Neural Computing and Applications (2020) 32:11173–11194
https://doi.org/10.1007/s00521-019-04627-6

REVIEW ARTICLE

Robust statistics-based support vector machine and its variants: a survey

Manisha Singla[1] · K. K. Shukla[1]

Received: 7 March 2019 / Accepted: 22 November 2019 / Published online: 2 December 2019

Abstract
Support vector machines (SVMs) are versatile learning models which are used for both classification and regression. Several authors have reported successful applications of SVM in a wide range of fields. With the continuous growth and development in machine learning using SVM, it was observed that SVM also has some limitations. This paper focuses on limitation regarding its boundary, i.e., sensitivity to noise or outliers in the dataset. Researchers have proposed many variants and extensions of SVM to make it robust. This paper gives an overview of the developments in the field of robust statistics in *support vector machines and its variants*. This paper includes an up to date survey of the research development in the field of robustness in SVM and its extensions. It also includes a *discussion* part which not only discusses the pros and cons of the proposed approaches but also highlights some important future directions in it. This paper would be helpful for researchers working in the field of robust statistics as well as supervised machine learning. This study would also encourage the researchers to work further in the development of SVM and even its variants to improve them.

Keywords Robust statistics · Noise · Outliers · Support vector machines · Optimization techniques · Twin SVM · One-class SVMs · Multi-class SVM · Fuzzy SVM

1 Introduction

Robustness in machine learning is broadly 'Algorithmic Stability.' It means that the algorithm would not face much deterioration when there are some changes in the training and testing dataset. There are many reasons due to which the values in the dataset may get slightly altered and because of this alteration, the algorithm may not perform as per the expectation. Therefore, the algorithm should perform well even after such adjustment in the dataset. This is how an algorithm can be made robust. This survey discusses the robust statistics based on SVM and its variants. This paper comprises of variants of SVM which are made robust by various researchers to overcome the limitation of sensitive to outliers or noise in the dataset. As machine learning is growing immensely in almost all fields, it becomes necessary to get better and accurate results from it. However, the performance is adversely affected in the field (in realistic settings) because of the presence of noise and outliers in the data. Many researchers have worked in this direction so that the model generated would not get disrupted by these problems. This paper focuses on existing techniques which are used to solve these problems. This work addresses the following research queries:

Query1: How can robust statistics help in maintaining algorithmic stability in the presence of outliers in the data?
Query2: What was the need for different variants of SVM?
Query3: What happens when different loss functions are considered in place of Hinge loss function?
Query4: How to choose the best loss function for a model?

✉ Manisha Singla
manishasigla.rs.cse17@iitbhu.ac.in

K. K. Shukla
kkshukla.cse17@iitbhu.ac.in

[1] Department of Computer Science and Engineering, Indian Institute of Technology (Banaras Hindu University), Varanasi, India

These queries are discussed in detail with the research work of various authors which can help in answering these queries. This survey includes research papers both published, preprints, thesis, dissertations, and books which are primarily focused on robustness in machine learning. We have presented a scenario of work-in progress in this field. It primarily deals with the effects of outliers and noise on SVM and its variants. We have included **126** papers (mainly from 2001 to 2018) in our study and some books which can be referred by the readers. Although many other related works can also be included in this paper, we strongly feel that the work contained in this paper would be very informative and would provide a comprehensive research view to the researchers who are interested in this area. The various digital libraries accessed during these papers selection are presented in Fig. 1.

Here, digital libraries accessed for the survey are listed according to the number of papers taken from these libraries. The next section of the work starts with the introduction of SVM followed by various variants of it which were proposed to add robustness toward outliers or label noise.

2 Support vector machine

2.1 Definition and preliminaries

Robust Statistics is deployed to ensure excellent performance of the algorithm corresponding to data obtained from various sources under a probabilistic distribution. Now the data usually obtained from different sources may contain *outliers* in it. These outliers can directly affect the performance of the algorithm so to tackle these problems, various techniques were used to make the algorithm robust to outliers. In this section, we will discuss Robust Statistics from *robustness to outliers* point of view. The list of symbols commonly used in this paper is given below. The rest of the symbols are defined at the place of their use.

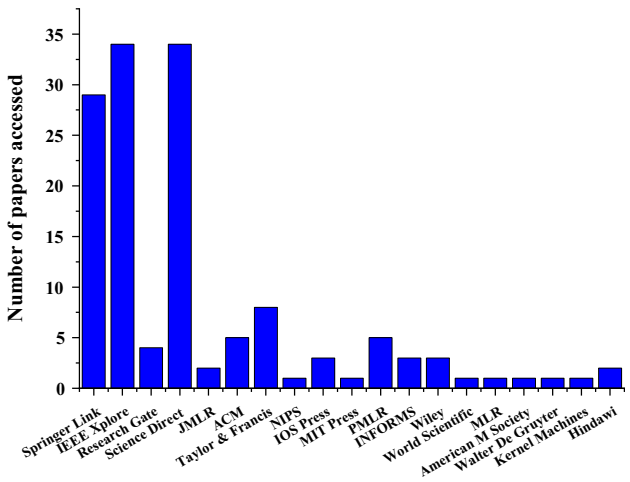

Fig. 1 Various digital libraries accessed for the survey showing number of papers retrieved

w	Weight vector
x_i	Feature vector
y	Target vector
ξ	Slack variable
b	Bias Vector
C	Regularization parameter
$\hat{y}$	Output variable
x^T	Transpose of x
ε	Error bound (in regression)
$\|.\|$	l_2 norm
$\|.\|$	Absolute function

2.2 Robust statistics with SVM

Query 1: How can robust statistics help in maintaining algorithmic stability in the presence of outliers in the data?

Machine learning has its roots in Statistical Learning Theory [102] which tackles dependency estimation problems. These dependency estimation problems may get affected if the data are contaminated with outliers [4] or some noise in it which leads to robust statistical learning. The papers which are discussed further in this section depict the generalized scenario as shown in Fig. 2.

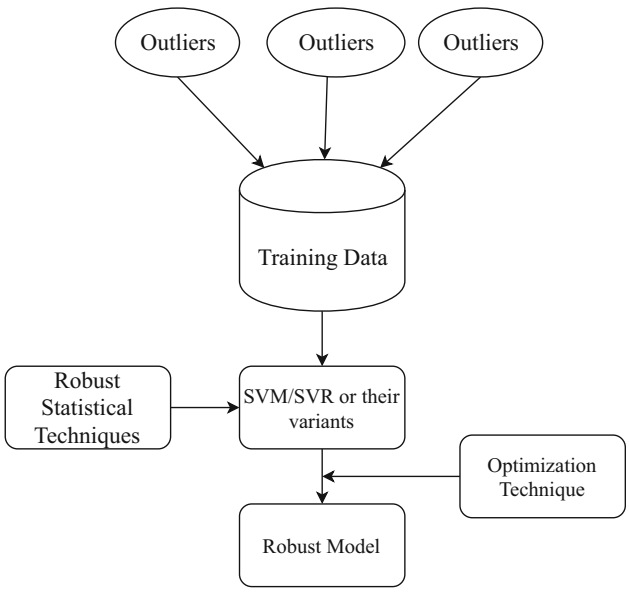

Fig. 2 Generalized framework of robust statistics

Figure 2 shows that when training data are contaminated with outliers or noise, the SVM/SVR model is trained using robust statistical techniques which is further optimized using optimization techniques to get a robust model. Therefore, Query 1 is addressed by every paper included in this work. SVMs are the supervised machine learning models which are capable of performing both the classification and regression tasks. SVMs were initially designed for two-class classification but later it was extended to multi-class classification and regression tasks. SVM efficiently works for both linear as well as nonlinear classification. The equation of the separating hyperplane in case of SVM can be seen from Cortes and Vapnik [23]. The optimization problem corresponding to SVM is a convex quadratic problem. Similarly, for regression problems, SVM is known as support vector regression (SVR). Although, both SVM and SVR possesses the same properties, as the target variable in case of regression comprises of real numbers, so maximum margin separation is difficult in SVR. In SVR, a margin of tolerance ε is there which is required to be defined prior to solving the optimization problem. The SVR problem for the training data set of m points $\{x_i, y_i\}_{i=1}^{m}$ is given by:

$$\min \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \tag{1}$$

subject to

$$\begin{cases} y_i - (w^T x_i + b) \le \varepsilon + \xi_i, \\ (w^T x_i + b) - y_i \le \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \ \ge 0 \end{cases} \tag{2}$$

In the above equation, w is the weight vector, b is the bias term corresponding to the hyperplane

$$f(x_i) = w^T x_i + b \tag{3}$$

with C as the regularization parameter. We have given the equation of soft margin SVR with ξ_i and ξ_i^* are the slack variables introduced to cope up with the otherwise infeasible constraints [88].

Since the origin of SVM/SVR, researchers had a keen interest in this classifier/regressor which originated several new researchers in this area. They moved toward testing the robustness of this classifier/regressor on various applications. On finding the limitation of sensitivity toward outliers, researchers started proposing algorithms which can make the model robust like the one given by Le Thi Hoai and Tao [48] in which authors have proposed an algorithm which can be used for both binary and multi-class classification. The proposed method of truncation can be applied to any convex unbounded loss function which usually suffers because of the presence of outliers in the dataset.

Similarly, another technique was proposed to deal with outliers in the dataset which considered the distance of each data point from the center of the class to evaluate the adaptive margin and then added the averaging technique into the classical SVM [89]. This addition helped them in getting a robust SVM, and it also controlled the amount of regularization. The technique was proved to be very useful as the number of support vectors obtained were also very less. They concentrated on the problem of bullet hole classification and applied their algorithm over it.

This was the time when robust statistics started building in the field of SVM. At that time, several outlier detection algorithms were also developed like an outlier detection method for industrial data which is otherwise difficult to find [39] solving regression problems. Similarly, in the SVR equation,

$$\begin{aligned} \min_{w,b,\xi,\hat{\xi}_i} \ & \frac{1}{2}\|w\|^2 + \frac{\lambda}{m}\sum_{i=1}^{m}(\xi_i + \hat{\xi}_i) \\ \text{s.t.} \ & (\langle w, x_i\rangle + b) - y_i \le \varepsilon + \xi_i, \quad i = 1, \ldots, m \\ & y_i - (\langle w, x_i\rangle + b) \le \varepsilon + \hat{\xi}_i, \quad i = 1, \ldots, m \\ & \xi, \hat{\xi}_i \ge 0, \quad i = 1, \ldots, m \end{aligned} \tag{4}$$

and the dual form is [39]:

$$\begin{aligned} \max \ & \sum_{i=1}^{m} y_i(\alpha_i{}^* - \alpha_i) - \varepsilon\sum_{i=1}^{m}(\alpha_i{}^* + \alpha_i) \\ & -\frac{1}{2}\sum_{i,j=1}^{m}(\alpha_i{}^* - \alpha_i)(\alpha_j{}^* - \alpha_j)K(x_i, x_j) \\ \text{s.t} \ & \sum_{i=1}^{m}(\alpha_i{}^* - \alpha_i) = 0, \\ & 0 \le \alpha_i, \alpha^*{}_i \le \frac{\lambda}{m}. \end{aligned} \tag{5}$$

Here, x_i, y_i are the input vectors and target vectors, respectively, and m is the number of instances with λ as the regularization parameter. In (5), α_i and $\alpha^*{}_i$ are the Lagrange multiplier which is bounded by $\frac{\lambda}{m}$. If the Lagrange multiplier's value is positive, it suggests that the constraint in the primal formulation is active while if it hits the upper boundary, it indicates that the corresponding data point lies beyond the ε-insensitive zone and considered as a slack variable. If the slack variable corresponding to it has a tremendous value, the data point lies outside ε-insensitive zone. Therefore, the authors considered those data points as outliers which were having a large value of the slack variable or whose Lagrange multiplier has upper bound values [39]. The main advantage of this proposed technique is its capability to find outliers even in case of high-dimensional datasets and also with rank deficient datasets.

Several non-convex optimization problems were also solved to get the robust classifier/regressor like the one

proposed by Zhao and Sun [143] which also worked in robust regression using a non-convex loss function for support vector regression (SVR). They made use of Huber loss. This non-convex optimization problem was turned into the convex function using convex–concave Procedure (CCCP) [141] and this convex problem was further solved with Newton-type algorithm to get robust SVR [143].

Till this time, SVM/SVR was extensively used in machine learning. Various applications were proposed in the context of SVM/SVR. Researchers started introducing the extensions and variants of SVM/SVR. As this work also discusses the multiple variants of SVM/SVR, Fig. 3 describes the various modifications of SVM/SVR which worked in the field of robust statistics.

2.2.1 Least square SVM

Toward this direction, firstly SVM was extended to 'Least Square SVM (LS-SVM).' This extension simply made use of square loss function. The optimization problem for LS-SVM is given by [91]:

$$\min_{w,b,\xi} L(w,\xi,b) = \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{m}\xi_i^2 \tag{6}$$

subject to *equality* constraints

$$y_i(w^T x_i + b) = 1 - \xi_i, i = 1,\ldots,m \tag{7}$$

Here, C is the regularization parameter and ξ is the slack variable corresponding to every data point.

Query 2: What was the need for least square variant of SVM?

From the above equation, it can be observed that the purpose of proposing LS-SVM was to make the optimization problem easier to solve. As in the optimization problem of LS-SVM, equality constraint is there which solves linear equations to get the solution instead of solving quadratic programming as proposed in SVM [91]. Now the question arises: Is this variant robust to outliers? The answer is *no*. In this work, the quadratic formulation was transformed into linear equations, but the noise sensitivity was still there because of the hinge loss function. It was observed from the work of Suykens et al. [92] that the standard LS-SVM has two significant limitations of sparseness and robustness in the proposed framework. These limitations were observed and solved by Suykens et al. [92] by proposing *weighted LS-SVM*. To obtain the robust version of LS-SVM, error variables ξ_i were weighted using weighting factor v_i and the resultant optimization problem obtained was [92]:

$$\min_{w,b,\xi} L(w,\xi,b) = \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{m} v_i\xi_i^2 \tag{8}$$

subject to

$$y_i = w^T x_i + b + \xi_i,\ \ i = 1,\ldots,m \tag{9}$$

It was experimentally observed that weighted LS-SVM is more robust than LS-SVM. Authors had also proposed a technique to make the weighted LS-SVM sparse using pruning methods. They suggested that pruning a relatively small amount of least meaningful data points can make a sparse approximation [92]. They used this technique for heavy-tailed Non-Gaussian error distribution and observed

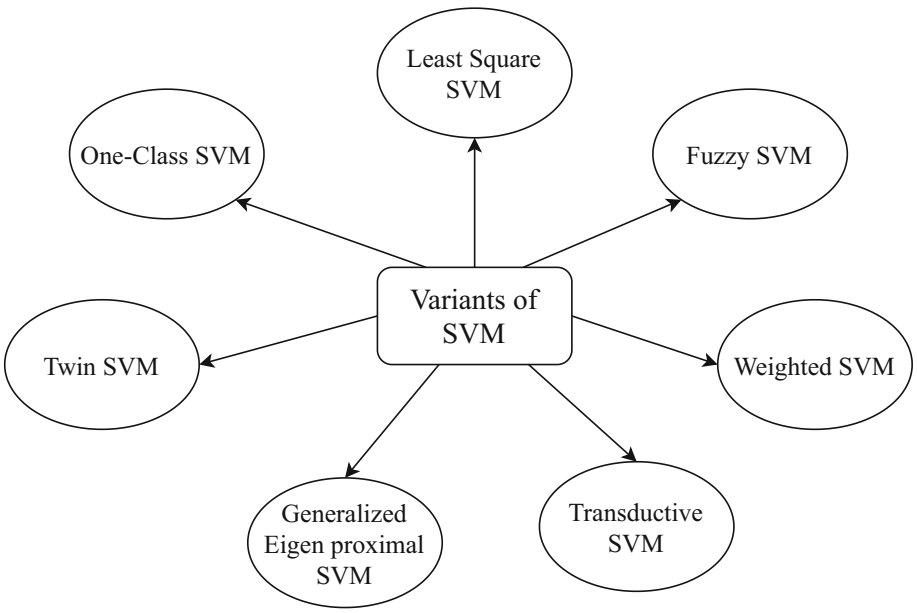

Fig. 3 SVM with its variants

that their approach adds robustness in the regression problems.

Later, this approach of weighted LS-SVM was used and extended by various researchers to solve problems like overfitting and noise in the dataset. Although, SVR has excellent robustness properties against noise, but when the parameters indulged in the optimization problem are not selected correctly, the issue of overfitting may occur. Also, the inclusion of outliers in the dataset may also lead to some serious overfitting. To solve such problems, a novel approach to robustify SVR network was presented by Chuang et al. [21]. In this paper, authors have combined two methods to robust learning theory and SVR to form robust SVR (RSVR) networks. There were two phases proposed for the RSVR networks: The initial phase where the network structure and the initial network weights assigned were determined using SVR theory as shown:

$$\hat{y} = \sum_{i=1}^{S} W_i K(\bar{x_i}, \bar{x}) + b \tag{10}$$

where $\hat{y}$ is the output variable, S is the number of support vectors, $K(\bar{x_i}, \bar{x})$ is the kernel function, W_i and b are the weight vector and bias, respectively .

The next phase is to adjust those weights which were initially assigned in the first phase. In this phase, robust cost functions are used instead of the quadratic standard cost function. The robust cost function used here was:

$$F_R(t) = \frac{1}{N}\sum_{i=1}^{N} \beta[e_i(t)], \tag{11}$$

where t is the epoch number, $e_i(t)$ is the error defined for epoch t and is defined as $e_i(t) = y_i - \hat{y_i}(t)$. β is the robust cost function used in place of standard cost functions. There were various advantages of this proposed approach. Through this proposed approach, the number of hidden nodes could be readily determined by the SVR theory. Overfitting was also handled by this approach efficiently. Learning performance was also improved for any named parameters by employing traditional robust learning approaches.

The work proposed by Jiang et al. [38] also contributed in adding sparseness to the model as this work made use of quadratic Renyi entropy. It improved both the learning speed and the sparseness of the model as Renyi entropy was the evaluating criteria for the working set selection.

Proposal of LS-SVM opened the room for various new researches. LS-SVM was also robustified through many different ways. Several loss functions like ramp loss function were also added with LS-SVM [58] to make it robust to outliers and noise. But a question arises here: What was the need for ramp loss LS-SVM? The answer to this question is given by [58]:

- Ramp loss LS-SVM controlled the sparseness of LS-SVM.
- As the ramp loss function is robust to outliers, ramp LS-SVM was able to incorporate noise and outliers suppression explicitly.

The optimization problem proposed was:

$$\min_{w,b} \frac{1}{2}\|w\| + \frac{C}{2} R_{\varepsilon,t}(y_i f(x_i) - 1) \tag{12}$$

where $R_{\varepsilon,t}(z)$, the ramp loss function, was defined as:

$$R_{\varepsilon,t}(z) = \begin{cases} (t-\varepsilon)^2, & |z| > t, \\ (|z|-\varepsilon)^2, & \varepsilon \le |z| \le t, \\ 0, & |z| < \varepsilon \end{cases} \tag{13}$$

The resultant non-convex problem of ramp LS-SVM was solved using CCCP. It was also observed that the proposed approach allows better parallelization.

2.2.2 Fuzzy SVM

SVM was later merged with fuzzy logic in such a manner that a fuzzy membership value $\sigma \le s_i \le 1$ ($\sigma \ge 0$) was also attached with the dataset $\{x_i, y_i, s_i\}$, $i = 1, \ldots, m$ where m is the number of data points. In 2002, fuzzy membership was firstly applied to SVM [53] where different data points were assigned with different fuzzy membership values to make different levels of contribution to the decision surface. The equation of fuzzy SVM proposed was:

$$\begin{aligned} &\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} s_i \xi_i \\ &\text{subject to} \\ &y_i(w^T \phi(x_i) + b) \ge 1 - \xi_i, \quad i = 1, \ldots, m \\ &\xi_i \ge 0, \quad i = 1, \ldots, m \end{aligned} \tag{14}$$

This provided a method to use fuzzy SVM to classify data points with noise or outliers. Although fuzzy SVM added robustness to the classifier, it was difficult to determine membership values for each data point adaptively. There were specific approaches proposed to deal with this problem. In 2004, researchers introduced two new factors to the training data points: the confident factor and the trashy factor [55]. The authors proposed a heuristic strategy of automatically generating membership function from these two factors and a mapping function [55]. The equation of the optimal hyperplane obtained from this modification was:

$$\begin{aligned} \min\ & \frac{1}{2} w.w + C \sum_{i1}^{m} p_x(x_i)\xi_i \\ & \text{subject to } y_i(w.z_i + b) \geq 1 - \xi_i, i = 1, \ldots, m \\ & \xi_i \geq 0, i = 1, \ldots, m \end{aligned} \tag{15}$$

where $p_x(x_i)$ denotes the probability density function given by [55]:

$$p_x(x) = \begin{cases} 1, & \text{if } h(x_i) \geq h_C, \\ \sigma, & \text{if } h(x_i) \leq h_T, \\ \sigma + (1-\sigma)\left(\dfrac{h(x) - h_T}{h_C - h_T}\right)^d, & \text{otherwise} \end{cases} \tag{16}$$

In the above equation, h_C is the confidence factor and h_T is the trashy factor with the given heuristic function $h(x)$. The experimental results proved that the proposed approach was more feasible than the existing approaches proposed to make the SVM robust toward outliers. Researchers have also suggested various methods for the automatic setting of fuzzy membership value [54].

Fuzzy logic was not only used with the classification problems but also with regression problems [56]. Fuzzy logic was also used in designing other classifiers using SVM or SVR (in case of regression) [56]. The ideas of fuzzy neural networks and fuzzy SVM were combined to form a new fuzzy SVM [96] which was more robust than these two. This method of using fuzzy membership with different classifiers was used in many applications. Fuzzy SVM was used to evaluate credit risk [109], for multi-class text categorization [108], etc.

Like the weighted version of LS-SVM, fuzzy weighted SVR was also proposed by Chuang [19] which can be used further to make fuzzy SVM more robust to outliers or noise.

Fuzzy SVM was also combined with outlier detection algorithm so that the resultant fuzzy SVM can be made insensitive to outliers as suggested by Lee et al. [49]. This approach was based on four steps: First, it applied the outlier detection algorithm on the given training data. Second, it calculated the membership value by using fuzzy sigmoid model. Third, kernel parameter estimation was performed which proceeded toward the last step of applying fuzzy SVM [49]. The advantage of the proposed approach was the use of outlier detection algorithm which made the model robust to outliers and gave better results than the earlier proposed methods.

In all the previous methods discussed on fuzzy SVM, there is one thing common, i.e., all the techniques needed some prior knowledge and assumed some simple data distribution. There was a need for developing a robust membership calculation function which does not consider any simple data distribution. Based on this requirement, a method was proposed which developed a membership function calculation scheme with the help of reconstruction error. The whole calculation was based on the statistics of the data; therefore, no prior knowledge or assumption of more straightforward data was required [33]. Although the proposed method was more robust toward noise than SVM and the variants of it, the computational complexity was high. Also, calculating reconstruction error was expensive in case of large datasets.

Fuzzy SVR was also made robust toward Gaussian noise by making use of the fuzzy triangular technique to represent fuzzy membership values as suggested by Wu and Law [114]. Similarly, based on c-means and even Mahalanobis distance, SVMs were made robust using fuzzy logic [142].

Authors have also tried to mitigate the problem of classification in the presence of outliers, noise, and class imbalance altogether in a single work as proposed by Batuwita and Palade [5]. They merged class imbalance learning (CIL) with fuzzy SVM (FSVM) and named it as FSVM-CIL algorithm. In this algorithm, they assign a fuzzy value or membership value m_i to each of the training examples in such a way that the effect of class imbalance can be suppressed and also to show the class importance of training examples to identify outliers. The equation of soft margin SVM after adding membership value is changed as shown (see [5]:

$$\begin{aligned} \min\ & \left(\frac{1}{2} w.w + C \sum_{i=1}^{l} m_i \varepsilon_i\right) \\ \text{s.t. } & y_i(w.\phi(x_i) + b) \geq 1 - \varepsilon_i \\ & \varepsilon_i \geq 0, \quad i = 1, \ldots, l. \end{aligned} \tag{17}$$

It was concluded that FSVM-CIL performs better than the previously stated algorithm when the class imbalance is there in the datasets.

In-between fuzzy logic with SVM or SVR, in the year 2006, an exciting technique was proposed in which authors tried to make the training phase robust by firstly identifying and then eliminating those training examples which are outliers [123]. They augmented the classical soft margin SVM with the indicator variables which help remove outliers. The objective function was changed in this manner [123]:

$$\min_x \frac{\beta}{2} \left\| \left(\mathrm{w} \right) \right\|^2 + \left(\sum_i \eta_i [1 - y_i (x_i)^T w]_+ \right), \eta_i \in \{0, 1\}. \tag{18}$$

However, in this equation loss function is not an upper bound on the misclassification error so to compensate this,

they have added one more term and defined their loss function as:

$$\eta - \text{hinge}(w, x, y) = \left(\eta[1 - yx^T w]_+ + 1 - \eta\right) \tag{19}$$

The resulting training procedure provides superior robustness to outliers than standard SVM training.

In 2011, another classification problem with outliers or noise was solved using a kernel fuzzy *c*-means clustering-based fuzzy SVM. According to this method, firstly FCM clustering was applied to the training set to cluster each of the two classes. It then found two farthest clusters from each of the class and assigned membership values to it. Lastly, fuzzy SVM was applied to the final classification results [136]. There were so many fuzzy SVM developed until this time, but in these methods, *within-class scatter* was not considered. In 2013, another fuzzy SVM was developed, but this also incorporated within-class scatter in it. The idea behind this method was to maximize the separation between the data points of two different classes but to minimize the within-class scatter [1]. In all the above discussed methods, it was required to initially set the fuzzy membership values for all the training data points which was a difficult task. To avoid setting fuzzy membership values, bilateral truncated loss-based robust SVM was proposed by Yang et al. [135]. To solve the proposed method, the authors used convex–concave procedure and Newton–Armijo algorithm. Although the proposed method was more robust than the previously discussed, the training time of the proposed method was usually more.

LS-SVM has the significant advantage of having equality constraints which makes the overall optimization problem easier to solve, but LS-SVM is sensitive to outliers as already discussed above. Researchers have applied fuzzy logic over LS-SVM to make the model robust to outliers or noises. In 2011, researchers proposed to assign fuzzy membership function to every data point as weight. As it was required to make the model robust, the weights were assigned according to the distance from the data point to the center of its neighborhood and radius of the neighborhood [140]. This also indicated the probability of data points to be an outlier. The important point which should be noted here corresponding to this research is the use of a heuristic method to get the fuzzy membership value. This method had less complexity as the authors did not use iterative training method for computing fuzzy membership value [140].

2.2.3 Weighted SVM

A weighted version of LS-SVM was proved to be effective than LS-SVM; similarly, a weighted version of SVM was also proposed. Similar to the LS-SVM, the basic idea behind this proposal was to assign weights to the training data points so that the algorithm learns the decision surface according to the importance given to data points. The task of weight assignment was done using fuzzy clustering algorithm and kernel-based possibilistic c-means algorithm [133]. The optimization problem obtained after assigning weights W_i was given by:

$$\min_{w,b,\xi} L(w, \xi, b) = \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{m} W_i \xi_i \tag{20}$$

subject to constraints

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \ i = 1, \ldots, m \tag{21}$$

It should be noted here that the weighted SVM was proposed to improve the outlier sensitivity problem of SVM. The complexities associated with training and testing can also be adjusted using the pruning method according to the outliers present in the dataset. This method yielded higher classification rate than standard SVM [133]. This method was later extended to one-class weighted SVM(OWSVM)(discussed later) and iteratively weighted SVM(IWSVM) given by Wu and Liu [116]. Both the methods were better than existing weighted classifier. Amongst these two, IWSVM outperformed OWSVM but the computational time of IWSVM was more than OWSVM as it required iterations as the name indicates. These methods were proposed for both binary and multi-class problems. Different ways of weight assignment were proposed after this approach. In 2013, researchers used association rules to assign weights. This method of weights assignment was better than the previous one as the use of association rules prevented bias to the majority class [57]. This method was suitable for both noisy data as well as imbalanced data.

Authors in 2014 presented a novel robust approach of LS-SVM which was based upon truncated loss function for classification and regression [134]. The optimization problem they considered is given by [134].

$$\min_{w,b} \min_{0 \leq s \leq 1} \frac{1}{2} w^T w + \frac{C}{2}\sum_{i=1}^{l} L_s(w, b, s_i, x_i, y_i) \tag{22}$$

and

$$\min_{w,b} \frac{1}{2} w^T w + \frac{C}{2}\sum_{i=1}^{l} robust_2(w, b, x_i, y_i), \tag{23}$$

where

$$L_s(w, b, s_i, x_i, y_i) = s(y - (w^T \phi(x) + b))^2 + p(1 - s), \quad p \geq 0 \tag{24}$$

and

$$\text{robust}_2(w, b, x_i, y_i) = \min(p, (y - (w^T \phi(x) + b))^2), \quad p \geq 0. \tag{25}$$

To solve the above equations, their dual form was formulated which were solved using CCCP and Newton algorithm explained by Chapelle [11]. The method was proved to be very useful against noise in the dataset. This work also discussed the relationship between weighted SVM and the proposed approach.

Not only SVM, but also least Square SVR (LS-SVR) was also made robust to outliers like the one proposed by Ning et al. [68] using the variational Bayesian framework. The authors of the proposed framework applied the strategy by first learning the parameters of LS-SVR in Bayesian framework and then replacing the Gaussian distribution with Student's t-distribution [68].

In the above discussed approaches, it was very much required to adequately select the parameters like weight parameters and fuzzy membership values. If these parameters are not chosen correctly, the algorithm may not perform as per expectation. Therefore, Chuang and Lee [20] proposed a technique where robust statistics was not required to deal with outliers in the training samples. This strategy comprises two phases: The initial phase where data preprocessing was accomplished, and SVM for regression was used to remove outliers from the training set (which is now known as reduced training dataset). Now, this reduced training set was directly used with non-robust least square SVM for regression.

Query 3: What happens when some robust loss functions are considered in place of hinge loss?

SVM with different loss functions

We have observed that the use of hinge loss function in classification was also replaced by some other loss functions like ramp loss function, pinball loss, rescaled hinge loss, truncated pinball loss, etc. These are the robust loss functions which were used with SVM to make the model robust. This part will discuss how these functions made the model robust and what are the features of these loss functions.

Authors proposed a novel approach to add robustness to their algorithm by making some changes in the hinge loss function. The hinge loss function is convex, unbounded and its results get adversely affected because of the presence of outliers in it. Therefore, the researchers have proposed a truncated hinge loss which is non-convex and can also handle outliers in the dataset [115]. This made the optimization problem, non-convex minimization problem. They used a non-convex optimization approach, Difference Convex (DC) to solve non-convex problems via a sequence of convex subproblems.

Similarly, another robust loss function, ramp loss function, was also added to SVM/SVR to make the optimization problem robust. These works are described below: Authors in 2008 also used the same technique of changing a non-convex function to the convex one using CCCP, but the non-convex function they used was ramp loss function [106]. The ramp loss function is insensitive to outliers. To solve the primal optimization, they used the same Newton-type algorithm and compared their results with existing robust algorithms using classification datasets. They got this idea of training SVM in the primal from Chapelle [11].

Huang et al. [35] also worked in the same direction as above mentioned. They also made use of Ramp loss function [22] to make the SVM more robust than previously existing methods but with a new addition of l_1 norm penalty to the optimization problem. The properties of the Ramp loss function is such that it is a non-convex smooth loss function and at the same time insensitive to outliers. They paired ramp loss function with $l_1$penalty to induce sparsity and named their method as Ramp-LPSVM (where LP is for linear programming). They found that their algorithm is more robust than standard Hinge loss function and ramp-SVM [106].

In 2011, researchers came with a new formulation by taking the sum of medians of hinge loss from two different classes and considered it as the total penalty function [61] . This approach gets some inspiration from 'Least Median Regression'. They used 'Rank and convex' algorithm to optimize their approach. It was observed that it is faster than classical SVM and earlier described ramp loss function which also worked in the direction of robustness to outliers. It was also observed that their approach is more rapid than the concave–convex procedure and was also adaptive to different levels of noise.

Next part discusses the use of another loss function with SVM, Pinball loss function. We have seen the hinge loss used with SVM in the classification problems. Although SVM with hinge loss works efficiently but hinge loss is not robust to outliers. Therefore, it was required to either change or modify the loss function to make it robust to outliers. Pinball loss function was initially used for regression problems. In 2014, pinball loss function was used for the first time with SVM classifier [36]. Authors corresponding to this work named it as pin-SVM. They proposed the given below optimization problem [36]:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} L_\tau(1 - y_i(w^T x_i + b)) \tag{26}$$

where L_τ is the pinball loss function and it is given as [36]:

$$L_\tau(u) = \begin{cases} u, & u \geq 0 \\ -\tau u, & u < 0 \end{cases} \tag{27}$$

This work was also extended to nonlinear classification and the optimization problem proposed was [36]:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} L_\tau(1 - y_i(w^T\phi(x_i) + b)) \tag{28}$$

The difference between hinge loss and pinball loss function is that the penalty was given to the correctly classified points by the pinball loss function. While, both the loss functions had similar computational complexity, pinball is less sensitive to outliers, more stable for re-sampling than hinge loss function [36]. This work was also extended further by using truncated pinball loss function with SVM as proposed by Shen et al. [85]. Now a question arises at this point: What was the need of truncated pinball loss function? Pinball loss function albeit robust to outliers, but it completely lost its sparseness with SVM, so truncated pinball loss function was introduced so that the classifier can be both robust to outliers as well as much sparser than earlier [85]. The resultant non-convex optimization problem was solved using CCCP. Truncating the loss functions was not a new concept in machine learning. Truncated least square loss [105] (for regression problems) and truncated logistic loss [72] (for classification problems) also contributed to robust statistics.

The use of robust loss function was also extended to regression problems as well. Authors in 2017 also proposed a robust regression algorithm for non-convex loss functions which was based on Laplace Kernel-induced loss function [132]. As LK-loss function is non-convex, therefore, DC programming was applied first. The advantage of considering LK-loss is its boundedness. In this paper, they proposed a new usage of LK-loss function by decomposing it into two formulations: one is regression formulation which was introduced as DC programming, (LKRE) and the other one was obtained by weighting it with a suitable parameter and named it as mixed loss function MLKRE where RE was used for regression model. Researchers gave a more robust formulation of L_p-norm based least squares SVR which gives the non-convex optimization problem to solve [138]. It provided a robust feature selection with faster and robust SVR compared to L_p-norm SVR and SVR method. Although the technique proposed was effective, it was slower than L_1-norm SVR and LS-SVR. Another robust SVR approach was proposed using least absolute deviation and to solve the optimization problem corresponding to it, and they used split Bregman iteration [12].

In 2017, another robust SVM for multi-class classification was proposed (multi-class SVM can be referred from Angulo et al. [2]. This method was based on ramp loss K-support vector classification-regression [3]. This method was solved further using CCCP procedure as the final optimization problem after ramp loss function was non-differentiable and non-convex.

Recently, the application of pinball SVM was extended to SVM+ and named it as PINSVM+ [147]. SVM+ and SVM have a difference that SVM+ considers additional information which is hidden in the training samples. This made the model robust to outliers and also more stable for re-sampling. Again, both the PINSVM+ and SVM+ had the same computational complexity. The optimization problem corresponding to PINSVM+ can be seen from Zhu et al. [147].

2.2.4 Transductive support vector machine

Ramp loss function was also used to make transductive support vector machine (TSVM) fast (than other transductive learning algorithms) and robust by Cevikalp and Franc [10]. This method can also be applied to large-scale data. Let there is a set of labeled training samples given by

$$L = \{(x_1, y_1), (x_2, y_2), \ldots, (x_L, y_L)\},\ x \in R^d,$$

$y \in \{-1, +1\}$ and another set of training samples which are not labeled (unlabeled) represented by

$$N = \{x_{L+1}, x_{L+2}, \ldots, x_{L+N}\}.$$

To find the best hyperplane, characterized by $H = (w, b)$ where w is the weight vector, and b is the bias term corresponding to that hyperplane. New labels are to be assigned on the basis this hyperplane and the equation is given by

$$f_H(x) = w^T x + b. \tag{29}$$

Using SVM, it is required to find the hyperplane which separates the classes with the maximum margin to the hyperplane, TSVM adds one more requirement that the unlabeled samples should be as far as possible away from the margin. Therefore, the optimization problem which is solved to achieve the above requirements is given below

$$\begin{aligned} \underset{w,b}{\text{argmin}}\ & \frac{1}{2}\|w\|^2 + P\sum_{i=1}^{L}\xi_i + P^*\sum_{i=L+1}^{L+N}\xi_i \\ \text{s.t. } & y_i(w^T x_i + b) \geq 1 - x_i, \quad i = 1, \ldots, L \\ & |w^T x_i + b| \geq 1 - \xi_i \quad i = L+1, \ldots, L+N. \end{aligned} \tag{30}$$

Now, it was required to introduce ramp loss function in the above equation instead of the hinge loss function. It was known that the ramp loss function could be written as the difference of two hinge loss function (as earlier described). Therefore, they replaced the symmetric hinge loss with symmetric ramp loss (for unlabeled points) as shown:

$$SR_\rho(t) = R_\rho(t) + R_\rho(-t). \tag{31}$$

where $R_\rho(t) = \min(1 - \rho, \max(0, 1 - t))$ was the ramp loss function used by Cevikalp and Franc [10]. Here

$-1 \leq \rho \leq 0$ was set according to the user. They used (31) to formulate a robust TSVM as shown below:

$$\begin{aligned} \underset{w,b}{\text{argmin}} \ & \frac{1}{2}\|w\|^2 + P\sum_{i=1}^{L} R_\rho(y_i(w^T x_i + b)) \\ & + P^* \sum_{i=L+1}^{L+N} SR_\rho(w^T x_i + b) \\ \text{s.t.} \ & \frac{1}{N}\sum_{i=L+1}^{L+N}(w^T x_i + b) = \frac{1}{L}\sum_{i=1}^{l} y_i. \end{aligned} \tag{32}$$

As the above problem is non-convex, they made use of the convex–concave procedure to solve the problem. Advantages of proposing this method are improved robustness of TSVM, and also it is much faster than existing methods. TSVM was again considered with margin distribution in Li et al. [52].

In 2017, Xu et al. [121] rescaled the loss function by making use of correntropy [60] and named it as a rescaled Hinge loss. Rescaled hinge loss overcomes the limitation of unboundedness and causes the loss function non-convex and bounded. The equation of the rescaled hinge loss function after applying correntropy is given by:

$$l_c(z) = \beta\left[1 - exp\left(-\frac{(1-z)^2}{2\sigma^2}\right)\right] \tag{33}$$

which can also be written as:

$$l_c(z) = \beta\left[1 - \exp(-\eta l_{\text{ls}}(z))\right] \tag{34}$$

where $\eta = \frac{1}{(2\sigma^2)} \geq 0$ is a scaling constant and $\beta = \frac{1}{(1-exp(-\eta))}$ The rescaled hinge loss function is given by:

$$l_{\text{rhinge}}(z) = \beta\left[1 - \exp(-\eta l_{\text{hinge}}(z))\right] \tag{35}$$

After rescaling, they made use of Half Quadratic Optimization method to get the optimal hyperplane from the changed loss function. In this work, authors also found that their approach is better than the previously existing methods and also proved the better sparseness of their method experimentally.

Another novel SVM formulation with L_t loss was proposed by Yang and Dong [130] which was more robust to outliers as compared to existing related approaches. To solve their non-convex objective function, they made use of DC (Difference of convex functions) algorithm [47, 95] and [131] and this function converged finitely concerning the problem. They also showed that their approach has sparsity and also follows Bayes rule. This method was compared with the existing SVM methods with pinball loss (PINSVM methods).

Query 4: How to choose the best loss function for a model?

As the surrogate loss functions were large in number, another problem was to best find the surrogate loss function for the classification problem. To solve this problem, a method of importance reweighting was proposed by Liu and Tao [59] according to which any kind of surrogate loss function can be used with a classification problem with label noise. It also solved the problem of obtaining the noise rate. This method depends on how accurately the conditional distribution $P_{D_\rho}(\hat{Y}/X)$ can be estimated where D_ρ is the distribution of corrupted variables $(X, \hat{Y})$. Both the efficiency and the robustness of the proposed approach was demonstrated experimentally with synthetic and real-world datasets. This part of the paper has discussed the use of different loss functions with SVM which can be seen in Fig. 4.

2.2.5 Generalized eigenvalue proximal SVM(GEPSVM)

Another variant of SVM which is next to be discussed is GEPSVM. This is the variant of SVM which was originated from proximal support vector machines(PSVMs) [62]. PSVMs were based on the concept of getting two parallel planes which are close to the datasets of the respective classes but as far as possible from each other. In PSVM, the final hyperplane was obtained from the midway of these two parallel planes. In contrast, GEPSVMs were having no restriction of parallel planes and the non-parallel planes were obtained by solving a pair of Generalized Eigenvalue problems [63]. Since its evolution, researchers are having a keen interest in developing various variants of it. This is because of GEPSVMs, Twin Support Vector Machines (TWSVMs) were introduced, and this model has helped a lot in the growth of machine learning. This will be discussed in the next subsection. GEPSVM has two main variants for classification: one is regularized GEPSVM [29] and the other one is improved GEPSVM given by Shao et al. [83]. In both the approaches, it was shown that these new variants of GEPSVM are better than the original one regarding computational time. Although robustness toward outliers was not discussed in these variants but, later in 2015, another extension of GEPSVM was proposed, which focused on outlier sensitivity of the model. In this approach, instead of considering two non-parallel planes, only one plane was considered which was proximal to the target and as far as possible from the outliers [26]. A similar approach was also proposed in 2016 which used L_1 norm with non-parallel proximal SVM [50]. The proposed technique had three significant advantages over GEPSVM, and these are listed below[50]:

- There was no need of solving two generalized eigenvalue problems like GEPSVM.
- L_1 norm made the problem more robust than GEPSVM.

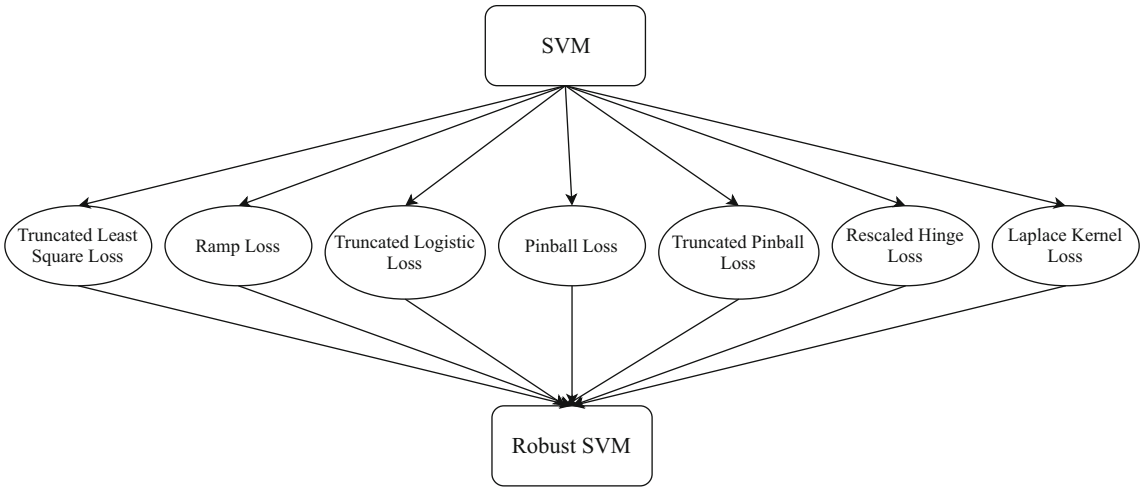

Fig. 4 Various loss functions used with SVM

– No parameter regularization was required in the proposed approach.

This can be considered as the improved version of GEPSVM as it mitigates the problems involved in GEPSVM.

Similarly, one more approach was proposed to deal with the sensitive nature of GEPSVM for the training set $T = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ where $x_i \in \mathcal{R}^n$ is the input and $y_i \in \{-1, 1\}$ is the output for $i = 1, \ldots, m$. They denoted class 1 by matrix $X_1 \in \mathcal{R}^{n \times m_1}$ having m_1 inputs and class 2 by matrix $X_2 \in \mathcal{R}^{n \times m_2}$ having m_2 inputs. This method retained the standard GEPSVM phenomena by considering non-parallel proximal SVM with $L_p(p \geq 0)$ norm regularization [90](LpNPSVM). The equation for LpNPSVM for $\bar{w_1} = (w_1, b_1)^T, \bar{w_2} = (w_2, b_2)^T, \bar{X_1} = (X_1, e_1)^T \in R^{n+1 \times m_1}, \bar{X_2} = (X_2, e_2)^T \in R^{n+1 \times m_2}$ is formulated as [90]:

$$\min_{\bar{w_1} \neq 0} \frac{\|\bar{w_1}^T \bar{X_1}\|_1 + \delta \|\bar{w_1}\|_p^p}{\|\bar{w_1}^T \bar{X_2}\|_1} \tag{36}$$

where $\delta > 0$ is the trade-off parameter in Tikhonov regularization terms. The important point which should be noted here about the proposed algorithm is its guaranteed convergence for $0 < p \leq 2$ but for $p > 2$, the method does not guarantee, therefore, this can be interesting to work in future. They have experimentally proved that the proposed approach is more robust than GEPSVM.

2.2.6 Twin support vector machine

GEPSVM leads to relevant research of introducing Twin Support Vector machines(TWSVM) to the world of machine learning, which was not only faster than the widely used model SVM but also showed good generalization. TWSVM was initially proposed for two-class classifier [43]. The algorithm defined two hyperplanes, one for each class. The algorithm classified points, according to which hyperplane a given point is closest to [43]. Following the similar notation as mentioned above while describing GEPSVM, TWSVM1(equation corresponding to first hyperplane) is defined as [43]:

$$\begin{aligned} &\min_{w_1, b_1} \frac{1}{2}(X_1 w_1 + e_1 b_1)^T (X_1 w_1 + e_1 b_1) + c_1 e_2^T q \\ &\text{subject to} - (X_2 w_1 + e_2 b_1) + q \geq e_2, q \geq 0 \end{aligned} \tag{37}$$

Similarly, TWSVM2 is defined as [43]:

$$\begin{aligned} &\min_{w_2, b_2} \frac{1}{2}(X_2 w_2 + e_2 b_2)^T (X_2 w_2 + e_2 b_2) + c_2 e_1^T q \\ &\text{subject to} - (X_1 w_2 + e_1 b_2) + q \geq e_1, q \geq 0 \end{aligned} \tag{38}$$

Here q is the term used to indicate slackness while classifying the points. Minimization of first term in both the above equations indicate the closeness of hyperplanes to points of one class and the constraints indicate that the hyperplane should be at least a distance of 1 from the points of other class. In the above Eqs. (37) and (38), e_1 and e_2 are the vectors of 1s with appropriate dimensions.

TWSVM has a significant advantage over the standard SVM, and that is the less computational time taken by TWSVM. Khemchandani et al. [43] have also proved it mathematically that TWSVM is four times faster than SVM. The effects of outliers or noise were not described in this work but because of its low computational time, various variants of TWSVM were proposed in which effects of noise or outliers were also discussed.

In 2008, the least square variant of TWSVM was proposed by Kumar and Gopal [45]. The proposal of least square TWSVM (LS-TWSVM) had the same reason as LS-SVM. The introduction of the squared error term in the optimization problem changed the inequality to equality

constraints and hence the quadratic problem changed accordingly. This can be viewed from the equation below of LS-TWSVM 1 [45]:

$$\min_{w_1,b_1} \frac{1}{2}(X_1w_1 + e_1b_1)^T(X_1w_1 + e_1b_1) + \frac{c_1}{2}q^Tq \quad \text{subject to} - (X_2w_1 + e_2b_1) + q = e_2,\ q \geq 0 \tag{39}$$

which can be further written as [45]:

$$\min_{w_1,b_1} \frac{1}{2}\|X_1w_1 + e_1b_1\|^2 + \frac{c_1}{2}\|X_2w_1 + e_2b_1 + e_2\|^2 \tag{40}$$

The above equation (40) was used further to get the values of w_1 and b_1 with c_1 as the regularization parameters. In this work, instead of solving two dual problems, two primal problems were solved using PSVM idea. This method was applied to both linear and nonlinear classifiers and it was experimentally shown that the method has higher generalization than PSVM. Although, the proposed approach had some significant advantages, but this approach was also sensitive to outliers [14]. To overcome this limitation, another variant of TWSVM was proposed, and that is weighted LS-TWSVM [14]. This was the weighted version of LS-TWSVM according to which weights were assigned to the error variables to reduce the impact of noise. The method was inspired by weighted LS-SVM. After introducing weights, the equation (39) was changed in the following manner [14]:

$$\min_{w_1,b_1} \frac{1}{2}(X_1w_1 + e_1b_1)^T(X_1w_1 + e_1b_1) + \frac{c_1}{2}\sum_{i=1}^{m_1} W_iq^Tq \quad \text{subject to} - (X_2w_1 + e_2b_1) + q = e_2,\ q \geq 0 \tag{41}$$

Similarly, weighted TWSVM2 was defined as:

$$\min_{w_2,b_2} \frac{1}{2}(X_2w_2 + e_2b_2)^T(X_2w_2 + e_2b_2) + \frac{c_2}{2}\sum_{i=1}^{m_2} W_iq^Tq \quad \text{subject to} - (X_1w_2 + e_1b_2) + q = e_1,\ q \geq 0 \tag{42}$$

Weights W_i were calculated using the following formula:

$$W_i = \begin{cases} 1, & \text{if } |q_i/\hat{s}| \leq J_1 \\ \dfrac{J_2 - |q_i/\hat{s}|}{J_2 - J_1}, & \text{if } J_1 \leq |q_i/\hat{s}| \leq J_2 \\ 10^{-4}, & \text{otherwise} \end{cases}$$

where $\hat{s} = \frac{IQR}{2\times 0.6745}$ or $\hat{s} = 1.483\text{MAD}(x_i)$. Here $\hat{s}$ stands for the extent of the estimated error distribution from a Gaussian distribution [14]. IQR was the interquartile range, and MAD is the mean absolute deviation with J_1, J_2 as 2.5 and 3 respectively. The approach was proposed for both linear as well as nonlinear cases. It has two significant advantages: one is robustness toward the noise, and the other one is computationally stable. Besides some benefits, its weight setting method can further be improved so that the technique can be extended to large-scale datasets [14]. Certain other methods were also proposed to assign different weights to the samples using the penalty parameter [126]. Many applications were also given by TWSVM [42, 44, 67, 100, 128] etc.

TWSVM was also extended for regression problems as well. Both the models had some major differences [75] except one similarity of getting two non-parallel planes around the data points. The equation for two hyperplanes proposed for twin support vector regression (TSVR) was:

$$\min \frac{1}{2}(Y - e\epsilon_1 - (X_1w_1 + eb_1))^T (Y - e\epsilon_1 - (X_1w_1 + eb_1)) + c_1e^Tq \quad \text{such that } Y - (X_1w_1 + eb_1) \geq e\epsilon_1 - q,\ q \geq 0, \tag{43}$$

and

$$\min \frac{1}{2}(Y + e\epsilon_2 - (X_2w_2 + eb_2))^T (Y + e\epsilon_2 - (X_2w_2 + eb_2)) + c_2e^Tq' \quad \text{such that } (X_2w_2 + eb_2) - Y \geq e\epsilon_2 - q',\ q' \geq 0 \tag{44}$$

where $C_1, C_2 \geq 0, \epsilon_1, \epsilon_2 \geq 0$ are the parameters used and q, q' are the slack vectors. The above proposed regression function was tested with different types of Gaussian noise according to which training samples were infected with zero means and different variances (Readers can see those equations from Peng [75]. It was observed that the proposed approach performed better than SVR and Least square regression problems [122].

Now, the question arises, Is TSVR robust to outliers? The answer to this question is *no*, and the reason is squared loss function which has the problem of sensitivity toward outliers [46]. It was also observed that the solution obtained from TSVR was not sparse, so there was a need to design a robust and sparse TSVR. This leads to the formulation of robust and sparse TSVR which comprised of L_1 loss function and L_1 regularizer [15]. The resulting problem was a linear problem which was solved by using the dual formulation of it. This function was further improved by following the same steps of introducing regularization first and then deriving the linear programming problem which was solved using Newton algorithm with Armijo step-size to resolve the corresponding exact exterior penalty problem [16]. This was an important approach proposed in the field of TWSVM for regression which leads to the discovery of parametric-margin SVM [76] and v-TWSVM [73](which was further improved in 2014 by introducing an extra regularization term in the objective function which indeed enhanced the generalization [124]. Like TWSVM, TSVR was also extended to its weighted versions which worked

in the direction of robustifying the regressors toward noise and outliers. Influenced from the fact that samples at different positions impacts differently at the bound function, weighted TSVR was proposed. In this regressor, different samples had different penalties and therefore it was very helpful in avoiding over-fitting problems. The equations corresponding to both the non-parallel planes with penalty coefficients σ_1, σ_2 are given by Xu and Wang [125]:

$$\begin{aligned} &\min_{w_1,b_1,q,q'} \ \frac{1}{2}\|Y - e\epsilon_1 - (X_1 w_1 + eb_1)\|^2 \\ &\quad + c_1(e^T q + \sigma_1 e^T q') \\ &\text{such that } Y - (X_1 w_1 + eb_1) \geq e\epsilon_1 - q - q', \\ &\quad 0e \leq q \leq \epsilon_1 e, \ q' \geq 0e, \end{aligned} \tag{45}$$

and

$$\begin{aligned} &\min_{w_2,b_2,\eta,\eta^*} \ \frac{1}{2}\|Y - e\epsilon_2 - (X_1 w_2 + eb_2)\|^2 + c_2(e^T \eta + \sigma_2 e^T \eta^*), \\ &\text{such that } (X_1 w_2 + eb_2) - Y \geq e\epsilon_2 - \eta - \eta^*, \\ &\quad 0e \leq \eta \leq \epsilon_2 e, \ \eta^* \geq 0e, \end{aligned} \tag{46}$$

where q, q', η, η^* are the slack vectors. Here σ_1, σ_2 were to be chosen apriori. The method was proposed for both linear and nonlinear weighted TSVR. As the penalties were assigned to the data points, outliers were assigned different penalties due to which the algorithm can deal with outliers but Does the introduction of penalty parameters increased the computational time? The answer is not much. But there was one disadvantage of the proposed approach. Just like TSVR, weighted TSVR also lost its sparsity [125]. Various researchers have tried to add robustness to least square TSVR [34, 51] to remove its limitation of lost sparsity. This work was also extended toward the discovery of least square variants of TSVR [144] and primal TSVR given by Peng [74]. TSVR was performed in various ways like proposed by Zhong et al. [145] and Shao et al. [84] which were further used to make the regressor robust. (We are not discussing these articles in detail as they are not concerned with robust statistics)

As the loss function used with SVM is generally hinge loss. TWSVM also performed the experiments considering hinge loss and SSE as we discussed earlier. Now the question arises: Can the change in loss function affects the robustness of TWSVM or TSVR?

The answer to this question is *yes*. Making use of robust loss functions can make the overall model robust to outliers or noise. TWSVM or TSVR were also made robust using various loss functions which are roust to noise and outliers. These include ramp loss function, pinball loss function, truncated pinball loss function, Huber loss, etc.

TWSVM/TSVR with different loss functions

Although we have discussed the least square variants of TWSVM/TSVR, now this part is all about the different robust loss functions used to make this classifier/regressor robust. The idea of using different loss functions to the non-parallel support vector machines were given by Mehrkanoon et al. [64] which was proposed for classifiers only. In this work, the objective function was comprised of a regularization term, a misclassification loss and a scatter loss function. It was initially proposed for GEPSVMs, TWSVM and also the least square variants of it. The loss functions they considered were hinge loss, least squares loss and pinball loss function [64].

Now, why hinge loss function was replaced and still being replaced by some other loss function in case of SVM or its variants like TWSVM? (This part will only discuss different loss functions on TWSVM/TSVR)

Because of the sensitivity to noise and instability for resampling [127], hinge loss was replaced. In 2017, pinball loss function was considered with TWSVM which dealt with the quantile distance, and it is also less sensitive to noise. Like an ordinary TWSVM problem, pin-TWSVM also found two hyperplanes [127]:

$$f_1(x) = w_1^T x + b_1 = 0, \tag{47}$$

and

$$f_2(x) = w_2^T x + b_2 = 0. \tag{48}$$

The above two equations are separated for positive and negative class respectively. The pinball loss function considered for pin-TWSVM is shown below [127]:

$$L_\tau(x, y, f(x)) = \begin{cases} 0 - yf(x), & 0 - yf(x) \geq 0 \\ -\tau(0 - yf(x)), & 0 - yf(x) < 0 \end{cases}$$

This loss function when used with TWSVM, it changed the equations of two hyperplanes in the following manner [127]:

$$\min_{w_1,b_1} \frac{v_1}{m_2} \sum_{j=1}^{m_2} \frac{|w_1^T x_j + b_1|}{\|w_1\|^2} + \frac{C_1}{m_1} \sum_{i=1}^{m_1} L_{\tau_1}(x_i^1, y_i, f_1(x_i^1)) \tag{49}$$

and

$$\min_{w_2,b_2} \frac{v_2}{m_1} \sum_{j=1}^{m_1} \frac{|w_2^T x_j + b_2|}{\|w_2\|^2} + \frac{C_2}{m_2} \sum_{i=1}^{m_2} L_{\tau_2}(x_i^2, y_i, f_2(x_i^2)) \tag{50}$$

Here, $C_1, C_2 > 0$ and $v_1, v_2 > 0$ were the parameters used. In this work, the use of pinball loss function instead of hinge loss function made the objective function insensitive to noise. It also added within-class scatter minimization and between-class distance maximization [127].

Similarly, if we consider regression problems in which ε-insensitive loss function is generally used, can it

effectively handles Gaussian noise in the dataset? The answer is *no*. To deal with this limitation of ε-insensitive loss function, researchers proposed to use Huber loss function in place of ε-insensitive loss function as the Huber loss function can handle a variety of noise and outliers in the dataset. This can also help in providing good generalization performance. The idea was applied in 2017 by Niu et al. [69].

Recently, ramp loss function was also used to add robustness to TSVR [94]. It was observed that the use of ramp loss function not only adds robustness but also provides an excellent sparseness. For the two non-parallel hyperplanes given by (41) and (42), ε_1-insensitive down-bound function $(f_1(x)+\varepsilon_1)$ and ε_2-insensitive up-bound function $(f_1(x)-\varepsilon_2)$ provided the below mentioned two primal problems [94]):

$$\min_{w_1,b_1} \frac{1}{2}w_1^T w_1 + C_1\sum_{i=1}^{N} R_{\varepsilon i}(|z_{1i}|) + C_3\sum_{i=1}^{N} R_{0i}(|z_{1i}|) \quad z_{1i} = f_1(x_i) + \varepsilon_1 - y_i \tag{51}$$

and

$$\min_{w_2,b_2} \frac{1}{2}w_2^T w_2 + C_2\sum_{i=1}^{N} R_{\varepsilon i}(|z_{2i}|) + C_4\sum_{i=1}^{N} R_{0i}(-z_{2i}) \quad z_{2i} = f_1(x_i) - \varepsilon_2 - y_i \tag{52}$$

with training dataset with N data points. Both the ramp ε-insensitive loss functions are given below [94]):

$$R_{\varepsilon i}(|z|) = \begin{cases} 0, & |z| < \varepsilon, \\ |z| - \varepsilon, & \varepsilon \le |z| \le v_i, \\ v_i - \varepsilon, & |z| > v_i \end{cases} \tag{53}$$

and

$$R_{0i}(z) = \begin{cases} 0, & z < 0, \\ z, & 0 \le z \le v_i, \\ v_i, & z > v_i \end{cases} \tag{54}$$

Now, how these two ramp loss functions, (53) and (54), are working against outliers?

From (53), it can be observed that the penalty for each training set is constrained to be no more than $v_i - \varepsilon$, which reduces the impact of outliers [94]). Similarly, in (54), value of $R_{0i}(z)$ builds a flat surface in the third condition to avoid the effects of outlier points [94]). As the ramp loss function makes the overall objective function a non-convex problem, the authors used CCCP to solve the function. This paper not only focused on robustness but also proved the sparseness and generalization ability of their proposed approach. This approach also provided the significant scalability to the large-scale regression problems.

Ramp loss function was also used with Twin Support Vector Clustering (TWSVC) (as proposed by Wang et al. [111] because of its boundedness [112]. It was observed that the use of ramp loss function with TWSVC also found intrinsic clusters which were difficult with other clustering methods(proposed approach was compared with Ye et al. [137].

2.2.7 One-class SVM(OCSVM)

As we have discussed a lot about SVM and its advantages. The limitations of sensitivity toward outliers and lack of sparsity have also been discussed but how will SVM respond toward class imbalance. As SVM was mainly used for two-class classification, it was observed that the performance of SVM is not that good in the case of class imbalance [120]. To deal with the class imbalance, one-class classification played a significant role. In this classification, only one class is considered, and the model was trained on the basis of that class. During the testing phase, the test set was evaluated on the basis of the model trained and gave the result whether the data points of the test set belong to that class or not [24]. There are various one-class classification methods developed so far like one-class support vector machine(OCSVM) [87] and support vector data description(SVDD) [98]. As this work is related to SVM and robust statistics, we will discuss here only those methods of one-class classification in which both SVM and robust statistics are involved.

Outliers may exist in the training samples which can affect the performance adversely. It was observed that the outliers also affects the performance of OCSVM[139]. To deal with the problem of outliers, various variants of OCSVM were proposed. Following the footsteps of SVM and its variants, least square variant of OCSVM was also developed [18] keeping in mind the same advantages as in the case of LS-SVM, LS-TWSVM, etc.

Like SVM, a weighted version of OCSVM was also proposed which was used to assign lower weight value to the sample with outlier so that it may lay a somewhat lesser impact on the model trained. There were various weighted versions of OCSVM were proposed, but initially, the idea was proposed in the year 2009[8]. Like SVM and TWSVM, weights assigned indicated the importance given to a sample while training the model. Therefore, there were many methods proposed to assign the weights to the training samples. In 2014, weights were assigned to the examples by their distance to the sample center in the feature space[139]. For the dataset $X = [x_1, \ldots, x_N]^T \in \mathcal{R}^{N\times M}$, the optimization problem corresponding to OCSVM is given by [139]:

$$\min_{w\in F,\xi\in\mathcal{R}^N,\rho\in\mathcal{R}} \frac{1}{2}\|w\|^2+\frac{1}{Nv}\sum_{i=1}^{N}\xi_i-\rho \quad \text{subject to } w.\phi(x_i)\geq\rho-\xi_i,\ \xi_i\geq 0 \tag{55}$$

where N are the number of data points, v is a regularization parameter, ξ_i is the slack variable for point x_i that allows it to locate outside the decision boundary [139]. The equation of the decision boundary given by [139]:

$$f(x)=w.\phi(x)-\rho \tag{56}$$

where ϕ is a mapping function with $x\in\mathcal{R}^M$. This equation (55) was changed according to the weighted version of OCSVM which includes an extra parameter W_i denoting weights corresponding to every training sample [139].

$$\min_{w\in F,\xi\in\mathcal{R}^N,\rho\in\mathcal{R}} \frac{1}{2}\|w\|^2+\frac{1}{Nv}\sum_{i=1}^{N}W_i\xi_i-\rho \quad \text{subject to } w.\phi(x_i)\geq\rho-\xi_i,\ \xi_i\geq 0 \tag{57}$$

In the above equation, W_i are the penalty parameters attached to every training point and the value of W_i is given by [139]:

$$\hat{W}_i=\|x_i-C\|^2 \tag{58}$$

and

$$W_i=W_{i,\max}/\hat{W}_i \tag{59}$$

here $W_{i,\max}$ is the maximal distance and C denotes the center coordinate of the dataset [139]. Now, the question arises: How is the insertion of W_i makes the model robust toward outliers? As the values of W_i for outliers is lesser than the rest of the data points, this leads the outliers to lie outside the decision boundary, and the model would be trained without outliers. In this way, weight values are making the model robust. In this technique, the limitation of assigning weights during each iteration was removed by Yang et al. [129] in which adaptive weighting technique was used and made the model robust to outliers.

Similarly, there were many other ways proposed to identify the weight vector in the weighted OCSVM to make the model robust and to minimize the complexity of finding weights like Zhu et al. [146] which used K-nearest neighbor to find weights of the training samples.

The idea of fuzzy membership was also applied to one-class classification problem. In 2008, fuzzy membership values were assigned to the data points, and the problem was reformulated to one-class SVM (OCSVM) where different data points made different contributions to the decision surface [32].

OCSVM was also used in some applications from a robustness point of view like patient classification problem which was considered as an outlier detection problem[66]. This work was based on the facial expressions of the patients corresponding to the patterns of fMRI(brain activation) response. If the response was sad, it was classified as an outlier in contrast to patterns of healthy control subjects [66]. Authors made use of OCSVM but there was one disadvantage of the OCSVM approach, it did not provide specificity to the specific disorders and also it can not be applied to the patient groups of larger samples.

Different Loss functions with OCSVM

Like SVM, various loss functions were proposed to suppress the effects of outliers on the model. Although with SVM, such researches are quite extensive in number but with OCSVM, such works are taking its pace to make the OCSVM robust to outliers. In contrast to SVM, OCSVM does not have a rich literature in this part. As using OCSVM, the primary objective is to find an optimal hyperplane as shown in (56) using the optimization problem (55). What happens when different loss functions are used with (55)? Authors in 2016 attempted to make the one-class SVM (OCSVM) robust and used it for fault detection [118]. The primal optimization problem corresponding to OCSVM is given by [118].

$$\min_{w,\xi,\rho} J(w,\rho)=\frac{1}{2}\|\mathrm{w}\|^2-\rho+\frac{1}{vn}\sum_{i=1}^{N}\xi_i \quad \text{s.t. } \langle w,\varphi(x_i)\rangle\geq\rho-\xi_i,\ \xi_i\geq 0 \tag{60}$$

where x_i is used for training samples which are N in number, v is the trade-off parameter and ξ_i are slack variables. In this paper, authors have shown that outliers affect the OCSVM decision boundary by becoming support vectors. Therefore, the idea of their proposed method is to edit the training set so that outliers are not included in this set and hence these outliers can be prevented from becoming support vectors. They named their method as vnuOCSVM and its computational complexity is $O(n^3)$. In 2016, an experiment was performed by same authors in which OCSVM was used with ramp loss function to make the model robust to outliers and noise. The equation (60) can be written as [119]:

$$\min_{w,\rho}\frac{1}{2}\|w\|^2-\rho+\frac{1}{Nv}\sum_{i=1}^{N}H_\rho(z_i) \tag{61}$$

where $z_i=\langle w,\phi(x_i)\rangle$ and $H_\rho(z_i)$ represents the hinge loss function [119]. The above was changed in the following manner when ramp loss function was used with it[119]:

$$\min_{w,\rho} \frac{1}{2}\|w\|^2 - \rho + \frac{1}{Nv}\sum_{i=1}^{N} R_{\rho,r}(z_i) \tag{62}$$

where $R_{\rho,r}$ used was the ramp loss function, and it can be defined as [119]:

$$R_{\rho,r}(z_i) = \begin{cases} 0, & z_i \geq \rho, \\ \rho - z_i, & r\rho \leq z_i < \rho, \\ \rho - r\rho, & z_i \leq r\rho \end{cases} \tag{63}$$

As discussed earlier, ramp loss function has a term $0 < r < 1$. But what is the use of ramp loss function with OCSVM? Conventional SVM has hinge loss as the loss function which is sensitive to outliers. Similarly, OCSVM also considers hinge loss function in its conventional form which is unable to depress the effects of outliers. To make the influence of outliers bounded, ramp loss function was used here in place of the hinge loss function. This was experimentally proved with other classifiers as well that ramp loss function is more robust to outliers than hinge loss function.

Ramp loss function defined below [119] was again used to make the model robust:

$$R_{\rho,r}(z_i) = \begin{cases} 0, & z_i \geq \rho \\ \rho - z_i, & r\rho \leq z_i \leq \rho \\ \rho - r\rho, & z_i \leq r\rho \end{cases} \tag{64}$$

where $0 \leq r \leq 1$ but with a difference. It was observed that ramp loss function can be written as the difference of two hinge loss functions. Therefore, ramp loss function can be written as: $R_{\rho,r}(z_i) = H_{\rho}(z_i) - H_{r\rho}(z_i)$. This gives the equation as shown below:

$$\min_{w,\xi,\rho} J(w,\rho) = \frac{1}{2}\|w\|^2 - \rho + \frac{1}{vn}\sum_{i=1}^{n} H_{\rho}(z_i) - \frac{1}{vn}\sum_{i=1}^{n} H_{r\rho}(z_i). \tag{65}$$

This ramp loss function introduced here was used to identify and remove outliers [119]. The novelty of this work lies in making use of ramp loss function for OCSVM to make it robust to outliers.

Recently, another article was proposed which used rescaled hinge loss [121] function with OCSVM to make the model more robust to outliers [120]. The final optimization problem to solve OCSVM with rescaled hinge loss is given below [120]:

$$\min_{w,\rho} \frac{1}{2}\|w\|^2 - \rho + \frac{1}{Nv}\sum_{i=1}^{N} H_{rhinge}(z_i) \tag{66}$$

where $H_{\text{rhinge}}(z_i)$ is the rescaled hinge loss used and as defined earlier, rescaled hinge loss is [120]:

$$H_{\text{rhinge}}(z_i) = \beta[1 - \exp\{\eta H_{\text{hinge}}(z_i)\}] \tag{67}$$

where $\beta = \frac{1}{1-\exp\{-\eta\}}$. Now another question arises. Why was Hinge loss function rescaled? We have already given its answer earlier defining the advantages of rescaled hinge loss function over hinge loss function. To solve the robust one class SVM, they also used the half quadratic method. Ramp loss function was also used with OCSVM to get a robust and sparse classifier [99]. The obtained non-convex optimization problem was solved using CCCP.

Similarly, SVDD was also made robust to outliers and noise. In 2015, two types of SVDD methods were proposed in which first one was R-SVDD which was based on cutoff distance-based local density of each sample and the other one was εNR-SVDD which considered ε-insensitive loss function with negative samples [13]. It was observed that the proposed methods were superior to the other existing outlier detection methods of that time.

Although OCSVM was experimented with different loss functions to make it robust to outliers, still, other bounded loss functions can also be tried to improve its robustness further and also to reduce the complexity and computational time. This area of machine learning is an ongoing research field which is covering different application domains [27, 37, 70] and also an essential family of machine learning, deep learning [71, 80]. These work can be extended further from the robustness point of view.

Not only in the field of supervised machine learning, SVM has also extended its roots in the field of unsupervised machine learning as well. Support vector clustering was the method developed to cluster the similar data points based on the technique of SVM [6]. According to this technique, data points are mapped to a high-dimensional feature space using the Gaussian kernel. Later, the enclosing sphere is searched in the high-dimensional feature space, and when this sphere is mapped again back to the data space, it separates into different clusters of points [6]. Since this proposal, various robust support vector clustering methods were proposed [25, 86, 104, 110]. Like SVM, robust fuzzy clustering approach was also proposed to deal with outliers [40]. Similarly, one class clustering similar to one-class SVM was also introduced to make the clustering robust [30]. This technique made use of the hybrid of global and local search. An interesting paper on the survey of various outlier detection techniques in unsupervised learning was also proposed by Rakhe and Vaidya [78]. We do not include the details related to the contribution of SVM in unsupervised learning as it can lead to an altogether a different survey. Adversarial classification is also spreading its roots in various fields. It was also considered for robust statistics as the malicious adversaries may manipulate the data which can further affect the outcome of the automatic analysis. Adversarial data

manipulation was discussed many a time with many different classifier. SVM was amongst those classifier. Robust SVM was proposed against adversarial manipulation by Biggio et al. [9]. Authors of this work made use of simple kernel matrix correction and proposed their method 'Label noise robust SVM' [9]. This method has a limitation of a prior agreement on the possible degree of label contamination. In a recent work, randomized SVM was also proposed against generalized adversarial attacks under uncertainty [17].

Before concluding the above work, we have given a table (Table 1) to summarize the pros and cons of various extensions of SVM discussed above.

2.3 Applications of robust SVM

Since the proposal of robust SVM, it has been applied in various areas which can be an altogether different survey. This part of the paper discusses some of the recent applications of robust SVM, which can motivate the researchers to work in these directions. In 2015, the SVM was used to formulate a two-class classification problem for voice activity detection (VAD). In this work, the SVM model was trained against different levels of noises for speech and non-speech classification [82]. Similarly, in 2016, an SVM based method was proposed to remove impulsive noise from grayscale images. Authors in this work have used fuzzy filter based classification which classified all the test images as either noisy or non-noisy [79]. Robust SVM played a significant role in the field of medical diagnosis. In 2018, an important work was proposed to deal with Ischemic stroke. In this work, MRI image analysis was performed to detect the lesion tissue in the brain image. The authors corresponding to this work made use of Kernelized fuzzy c-means clustering with adaptive thresholding algorithm [103]. SVM was also used to detect the presence of noise in the speech signal. This work made use of cumulative short-time Fourier transform for the classification of noise [65]. As various types of noises are present in the signal, SVM for multi-class classification was used.

When any classifier is used in digital image processing, its primary challenge is to remove noise from the images. SVM was also combined with low-rank matrix decomposition(LRMD) to denoise the image [7]. SVM was merged with LRMD because it helps in the removal of various types of noises from the image simultaneously. It was observed that the proposed method could remove both noise and residual aliasing artifact from pMRI reconstructed noisy images [7].

The current applications of SVM include the detection of malicious Facebook posts [31] through intelligent systems. This uses SVM as the classifier to accurately classify the posts into malicious and non-malicious. Similarly, other recent applications of SVM include the improvement in a human detection system for security and military applications [28].

There are certain other applications in which SVM is ensembled with other models like the one proposed for

Table 1 Summary of variants of SVM

Variants	Pros	Cons
LS-SVM [38, 91]	Converts inequality constraints to equality constraints	Lost sparseness
	Easy to solve optimization problem	Lost robustness
Fuzzy SVM [53]	Extends the application horizon of SVM	How to adaptively determine a suitable model of fuzzy membership function
	More feasible in reducing the effects of noise than SVM	
Weighted SVM [133]	Provides higher classification rate than SVM	Weight assignment is a problem
	less sensitive to outliers	Computational complexity is higher
Transductive SVM [10]	Considers even unlabeled data points	Optimization problem is relatively harder to solve
	Can also be used for semi-supervised learning	
GEPSVM [63]	Computational time is low	Still outlier sensitive
	Solves two equations for non-parallel planes instead of one complex quadratic problem	
TWSVM [43, 75]	Four times faster than SVM	Needs to compute the inverse of matrices in standard TWSVM
	Can handle class imbalance problem	Standard TWSVM/TSVR are outlier sensitive
	Its variants nicely handles outlier sensitivity problem	Lack of sparseness
OCSVM [87]	Handles class imbalance	Outlier sensitive
	Train the classifier using only patterns belonging to target class	

traffic incident detection in Xiao [117]. Furthermore, the ensemble of SVM with kernel spherical K-means was also proposed and this model was used for acute sinusitis classification [81]. The ensemble model of SVM was also used in the field of image communication like the one proposed by Wang et al. [107]. This classifier has also been used with nature-inspired optimization algorithms like Takruri et al. [93] with particle swarm optimization (PSO), Tao et al. [97] with genetic algorithm (GA), Wang et al. [113] with improved chicken algorithm optimization and Rajalaxmi and Vidhya [77] with mutated salp swarm optimization etc. The hybrid of nature-inspired optimization algorithms have also contributed in optimizing SVM parameters [41, 101].

After this discussion on SVM and its variants based on robust statistics, it can be observed that SVM has contributed a significant part in the field of machine learning and there is a lot yet to be discovered in this domain. Fig. 5 gives a plot indicating the number of papers discussed year-wise.

3 Discussion and future scope

This paper described the use of robust statistics in SVM. In this paper, we have observed that there are various methods proposed to make the model robust toward outliers or noise in the dataset. This work also explained the multiple variants of SVM which were made robust. This paper also shows how some old techniques like weighted versions of LS-SVM are still being considered in multiple researches. From the theory, the reason for proposal of different variants of SVM can be analyzed. This work also motivates the researchers to work in new and less explored extensions of SVM like TWSVM, OCSVM, etc. Also, there are some variants which lack practical application background like TSVM. Some of the variants like OCSVM have relatively less amount of work in robust statistics which can be considered further. Therefore, from the above conclusion, future research may include:

- How to effectively add robustness in TWSVM while improving sparsity of it. As sparseness is the major concern in TWSVMs, it is essential to enhance sparseness at the same time while adding robustness.
- How to extend the use of robust OCSVM to applications like medical diagnosis etc. As it was observed from the discussion that OCSVM has comparatively lesser applications than the rest of the variants. Therefore, it is required to extend the use of OCSVM while adding robustness to it.
- As we have seen how the use of robust loss functions with different variants enhanced their robustness. Similarly, other robust loss functions can also be tried.
- TSVMs are difficult to implement because they handle labeling of unlabeled data points in addition to maximizing the margin. More efficient algorithms can be explored for this problem.

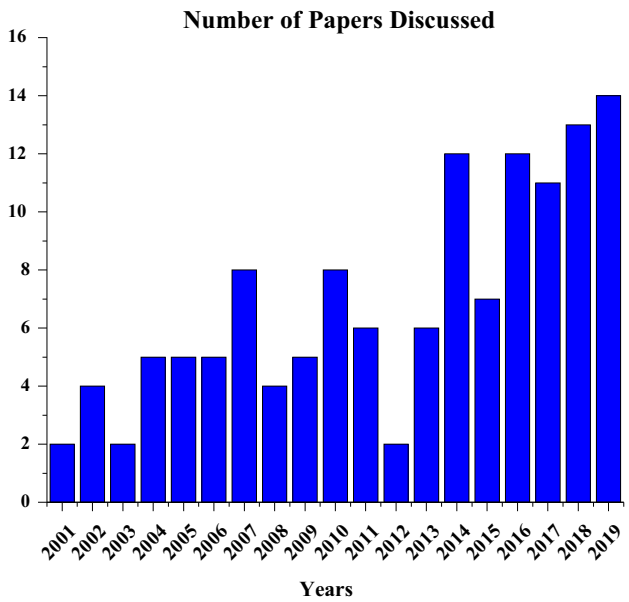

Fig. 5 Number of papers discussed

Acknowledgements The authors would like to thank the anonymous reviewers for their valuable comments that have resulted in the significant improvement of the paper. The first author would like to thank IIT BHU for providing the research fellowship.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. An W, Liang M (2013) Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises. Neurocomputing 110:101–110
2. Angulo C, Parra X, Catala A (2003) K-SVCR a support vector machine for multi-class classification. Neurocomputing 55(1–2):57–77
3. Bamakan SMH, Wang H, Shi Y (2017) Ramp loss k-support vector classification-regression; a robust and sparse multi-class approach to the intrusion detection problem. Knowl Based Syst 126:113–126
4. Barnett V, Lewis T (1974) Outliers in statistical data. Wiley, Hoboken
5. Batuwita R, Palade V (2010) Fsvm-cil: fuzzy support vector machines for class imbalance learning. IEEE Trans Fuzzy Syst 18(3):558–571
6. Ben-Hur A, Horn D, Siegelmann HT, Vapnik V (2001) Support vector clustering. J Mach Learn Res 2(Dec):125–137

7. Bhukra MJ, Sharma KK (2018) Rician noise reduction with SVM, IMRD and iterative bilateral filter in different type of medical images using digital image processing
8. Bicego M, Figueiredo MA (2009) Soft clustering using weighted one-class support vector machines. Pattern Recognit 42(1):27–32
9. Biggio B, Nelson B, Laskov P (2011) Support vector machines under adversarial label noise. In: Asian conference on machine learning, pp 97–112
10. Cevikalp H, Franc V (2017) Large-scale robust transductive support vector machines. Neurocomputing 235:199–209
11. Chapelle O (2007) Training a support vector machine in the primal. Neural Comput 19(5):1155–1178
12. Chen C, Li Y, Yan C, Guo J, Liu G (2017) Least absolute deviation-ased robust support vector regression. Knowl Based Syst 131:183–194
13. Chen G, Zhang X, Wang ZJ, Li F (2015) Robust support vector data description for outlier detection with noise or uncertain data. Knowl Based Syst 90:129–137
14. Chen J, Ji G (2010) Weighted least squares twin support vector machines for pattern classification. In: 2010 The 2nd international conference on computer and automation engineering (ICCAE), IEEE, vol 2, pp 242–246
15. Chen X, Yang J, Liang J, Ye Q (2010) Robust and sparse twin support vector regression via linear programming. In: 2010 Chinese conference on pattern recognition (CCPR), IEEE, pp 1–6
16. Chen X, Yang J, Chen L (2014) An improved robust and sparse twin support vector regression via linear programming. Soft Comput 18(12):2335–2348
17. Chen Y, Wang W, Zhang X (2018) Randomizing svm against adversarial attacks under uncertainty. In: Pacific-Asia conference on knowledge discovery and data mining, Springer, pp 556–568
18. Choi YS (2009) Least squares one-class support vector machine. Pattern Recognit Lett 30(13):1236–1240
19. Chuang CC (2007) Fuzzy weighted support vector regression with a fuzzy partition. IEEE Trans Syst Man Cybern Part B (Cybern) 37(3):630–640
20. Chuang CC, Lee ZJ (2011) Hybrid robust support vector machines for regression with outliers. Appl Soft Comput 11(1):64–72
21. Chuang CC, Su SF, Jeng JT, Hsiao CC (2002) Robust support vector regression networks for function approximation with outliers. IEEE Trans Neural Netw 13(6):1322–1330
22. Collobert R, Sinz F, Weston J, Bottou L (2006) Trading convexity for scalability. In: Proceedings of the 23rd international conference on Machine learning, ACM, pp 201–208
23. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
24. David M (2001) Tax one-class classification; concept-learning in the absence of counter-examples. ASCI dissertation series, 65
25. Du H, Zhao S, Zhang D, Wu J (2016) Novel clustering-based approach for local outlier detection. In: 2016 IEEE conference on computer communications workshops (INFOCOM WKSHPS), IEEE, pp 802–811
26. Dufrenois F, Noyer JC (2015) Generalized eigenvalue proximal support vector machines for outlier description. In: 2015 International joint conference on neural networks (IJCNN), IEEE, pp 1–9
27. Erfani SM, Rajasegarar S, Karunasekera S, Leckie C (2016) High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. Pattern Recognit 58:121–134
28. Goel V, Raj H, Muthigi K, Kumar SS, Prasad D, Nath V (2019) Development of human detection system for security and military applications. In: Proceedings of the 3rd international conference on microelectronics, computing and communication systems, Springer, pp 195–200
29. Guarracino MR, Cifarelli C, Seref O, Pardalos PM (2007) A classification method based on generalized eigenvalue problems. Optim Methods Softw 22(1):73–81
30. Gupta G, Ghosh J (2005) Robust one-class clustering using hybrid global and local search. In: Proceedings of the 22nd international conference on machine learning, ACM, pp 273–280
31. Gurumurthy S, Sushama C, Ramu M, Nikhitha KS (2019) Design and implementation of intelligent system to detect malicious facebook posts using support vector machine (SVM). In: Soft computing and medical bioinformatics, Springer, pp 17–24
32. Hao PY et al (2008) Fuzzy one-class support vector machines. Fuzzy Sets Syst 159(18):2317–2336
33. Heo G, Gader P (2009) Fuzzy svm for noisy data: A robust membership calculation method. In: IEEE international conference on fuzzy systems, 2009. FUZZ-IEEE 2009. IEEE, pp 431–436
34. Huang H, Wei X, Zhou Y (2016) A sparse method for least squares twin support vector regression. Neurocomputing 211:150–158
35. Huang X, Shi L, Suykens JA (2014) Ramp loss linear programming support vector machine. J Mach Learn Res 15(1):2185–2211
36. Huang X, Shi L, Suykens JA (2014) Support vector machine classifier with pinball loss. IEEE Trans Pattern Anal Mach Intell 36(5):984–997
37. Jeragh M, AlSulaimi M (2018) Combining auto encoders and one class support vectors machine for fraudulant credit card transactions detection. In: 2018 Second world conference on smart trends in systems, security and sustainability (WorldS4), IEEE, pp 178–184
38. Jiang J, Wu C, Song C et al (2006) Adaptive and iterative gene selection based on least squares support vector regression. J Inf Comput Sci 3(4):443–451
39. Jordaan EM, Smits GF (2004) Robust outlier detection using SVM regression. IEEE Int Jt Conf Neural Netw 3:2017–2022
40. Joshi A, Krishnapuram R (1998) Robust fuzzy clustering methods to support web mining. In: Proceedings of workshop in data mining and knowledge discovery, SIGMOD, Citeseer, pp 1–15
41. Kaya D (2019) Optimization of SVM parameters with hybrid CS-PSO algorithms for Parkinson's disease in labview environment. Parkinson's Disease 2019
42. Khemchandani R, Sharma S (2016) Robust least squares twin support vector machine for human activity recognition. Appl Soft Comput 47:33–46
43. Khemchandani R, Chandra S et al (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
44. Kumar MA, Gopal M (2008) Application of smoothing technique on twin support vector machines. Pattern Recognit Lett 29(13):1842–1848
45. Kumar MA, Gopal M (2009) Least squares twin support vector machines for pattern classification. Expert Syst Appl 36(4):7535–7543
46. Kwak N (2008) Principal component analysis based on l1-norm maximization. IEEE Trans Pattern Anal Mach Intell 30(9):1672–1680
47. Le HM, Le Thi HA, Nguyen MC (2015) Sparse semi-supervised support vector machines by DC programming and DCA. Neurocomputing 153:62–76

48. Le Thi Hoai A, Tao PD (1997) Solving a class of linearly constrained indefinite quadratic problems by dc algorithms. J Glob Optim 11(3):253–285
49. Lee G, Taur J, Tao C (2006) A robust fuzzy support vector machine for two-class pattern classification. Int J Fuzzy Syst 8(2):76–86
50. Li CN, Shao YH, Deng NY (2016) Robust l1-norm non-parallel proximal support vector machine. Optimization 65(1):169–183
51. Li Q, Li X, Ba W (2015) Sparse least squares support vector machine with l 0-norm in primal space. In: 2015 IEEE international conference on information and automation, IEEE, pp 2778–2783
52. Li Y, Wang Y, Bi C, Jiang X (2018) Revisiting transductive support vector machines with margin distribution embedding. Knowl Based Syst 152:200–214
53. Lin CF, Wang SD (2002) Fuzzy support vector machines. IEEE Trans Neural Netw 13(2):464–471
54. Lin Cf, Wang Sd (2005) Fuzzy support vector machines with automatic membership setting. Theory Appl Support Vector Mach 177:233–254
55. Lin CF et al (2004) Training algorithms for fuzzy support vector machines with noisy data. Pattern Recognit Lett 25(14):1647–1656
56. Lin CT, Liang SF, Yeh CM, Fan KW (2005) Fuzzy neural network design using support vector regression for function approximation with outliers. In: 2005 IEEE international conference on systems, man and cybernetics, IEEE, vol 3, pp 2763–2768
57. Liu CY, Sun L, Zhou ZJ (2013) Weighted support vector machine based on association rules. In: 2013 International conference on machine learning and cybernetics (ICMLC), IEEE, vol 1, pp 381–386
58. Liu D, Shi Y, Tian Y, Huang X (2016) Ramp loss least squares support vector machine. J Comput Sci 14:61–68
59. Liu T, Tao D (2016) Classification with noisy labels by importance reweighting. IEEE Trans Pattern Anal Mach Intell 38(3):447–461
60. Liu W, Pokharel PP, Príncipe JC (2007) Correntropy: properties and applications in non-Gaussian signal processing. IEEE Trans Signal Process 55(11):5286–5298
61. Ma Y, Li L, Huang X, Wang S (2011) Robust support vector machine using least median loss penalty. IFAC Proc Vol 44(1):11208–11213
62. Mangasarian OL, Wild EW (2001) Proximal support vector machine classifiers. In: Proceedings KDD-2001: knowledge discovery and data mining, Citeseer
63. Mangasarian OL, Wild EW (2006) Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Trans Pattern Anal Mach Intell 28(1):69–74
64. Mehrkanoon S, Huang X, Suykens JA (2014) Non-parallel support vector classifiers with different loss functions. Neurocomputing 143:294–301
65. Mohdiwale S, Sahu TP, Chaurasia RK, Nagwani NK, Verma S (2018) Detection and classification of noise using bark domain features. In: Proceedings of the 6th international conference on communications and broadband networking, ACM, pp 18–21
66. Mourão-Miranda J, Hardoon DR, Hahn T, Marquand AF, Williams SC, Shawe-Taylor J, Brammer M (2011) Patient classification as an outlier detection problem: an application of the one-class support vector machine. Neuroimage 58(3):793–804
67. Nasiri JA, Charkari NM, Mozafari K (2014) Energy-based model of least squares twin support vector machines for human action recognition. Signal Process 104:248–257
68. Ning K, Liu M, Dong M, Wu Z (2014) Robust ls-svr based on variational bayesian and its applications. In: 2014 International joint conference on neural networks (IJCNN), IEEE, pp 2920–2926
69. Niu J, Chen J, Xu Y (2017) Twin support vector regression with huber loss. J Intell Fuzzy Syst 32(6):4247–4258
70. Oliva JT, Rosa JLG (2017) The use of one-class classifiers for differentiating healthy from epileptic EEQ segments. In: 2017 International joint conference on neural networks (IJCNN), IEEE, pp 2956–2963
71. Oza P, Patel VM (2019) One-class convolutional neural network. IEEE Signal Process Lett 26(2):277–281
72. Park SY, Liu Y (2011) Robust penalized logistic regression with truncated loss functions. Can J Stat 39(2):300–323
73. Peng X (2010) A v-twin support vector machine (v-tsvm) classifier and its geometric algorithms. Inf Sci 180(20):3863–3875
74. Peng X (2010) Primal twin support vector regression and its sparse approximation. Neurocomputing 73(16–18):2846–2858
75. Peng X (2010) TSVR: an efficient twin support vector machine for regression. Neural Netw 23(3):365–372
76. Peng X (2011) TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition. Pattern Recognit 44(10–11):2678–2692
77. Rajalaxmi R, Vidhya E (2019) A mutated salp swarm algorithm for optimization of support vector machine parameters. In: 2019 5th International conference on advanced computing & communication systems (ICACCS), IEEE, pp 979–983
78. Rakhe SS, Vaidya AS (2015) A survey on different unsupervised techniques to detect outliers. Int Res J Eng Technol (IRJET) 2
79. Roy A, Singha J, Devi SS, Laskar RH (2016) Impulse noise removal using svm classification based fuzzy filter from grayscale images. Signal Process 128:262–273
80. Ruff L, Görnitz N, Deecke L, Siddiqui SA, Vandermeulen R, Binder A, Müller E, Kloft M (2018) Deep one-class classification. In: International conference on machine learning, pp 4390–4399
81. Rustam Z, Pandelaki J, Siahaan A, et al. (2019) Kernel spherical k-means and support vector machine for acute sinusitis classification. In: IOP conference series: materials science and engineering, IOP Publishing, vol 546, p 052011
82. Saeedi J, Ahadi SM, Faez K (2015) Robust voice activity detection directed by noise classification. Signal Image Video Process 9(3):561–572
83. Shao YH, Deng NY, Chen WJ, Wang Z (2013) Improved generalized eigenvalue proximal support vector machine. IEEE Signal Process Lett 20(3):213–216
84. Shao YH, Zhang CH, Yang ZM, Jing L, Deng NY (2013) An ε-twin support vector machine for regression. Neural Comput Appl 23(1):175–185
85. Shen X, Niu L, Qi Z, Tian Y (2017) Support vector machine classifier with truncated pinball loss. Pattern Recognit 68:199–210
86. Shi Y, Zhang L (2011) Coid: a cluster-outlier iterative detection approach to multi-dimensional data analysis. Knowl Inf Syst 28(3):709–733
87. Shin HJ, Eom DH, Kim SS (2005) One-class support vector machines-an application in machine fault detection and classification. Comput Ind Eng 48(2):395–408
88. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222
89. Song Q, Hu W, Xie W (2002) Robust support vector machine with bullet hole image classification. IEEE Trans Syst Man Cybern Part C (Appl Rev) 32(4):440–448
90. Sun XQ, Chen YJ, Shao YH, Li CN, Wang CH (2018) Robust nonparallel proximal support vector machine with lp-norm regularization. IEEE Access 6:20334–20347

91. Suykens JA, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293–300
92. Suykens JA, De Brabanter J, Lukas L, Vandewalle J (2002) Weighted least squares support vector machines: robustness and sparse approximation. Neurocomputing 48(1–4):85–105
93. Takruri M, Mahmoud A, Khaled M, Al-Jumaily A (2019) PSO-SVM hybrid system for melanoma detection from histo-pathological images. Int J Electr Comput Eng 2088–8708:9
94. Tang L, Tian Y, Yang C, Pardalos PM (2018) Ramp-loss nonparallel support vector regression: robust, sparse and scalable approximation. Knowl Based Syst 147:55–67
95. Tao PD et al (2005) The dc (difference of convex functions) programming and dca revisited with dc models of real world nonconvex optimization problems. Ann Oper Res 133(1–4):23–46
96. Tao Q, Wang J (2004) A new fuzzy support vector machine based on the weighted margin. Neural Process Lett 20(3):139–150
97. Tao Z, Huiling L, Wenwen W, Xia Y (2019) Ga-svm based feature selection and parameter optimization in hospitalization expense modeling. Appl Soft Comput 75:323–332
98. Tax DM, Duin RP (2004) Support vector data description. Mach Learn 54(1):45–66
99. Tian Y, Mirzabagheri M, Bamakan SMH, Wang H, Qu Q (2018) Ramp loss one-class support vector machine; a robust and effective approach to anomaly detection problems. Neurocomputing 310:223–235
100. Tomar D, Agarwal S (2014) Feature selection based least square twin support vector machine for diagnosis of heart disease. Int J Bio-Sci Bio-Technol 6(2):69–82
101. Tuba E, Strumberger I, Bacanin N, Jovanovic R, Tuba M (2019) Bare bones fireworks algorithm for feature selection and SVM optimization. In: 2019 IEEE congress on evolutionary computation (CEC), IEEE, pp 2207–2214
102. Vapnik VN (1998) Adaptive and learning systems for signal processing communications, and control. Stat Learn Theory
103. Vijayalakshmi V, Babu MS, Lakshmi RP (2018) Kfcm algorithm for effective brain stroke detection through SVM classifier. In: 2018 IEEE international conference on system, computation, automation and networking (ICSCA), IEEE, pp 1–6
104. Wang JS, Chiang JC (2008) A cluster validity measure with outlier detection for support vector clustering. IEEE Trans Syst Man Cybern Part B (Cybern) 38(1):78–89
105. Wang K, Zhong P (2014) Robust non-convex least squares loss function for regression with outliers. Knowl Based Syst 71:290–302
106. Wang L, Jia H, Li J (2008) Training robust support vector machine with smooth ramp loss in the primal space. Neurocomputing 71(13–15):3020–3025
107. Wang R, Li W, Li R, Zhang L (2019) Automatic blur type classification via ensemble svm. Signal Process Image Commun 71:24–35
108. Wang T-Y, Chiang H-M (2007) Fuzzy support vector machine for multi-class text categorization. Inf Process Manag 43(4):914–929
109. Wang Y, Wang S, Lai KK (2005) A new fuzzy support vector machine to evaluate credit risk. IEEE Trans Fuzzy Syst 13(6):820–831
110. Wang YF, Jiong Y, Su GP, Qian YR (2019) A new outlier detection method based on optics. Sustain Cities Soc 45:197–212
111. Wang Z, Shao YH, Bai L, Deng NY (2015) Twin support vector machine for clustering. IEEE Trans Neural Netw Learn Syst 26(10):2583–2588
112. Wang Z, Chen X, Li CN, Shao YH (2018) Ramp-based twin support vector clustering. arXiv preprint arXiv:181203710
113. Wang Z, Wang S, Kong D, Liu S (2019) Methane detection based on improved chicken algorithm optimization support vector machine. Appl Sci 9(9):1761
114. Wu Q, Law R (2010) Fuzzy support vector regression machine with penalizing gaussian noises on triangular fuzzy number space. Expert Syst Appl 37(12):7788–7795
115. Wu Y, Liu Y (2007) Robust truncated hinge loss support vector machines. J Am Stat Assoc 102(479):974–983
116. Wu Y, Liu Y (2013) Adaptively weighted large margin classifiers. J Comput Graph Stat 22(2):416–432
117. Xiao J (2019) SVM and KNN ensemble learning for traffic incident detection. Phys A Stat Mech Appl 517:29–35
118. Xiao Y, Wang H, Xu W, Zhou J (2016) Robust one-class SVM for fault detection. Chemom Intell Lab Syst 151:15–25
119. Xiao Y, Wang H, Xu W (2017) Ramp loss based robust one-class SVM. Pattern Recognit Lett 85:15–20
120. Xing HJ, Ji M (2018) Robust one-class support vector machine with rescaled hinge loss function. Pattern Recognit 84:152–164
121. Xu G, Cao Z, Hu BG, Principe JC (2017) Robust support vector machines based on the rescaled hinge loss function. Pattern Recognit 63:139–148
122. Xu H, Caramanis C, Mannor S (2009) Robust regression and lasso. In: Advances in neural information processing systems, pp 1801–1808
123. Xu L, Crammer K, Schuurmans D (2006) Robust support vector machine training via convex outlier ablation. AAAI 6:536–542
124. Xu Y, Guo R (2014) An improved *v*-twin support vector machine. Appl Intell 41(1):42–54
125. Xu Y, Wang L (2012) A weighted twin support vector regression. Knowl Based Syst 33:92–101
126. Xu Y, Lv X, Wang Z, Wang L (2014) A weighted least squares twin support vector machine. J Inf Sci Eng 30(6):1773–1787
127. Xu Y, Yang Z, Pan X (2017) A novel twin support-vector machine with pinball loss. IEEE Trans Neural Netw Learn Syst 28(2):359–370
128. Yang HY, Wang XY, Niu PP, Liu YC (2014) Image denoising using nonsubsampled shearlet transform and twin support vector machines. Neural Netw 57:152–165
129. Yang J, Deng T, Sui R (2016) An adaptive weighted one-class svm for robust outlier detection. In: Proceedings of the 2015 Chinese intelligent systems conference, Springer, pp 475–484
130. Yang L, Dong H (2018) Support vector machine with truncated pinball loss and its application in pattern recognition. Chemom Intell Lab Syst 177:89–99
131. Yang L, Zhang S (2016) A sparse extreme learning machine framework by continuous optimization algorithms and its application in pattern recognition. Eng Appl Artif Intell 53:176–189
132. Yang L, Ren Z, Wang Y, Dong H (2017) A robust regression framework with laplace kernel-induced loss. Neural Comput 29(11):3014–3039
133. Yang X, Song Q, Wang Y (2007) A weighted support vector machine for data classification. Int J Pattern Recognit Artif Intell 21(05):961–976
134. Yang X, Tan L, He L (2014) A robust least squares support vector machine for regression and classification with noise. Neurocomputing 140:41–52
135. Yang X, Han L, Li Y, He L (2015) A bilateral-truncated-loss based robust support vector machine for classification problems. Soft Comput 19(10):2871–2882
136. YangX W, ZhangG Q et al (2011) A kernel fuzzy c-means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises. IEEE Trans FuzzySyst 19(1):105–115

137. Ye Q, Zhao H, Li Z, Yang X, Gao S, Yin T, Ye N (2018) L1-norm distance minimization-based fast robust twin support vector *k*-plane clustering. IEEE Trans Neural Netw Learn Syst 29(9):4494–4503
138. Ye YF, Shao YH, Deng NY, Li CN, Hua XY (2017) Robust lp-norm least squares support vector regression with feature selection. Appl Math Comput 305:32–52
139. Yin S, Zhu X, Jing C (2014) Fault detection based on a robust one class support vector machine. Neurocomputing 145:263–268
140. You L, Jizhen L, Yaxin Q (2011) A new robust least squares support vector machine for regression with outliers. Procedia Eng 15:1355–1360
141. Yuille AL, Rangarajan A (2003) The concave–convex procedure. Neural Comput 15(4):915–936
142. Zhang Y, Xie F, Huang D, Ji M (2010) Support vector classifier based on fuzzy c-means and mahalanobis distance. J Intell Inf Syst 35(2):333–345
143. Zhao Y, Sun J (2008) Robust support vector regression in the primal. Neural Netw 21(10):1548–1555
144. Zhao YP, Zhao J, Zhao M (2013) Twin least squares support vector regression. Neurocomputing 118:225–236
145. Zhong P, Xu Y, Zhao Y (2012) Training twin support vector regression via linear programming. Neural Comput Appl 21(2):399–407
146. Zhu F, Yang J, Gao C, Xu S, Ye N, Yin T (2016) A weighted one-class support vector machine. Neurocomputing 189:1–10
147. Zhu W, Song Y, Xiao Y (2018) A new support vector machine plus with pinball loss. J Classif 35(1):52–70