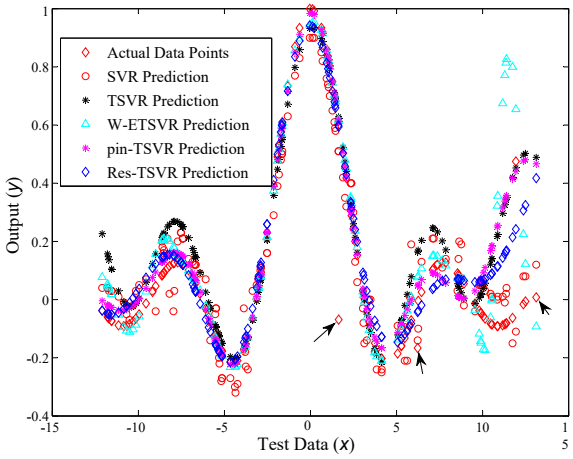
Dats Sets	Techniques	RMSE	NMSE	R^2	Time(in sec)	(C, γ, $\hat{\eta}$, p)
Synthetic Data Set 1	SVR	0.0369	0.0147	0.9852	0.015	(1, 1, -, -)
	TSVR	0.0163	0.0028	0.9972	2.6488	(10^{-5}, 5, -, -)
	W-ETSVR	0.0216	0.0050	0.9950	1.3397	(10, 1, -, -)
	pin-TSVR	0.0207	0.0046	0.9954	4.26254	(10^{-2}, 3, -, 0.4)
	Res-TSVR	**0.0160**	**0.0027**	**0.9973**	3.1227	(1, 3, 5, -)
Synthetic Data Set 2	SVR	0.0304	0.0070	0.9929	0.0149	(5, 3.5, -, -)
	TSVR	0.0294	0.0065	0.9935	3.4252	(10^{-5}, 3.5, -, -)
	W-ETSVR	0.0331	0.0083	0.9917	1.8191	(10, 5, -, -)
	pin-TSVR	0.0559	0.0237	0.9763	2.9633	(10^{-1}, 4, -, 0.4)
	Res-TSVR	**0.0278**	**0.0058**	**0.9942**	7.995	(1, 5, 10, -)
Synthetic Data Set 3	SVR	0.1060	0.0700	0.9299	0.00107	(2, 3, -, -)
	TSVR	**0.1039**	**0.0673**	**0.9327**	3.5778	(10^{-1}, 2, -, -)
	W-ETSVR	0.11127	0.0791	0.9209	1.9630	(1, 3, -, -)
	pin-TSVR	0.1206	0.0906	0.9094	2.9646	(10^{-1}, 12, -, 0.4)
	Res-TSVR	0.1043	0.0677	0.9323	7.9283	(1, 8, 1, -)

for the rest two data sets, it was best for the highest value of the kernel parameter, γ. After adding Gaussian noise in the data set, the proposed technique handled the noisy data better than the rest. Although the performance of Res-TSVR was pretty much similar to TSVR in the case of synthetic data set 1, but in the rest of the two data sets, Res-TSVR outperformed. In Table 4.5, it can be observed that the Res-TSVR performed better than the earlier proposed approaches, but it is very close to TSVR in the case of synthetic data set 3 (the difference in RMSE was 0.0004).

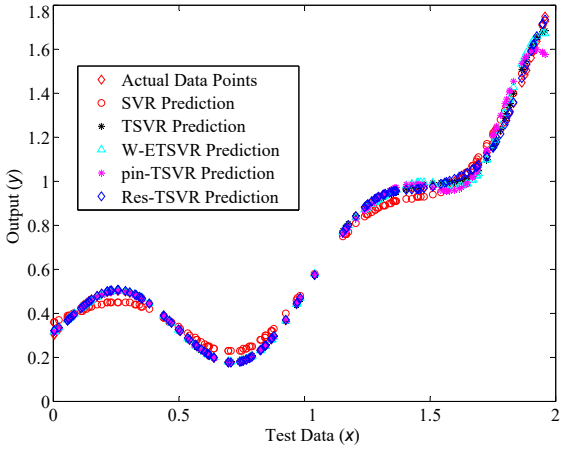
In Figure 4.4, a comparison of all the methods on synthetic data sets is shown. This figure indicates the models' performance when the data set was noise-free. In the case of noisy data, the models were compared over Gaussian noise. Some outliers were also added in the data set to test the performance of all the methods. Arrows in Figures 4.4 (b), (d) and (f) indicate the outliers. From these figures, it was observed that the proposed technique (diamonds in blue) predicted much closer to the actual data points (diamonds in orange) than the rest of the techniques. Although, after the addition of noise, its performance degraded (much distortion in plots on the right side in Figure 4.4, but this degradation is much lesser than the rest. Blue diamonds are



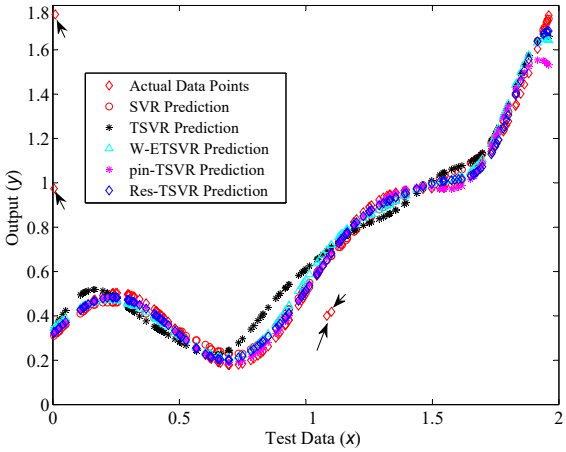
(a) Synthetic data set 1 (without noise)



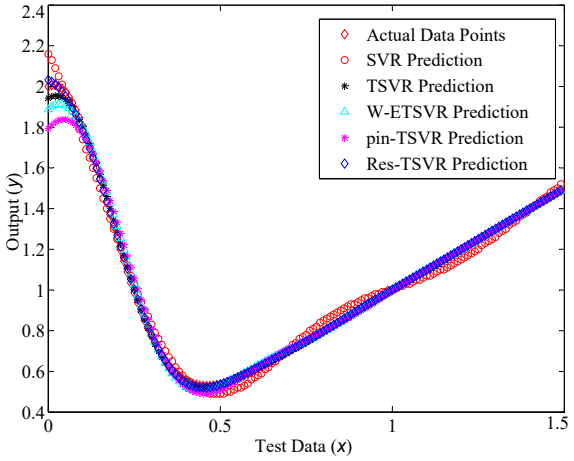
(b) Synthetic data set 1 (with noise)



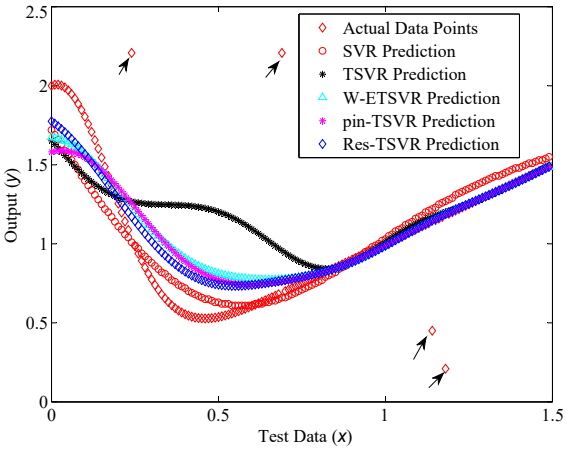
(c) Synthetic data set 2 (without noise)



(d) Synthetic Datset 2 (with noise)



(e) Synthetic data set 3 (without noise)



(f) Synthetic data set 3 (with noise)

Figure 4.4: Comparison of the Proposed Technique with Other Methods in the Absence and Presence of Noise (Outliers are Indicated by Arrows in Case of Noisy Data)

either overlapping the orange diamonds (see Figures 4.4 (a), (c) and (e)) or these are much closer to the orange diamonds. Hence, the Res-TSVR method is more robust than existing techniques.

4.5.3 Performance on Real-world Data Sets

To further investigate the robustness of our proposed scheme, the experiments were performed over publicly available real-world data sets [104–106]. The data sets are mentioned in Table 4.6. Like synthetic data sets, all the data sets were divided in the ratio of 70 : 30, where 70% of the data set was used as the training set while the remaining 30% was the test set. These were selected randomly from the data set. It should be noted here that all the data sets were normalized in the range of $[0, 1]$ before training.

Table 4.6: Real-world Data Sets Used for Experimentation Purposes

Data Sets	No. of Instances	No. of Features	Source
Pollution	60	15	[104]
Servo	167	4	[105]
Head Brain	237	3	[106]
Body Fat	252	14	[106]
Boston Housing	506	13	[104]
Concrete CS	1030	8	[105]
Wine-Red	1599	11	[104]
Abalone	4177	8	[105]

The source from where these data sets were obtained are also mentioned in Table 4.6 to show that the data sets were selected from various sources like UCI, Kaggle, and STATLIB.

The proposed method was then compared with the existing techniques over noisy real-world data sets. Like synthetic data sets, Gaussian noise was first added with mean = 0 and a standard deviation of 0.25 in the randomly selected 10% data set. The results are reported in Table 4.7. Furthermore, results were computed by adding uniform noise in the data sets, and the results are reported in Table 4.8. In Tables 4.7 and 4.8, the

Table 4.7: Comparison of Various Techniques over Real-World Data Sets using RBF Kernel (With Gaussian Noise $\mathcal{N}(0, 0.25)$)

Data Sets	Techniques	RMSE	NMSE	R^2	Time(in sec)	(C, γ, η, p)
Pollution	SVR	0.1408±0.018	0.5364±0.2250	0.4057±0.2250	0.0050	(1, 1, -, -)
	TSVR	0.1403±0.0423	0.5534±0.1995	0.3953±0.0649	0.0405	(10^{-3}, 3, -, -)
	W-ETSVR	0.1275±0.0396	0.6724±0.1290	0.4259±0.1113	0.0323	(8, 3.5, -, -)
	pin-TSVR	0.1549±0.0244	0.7275±0.0701	0.4001±0.2321	0.0626	(10^{-1}, 3.5, -, 0.4)
	Res-TSVR	0.1272±0.0239	0.5260±0.0568	0.5392±0.0657	0.0509	(1, 1, 10, -)
Servo	SVR	0.1172±0.0060	0.1848±0.0172	0.8150±0.0172	0.0030	(1, 1, -, -)
	TSVR	0.1226±0.0040	0.1826±0.0273	0.6155±0.0132	0.1320	(10^{-1}, 3, -, -)
	W-ETSVR	0.1302±0.0259	0.2221±0.0850	0.7044±0.0273	0.2061	(16, 3, -, -)
	pin-TSVR	0.1521±0.0167	0.3265±0.0588	0.4625±0.0407	0.3114	(10^{-1}, 3, -, 0.4)
	Res-TSVR	0.1184±0.0273	0.1825±0.0744	0.8165±0.0586	0.1928	(1, 3.5, 8, -)
Head Brain6	SVR	0.1060±0.0048	0.3840±0.0457	0.6190±0.0360	0.0029	(1, 5, -, -)
	TSVR	0.0981±0.0205	0.4102±0.0805	0.5736±0.0699	0.2162	(10^{-6}, 1, -, -)
	W-ETSVR	0.1020±0.0130	0.3623±0.0522	0.6765±0.1134	0.3438	(4, 1.5, -, -)
	pin-TSVR	0.1142±0.0189	0.4285±0.0601	0.4531±0.0730	0.4339	(10, 1.5, -, 0.4)
	Res-TSVR	0.0915±0.0070	0.3724±0.0305	0.6192±0.0537	0.3846	(1, 0.5, 6, -)
Body Fat	SVR	0.1038±0.0139	0.3873±0.0553	0.5920±0.1380	0.0057	(1, 1, -, -)
	TSVR	0.0968±0.0103	0.4130±0.0598	0.6805±0.1972	0.2454	(10^{-6}, 1.5, -, -)
	W-ETSVR	0.1044±0.0315	0.3883±0.1230	0.6618±0.0862	0.3936	(16, 1.5, -, -)
	pin-TSVR	0.1098±0.0612	0.4948±0.2015	0.6671±0.1541	0.4982	(10^{-1}, 1.5, -, 0.4)
	Res-TSVR	0.0838±0.0123	0.3682±0.0764	0.6845±0.1190	0.4137	(1, 0.5, 6, -)
Boston Housing	SVR	0.0867±0.0042	0.1821±0.0050	0.7080±0.0436	0.0143	(1, 1, -, -)
	TSVR	0.0916±0.0177	0.1522±0.0439	0.8733±0.0518	0.8779	(10^{-4}, 1, -, -)
	W-ETSVR	0.0956±0.0201	0.1640±0.0166	0.7903±0.0627	1.0935	(16, 2, -, -)
	pin-TSVR	0.1259±0.0180	0.3132±0.0218	0.5407±0.0552	1.8518	(10^{-1}, 4, -, 0.2)
	Res-TSVR	0.0941±0.0210	0.1587±0.0237	0.8397±0.0674	1.6897	(4, 3.5, 8, -)
Concrete CS	SVR	0.0935±0.0010	0.2070±0.0102	0.7742±0.0230	0.0565	(1, 1, -, -)
	TSVR	0.1020±0.0172	0.2105±0.0177	0.7099±0.0186	3.6774	(10^{-8}, 1, -, -)
	W-ETSVR	0.0998±0.0136	0.2089±0.0164	0.8830±0.0611	5.8145	(4, 1.5, -, -)
	pin-TSVR	0.1132±0.0184	0.2818±0.0141	0.5985±0.0462	12.0539	(10^{-1}, 5, -, 0.4)
	Res-TSVR	0.0935±0.0017	0.1942±0.0089	0.7446±0.0300	7.5288	(1, 5.5, 8, -)
Wine-Red	SVR	0.1310±0.0046	0.6250±0.0259	0.3714±0.0303	0.1083	(1, 1, -, -)
	TSVR	0.1307±0.0168	0.6315±0.0132	0.3772±0.0513	10.92776	(10^{-6}, 3, -, -)
	W-ETSVR	0.1301±0.0109	0.6433±0.0262	0.4607±0.0453	21.361	(4, 1, -, -)
	pin-TSVR	0.1310±0.0174	0.6606±0.0251	0.3772±0.0386	45.175	(10^{-1}, 1, -, 0.4)
	Res-TSVR	0.1252±0.0291	0.6208±0.0183	0.4660±0.0365	21.6608	(1, 3.5, 8, -)

proposed method is compared with the previously discussed methods based on NMSE, RMSE, R^2 and CPU time. All the experiments were performed using the RBF kernel. In Tables 4.7 and 4.8, the optimal parameters used for the computation (last column) are also shown. All the methods, including the proposed one, equal regularization parameters of the two TSVRs, and the error bound $\varepsilon = 0.05$, were considered.

For SVR [9], the regularization parameter C was chosen from $\{1, 2, \cdots, 16\}$ and the kernel parameter, γ was taken from

$$\left\{2^{-i} + s\ 2^{j} : i \in \{-2, -1, 0, 1, 2, 3, 4\},\ j \in \{0, 1, 2\}, s \in \{0, 1\}\right\}$$

Table 4.8: Comparison of Various Techniques over Real-World Data Sets using RBF Kernel (With Uniform Noise $\mathcal{U}(0, 0.25)$)

Data Sets	Techniques	RMSE	NMSE	R^2	Time(in sec)	(C, γ, η, p)
Pollution	SVR	0.1497±0.034	0.6321±0.1323	0.3678±0.1323	0.00156	(1, 3, -, -)
	TSVR	0.1425±0.0012	0.7839±0.0876	0.2161±0.0120	0.0262	(1, 3, -, -)
	W-ETSVR	0.1619±0.0021	0.7658±0.2887	0.2342±0.023	0.1379	(10^{-3}, 3, -, -)
	pin-TSVR	0.1701±0.0001	0.8569±0.1350	0.1431±0.0001	0.0332	(10^{-1}, 3.5, -, 0.2)
	Res-TSVR	0.1394±0.0020	0.6079±0.1038	0.3921±0.0001	0.0365	(1, 1, 10, -)
Servo	SVR	0.1106±0.0162	0.1756±0.0664	0.8243±0.0664	0.00309	(3, 3, -, -)
	TSVR	0.1233±0.0140	0.2139±0.0466	0.7861±0.0466	0.05221	(10^{-1}, 3, -, -)
	W-ETSVR	0.1476±0.0076	0.2793±0.0603	0.7207±0.0603	0.1377	(16, 3, -, -)
	pin-TSVR	0.1584±0.0142	0.3192±0.0297	0.6808±0.0297	0.1825	(10^{-1}, 3, -, 0.4)
	Res-TSVR	0.1099±0.0130	0.1640±0.0380	0.8360±0.0380	0.1941	(3, 3, 7, -)
Head Brain6	SVR	0.1120±0.0008	0.4106±0.0810	0.589±0.081	0.0039	(3, 3, -, -)
	TSVR	0.1058±0.0098	0.3982±0.0658	0.6018±0.0648	0.8439	(10^{-5}, 0.5, -, -)
	W-ETSVR	0.1040±0.0068	0.3958±0.0773	0.6042±0.0773	0.2584	(16, 1.5, -, -)
	pin-TSVR	0.1219±0.0092	0.4621±0.0520	0.5379±0.0520	0.3197	(10^{-1}, 1.5, -, 0.4)
	Res-TSVR	0.1026±0.0093	0.3777±0.0697	0.6223±0.0697	0.4921	(1, 0.5, 3, -)
Body Fat	SVR	0.1139±0.0158	0.5141±0.1135	0.4858±0.1135	0.00549	(3, 3, -, -)
	TSVR	0.1024±0.0207	0.4064±0.1245	0.5936±0.1245	0.9244	(10^{-6}, 1.5, -, -)
	W-ETSVR	0.1025±0.0135	0.4130±0.0879	0.5870±0.0879	0.3584	(16, 1.5, -, -)
	pin-TSVR	0.1085±0.0144	0.4385±0.0629	0.5615±0.0629	0.3825	(10^{-1}, 1.5, -, 0.4)
	Res-TSVR	0.0991±0.0135	0.3834±0.1165	0.6166±0.1165	0.4580	(1, 1.5, 7, -)
Boston Housing	SVR	0.1010±0.01762	0.2456±0.083	0.7544±0.083	0.1079	(5, 5, -, -)
	TSVR	0.0976±0.00095	0.2285±0.0369	0.7715±0.0369	3.5566	(10^{-4}, 1, -, -)
	W-ETSVR	0.1040±0.0179	0.2367±0.0587	0.7633±0.0587	1.0862	(16, 2, -, -)
	pin-TSVR	0.1180±0.0178	0.3517±0.0569	0.6483±0.0569	1.6892	(10^{-1}, 4, -, 0.2)
	Res-TSVR	0.0959±0.0114	0.2271±0.0583	0.7729±0.0583	1.6865	(1, 1.5, 8, -)
Concrete CS	SVR	0.0942±0.0056	0.2030±0.0452	0.7969±0.0230	0.04527	(5, 5, -, -)
	TSVR	0.1015±0.0037	0.2391±0.0196	0.7609±0.0196	12.8890	(10^{-6}, 1.5, -, -)
	W-ETSVR	0.1069±0.0048	0.2852±0.0297	0.7148±0.0297	6.1521	(4, 1.5, -, -)
	pin-TSVR	0.1129±0.043	0.2917±0.0141	0.7083±0.0283	11.1798	(10^{-1}, 5, -, 0.4)
	Res-TSVR	0.0921±0.0046	0.1964±0.0269	0.8036±0.0270	14.9959	(3, 3, 7, -)
Wine-Red	SVR	0.1299±0.0039	0.6307±0.0148	0.3692±0.0148	0.7145	(1, 1, -, -)
	TSVR	0.1269±0.0036	0.6480±0.0317	0.3520±0.0317	32.851	(10^{-6}, 3, -, -)
	W-ETSVR	0.1272±0.0040	0.6284±0.0302	0.3716±0.0302	32.5663	(4, 1.5, -, -)
	pin-TSVR	0.1262±0.0028	0.6319±0.0186	0.3681±0.0186	35.333	(10^{-1}, 1, -, 0.4)
	Res-TSVR	0.1259±0.0056	0.6232±0.0390	0.3768±0.0390	21.0669	(1, 3.5, 7, -)

Table 4.9: Ranks of Various Techniques over All the Data Sets with Gaussian Noise

Techniques	Metrics	Syn Data 1	Syn Data 2	Syn data 3	Pollution	Servo	Head Brain 6	Body Fat	Boston Housing	Concrete CS	Wine-Red
SVR	RMSE	5	3	2	4	1	4	3	1	1	4
	NMSE	5	3	2	2	3	3	2	4	3	2
	R^2	5	3	2	3	2	3	5	4	2	4
	Average	5	3	2	3	2	3.33	3.33	3	2	3.33
TSVR	RMSE	1	2	5	3	3	2	2	2	3	3
	NMSE	2	2	5	3	1	4	4	1	4	3
	R^2	1	2	5	5	4	4	2	1	4	3
	Average	1.33	2	5	3.66	2.66	3.33	2.33	1.33	3.66	3
W-ETSVR	RMSE	3	4	3	2	4	3	4	4	2	2
	NMSE	3	3	3	4	4	1	3	3	3	4
	R^2	2	2	3	2	3	1	3	3	1	2
	Average	2.66	3	3	2.66	3.66	1.66	3.33	3.66	2	2.66
pin-TSVR	RMSE	3	5	4	5	5	5	5	5	4	3
	NMSE	4	4	4	5	5	5	5	5	5	5
	R^2	2	5	4	4	5	5	4	5	5	3
	Average	3	4.66	4	4.66	5	5	4.66	5	4.66	3.66
Res-TSVR	RMSE	1	1	1	1	2	1	1	3	1	1
	NMSE	1	1	1	1	2	2	1	2	1	1
	R^2	1	1	1	1	1	2	1	2	3	1
	Average	1	1	1	1	1.66	1.66	1	2.33	1.66	1

for synthetic data sets and $\{a + b : a \in \{0, 1, \cdots, 15\}, b \in \{0, 0.5\}\}$ for real-world data sets. The same set of γ's was used for all the regressors. For TSVR [33], the regularization parameters was taken as

$$C_1 = C_2 \in \Big\{10^{-i} : i \in \{-10, -9, -8, \cdots, -1, 0\}\Big\}.$$

Similarly, for W-ETSVR [98],

$$C_1 = C_2 \in \Big\{2^{-i} : i \in \{0, 1, 2, 3, 4, 5\}\Big\} \text{ and } v_1 = v_2 = 2^{-8}.$$

The pin-TSVR [100] had the regularization parameters in the range of $\Big\{10^{i} : i \in \{-5, -4, \cdots, 5\}\Big\}$ and the value of p lied in $\Big\{0.1, 0.2, 0.3, 0.4, 0.5\Big\}$. Lastly, for Res-TSVR, the regularization parameters C_1 and C_2, were picked from

$\Big\{2^{-i} : i \in \{-3, -2, -1, \cdots, 3, 4\}\Big\}$ with $C_1 = C_2$ and $\hat{\eta}$ from $\{1, 2, \ldots, 16\}$. These sets of parameters were used for both synthetic and real-world data sets.

As the training points were randomly selected, the value of the performance metrics changed with every run. Therefore, the mean of all the various performance metrics was computed over ten runs with the standard deviation.

The proposed model was evaluated and compared with existing approaches based on the standard performance metrics like RMSE, NMSE, and R^2. Ranks are also assigned to the models based on all three performance metrics to make a fair comparison. For RMSE and NMSE, the lowest rank is assigned to the lowest RMSE and NMSE values. For R^2, a low rank is assigned to the model with the highest R^2 value. The ranks are presented in Table 4.9. These ranks are assigned based only on the Gaussian noise in synthetic (Table 4.4) and real-world data sets (Table 4.7) as with uniform noise, Res-TSVR outperformed.

Combining the performance of the methods over all the data sets leads to the final average ranks tabulated in Table 4.10, where 'Syn Data' refers to Synthetic Data Set.

This table shows that the proposed method has the lowest rank as compared to the existing techniques. Therefore, it can be concluded that the proposed technique handles the noise better than the other techniques and hence is more robust towards the noise.

4.5.3.1 Discussion and Comparison

Table 4.9 shows that the Res-TSVR achieves the lowest rank for all the synthetic data sets. It was discovered that increasing the number of iteration may improve the performance in some data sets. In the case of Servo and Boston housing data sets, although the proposed method is not outperforming the rest of the methods in terms of RMSE, overall, the average rank of the proposed method over these data sets is better than the rank of the state of the art methods (Please see Table 4.9).

Further, computing the final average rank of all the regressors (Table 4.10), Res-TSVR gets the lowest rank of 1.33. The second-lowest rank is 2.82, which W-ETSVR achieved. This shows that the proposed method is more accurate and robust in terms of noise than the state of art methods.

Table 4.10: Average Ranks of all the Techniques

Techniques	Syn Data 1	Syn Data 2	Syn Data 3	Pollution	Servo	Head Brain	Body Fat	Boston Housing	Concrete CS	Wine-Red	Rank
SVR	5	3	2	3	2	3.33	3.33	3	2	3.33	2.99
TSVR	1.33	2	5	3.66	2.66	3.33	2.33	1.33	3.66	3	2.83
W-ETSVR	2.66	3	3	2.66	3.66	1.66	3.33	3.66	2	2.66	2.82
pin-TSVR	3	4.66	4	4.66	5	5	4.66	5	4.66	3.66	4.43
Res-TSVR	1	1	1	1	1.66	1.66	1	2.33	1.66	1	**1.33**

The statistical significance test was also performed using the paired t-test. The null hypothesis is that there is no significant improvement with the proposed algorithm, Res-TSVR. Table 4.11 lists the p-values [107] considering the performance criterion RMSE. The proposed method is compared with the previously described techniques at 1% significance level in this table. It is evident from Table 4.11 that the null hypothesis is rejected at 1% significance level for majority of the cases. Hence, the proposed method is superior in terms of various metrics and is also more robust to Gaussian and uniform noise. For experimentation, all the results are presented according to the increased

Table 4.11: Results of p-Significance Test Comparing Res-TSVR with Existing Methods at 1% Significance Level

Data Set	SVR	TSVR	W-ETSVR	pin-TSVR
Pollution	0.0686	0.0160	0.0051	0.0030
Servo	0.0089	0.4863	0.8299	0.0004
Head Brain	0.0067	0.3659	0.0464	0.0011
Body fat	0.0056	0.0034	0.0026	0.0048
Boston housing	0.0744	0.0091	0.0053	0.0000
Concrete CS	0.0000	0.3493	0.0059	0.0000
Wine-red	0.6092	0.3955	0.0068	0.0011

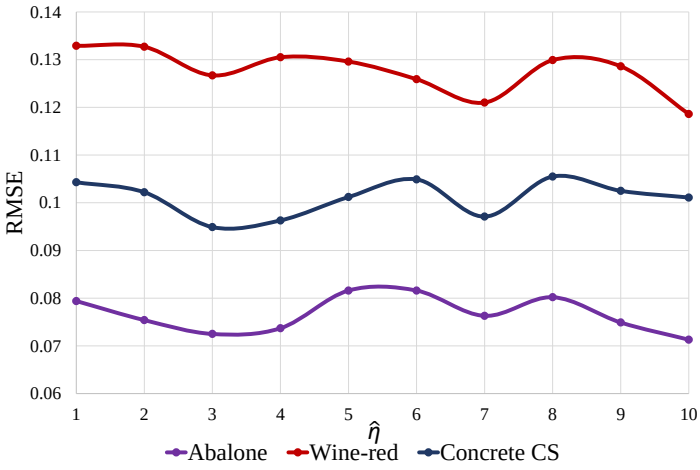
number of instances. Also, one data set, Abalone [105], was used for identifying the effects of $\hat{\eta}$ and other parameters on the performance of the proposed algorithms. As in the proposed algorithms, the number of iterations was not fixed in our experiments; the time taken was reported by a single iteration corresponding to different data sets.

4.6 Discussion About the Rescaling Parameter, $\hat{\eta}$

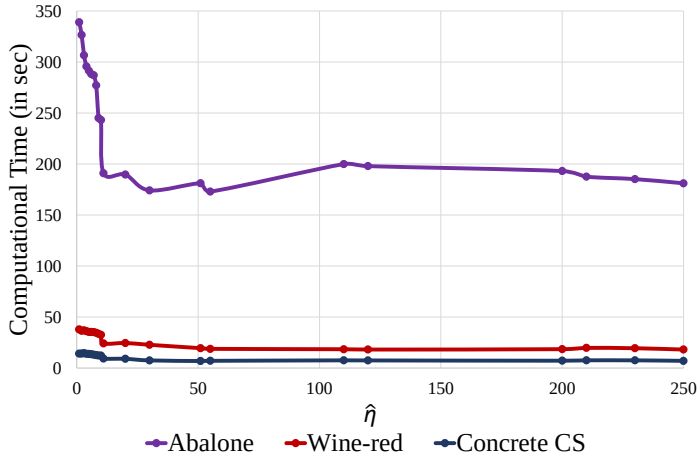
This part discusses the effects of $\hat{\eta}$ on the performance of our proposed method. Abalone data set [105] was used with the already used wine-red and concrete CS data sets in this subsection. Figure 4.5(a) describes the effect of $\hat{\eta}$ (X-axis) on RMSE values (Y-axis).

It can be observed that the values of the RMSE increased initially, and then it decreased when $\hat{\eta}$ reaches the value of 10. Sometimes, a minimum value of RMSE can be obtained at $\hat{\eta} = 7$. Please note that the values of R^2 and NMSE also changed accordingly. In these experiments, $\mathcal{N}(0, 0.2)$, $\gamma = 5.5$ and regularization parameter as 1 were considered.

Till now, we considered the value of $\hat{\eta}$ up to 10; but how will the computational time get affected if the value of $\hat{\eta}$ goes beyond 10? This part discusses the variation of computational time with $\hat{\eta}$ values. Figure 4.5(b) shows that the computational time (in sec) decreases with increasing $\hat{\eta}$ values. In this figure, three data sets were used with the highest number of instances (to explicitly show the variation of computation time with $\hat{\eta}$). From Figure 4.5(a) and (b), it is observed that



(a) RMSE with Increasing $\hat{\eta}$



(b) CPU Time with Increasing $\hat{\eta}$

Figure 4.5: Effect of Increasing $\hat{\eta}$ Values on RMSE and Computational Time

(i) the minimum RMSE is generally achieved at $\hat{\eta} = 3$, 7 or 10, and

(ii) CPU time decreases with increasing $\hat{\eta}$ values.

From the presented work, it can be concluded that Res-TSVR achieves better robustness considering RMSE, NMSE, and R^2 for both synthetic and artificial data sets. A noticeable change in performance metrics is also shown by evaluating various models before and after adding noise (synthetic data sets). The experiments were performed on a diverse range of noises to cover the maximum possibilities of variation. Before concluding, the performance (based on RMSE) of all the techniques is compared over

different SNR. Towards this direction, synthetic data set 1 was used for this experimentation. Figure 4.6 shows that as the value of SNR increases, the RMSE decreases for all the techniques, but the proposed technique consists of the least RMSE for these SNRs (red curve is at the bottom). Gaussian noise was added in the Synthetic data set 1 to obtain different SNRs. Next, a summary of the work is given.

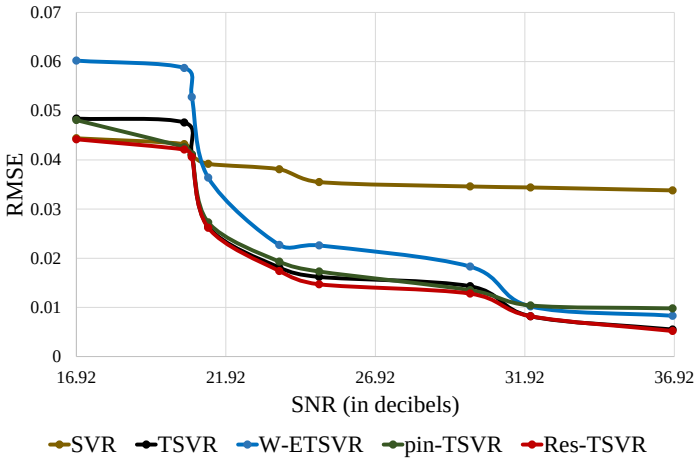


Figure 4.6: The RMSE of Different Regressors Against SNR

4.7 Summary

In this chapter, a novel TSVR model was proposed with rescaled hinge loss. Since the corresponding optimization problem was non-convex, it was transformed into a dual form for easy implementation. To implement the dual form, an algorithm named Res-TSVR was proposed. In this work, robustness was achieved against Gaussian and non-Gaussian noise in the data set. A rigorous comparison of the proposed approach was shown with SVR, TSVR, W-ETSVR, and pin-TSVR based on RMSE, NMSE, and R^2. Ten standard data sets were used for the comparison, of which three were synthetic data sets, and seven were real-world data sets. Also, Abalone data set was used to study the effects of parameters on the performance metrics. It was observed that the minimum RMSE value was generally achieved at $\hat{\eta} = 3$, 7, and 10. The

proposed method had either better or the same performance as its competitors in all the studied data sets. Further, in terms of all the performance measures, Res-TSVR ranked among the top three except Boston Housing and Servo data sets. However, overall the average rank (across various measures) of the proposed method was better than the rest of the methods. Thus, experimental results over synthetic and real-world data sets demonstrated the efficiency and robustness of Res-TSVR over the existing methods. Statistical hypothesis testing (p-test) showed that Res-TSVR outperformed others at a 1% significance level. Also, the performance of all the techniques was compared over different SNRs. This also has revealed that Res-TSVR gave the least value of RMSE for different SNRs.

Besides the advantages mentioned above, there are some limitations to the proposed approach. From (4.35), (4.36), (4.40) and (4.41), it was observed that one requires to optimally select the value of $\hat{\eta}$. In this work, 10-fold cross-validation was used to select this value, but this selection was very time-consuming in the case of large data sets. The method may fail to find the optimal solution for large data sets if the training time is limited. The solution could be the use of a heuristic method to obtain the optimal value of $\hat{\eta}$.

Also, to compute the dual formulation of the problem statement, an alternating minimization technique was used, which also affected the performance of the model because of the following reasons:

(i) It involved a higher level of abstraction in each iteration. It solved a small optimization problem in each iteration, unlike standard methods—Newton methods and gradient descent—which involved multiplications, gradient finding, and Hessians as base operations. This increased the time consumption of the model.

(ii) Although, this technique can find an approximate solution quickly, it may take considerable time to converge to a high accuracy solution. Therefore, it is preferable where a modest accuracy model is sufficient.