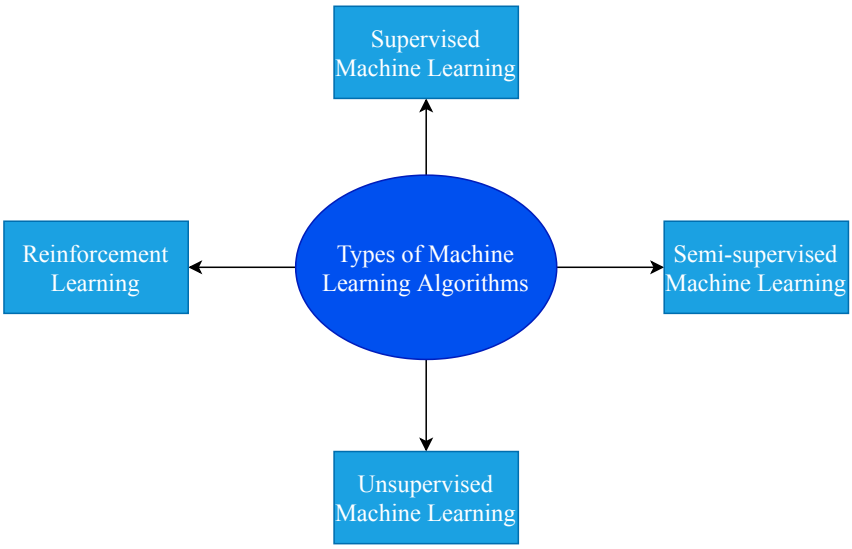

Figure 1.1: Types of Machine Learning Algorithms

points, the classical soft margin SVM can be represented as [9]

$$\min_{w,b} \lambda \|w\|^2 + \frac{1}{m} \sum_{i=0}^{m} \max(0, 1 - y_i(w^T x_i + b)), \tag{1.1}$$

where $\lambda \|w\|^2$ is the regularization term with λ as the regularization parameter and $\max(0, 1 - z)$ is the hinge loss function, where $z = y(w^T x + b)$. Note that w is the weight vector normal to the hyperplane, and b is the bias term. This algorithm is used for regression as well, and the regression variant of SVM is support vector regression (SVR) [10]. The optimization problem of SVR corresponding to the hyperplane $f(x_i) = w^T x_i + b$ is

$$\begin{aligned} \min_{w,b} \ & \frac{1}{2} \|w\|^2 + \lambda \sum_{i=1}^{m} (\xi_i + \xi_i^*) \\ \text{s.t. } & y_i - (w^T x_i + b) \leq \varepsilon + \xi_i, \\ & (w^T x_i + b) - y_i \leq \varepsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0, \end{aligned} \tag{1.2}$$

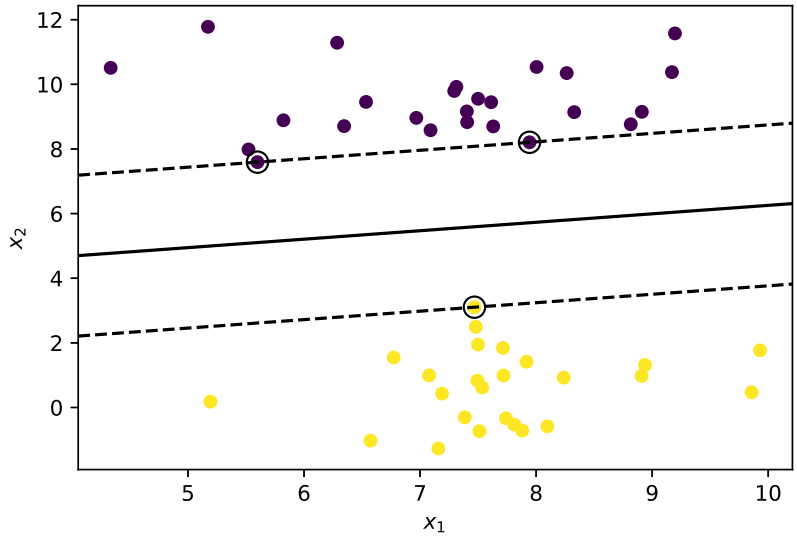

Figure 1.2: Support Vector Machine

where ξ_i and ξ_i* are the slack variables to handle the presence of infeasible constraints, and ε denotes the margin of error tolerance. Although SVM/SVR is a classical ML algorithm, it still contributes to many real-world applications like cancer genomics [11], bioinformatics [12], time-series prediction [13], electrocardiography (ECG) signal processing [14] and medical diagnosis [15–17] etc. Due to its efficient classification/ regression, the model has been applied to other machine learning frameworks. In the next subsection, the variants of SVM are discussed.

1.1 Variants of SVM

Variants of SVM are those models whose base is SVM [1]. In this thesis, supervised and semi-supervised variants of SVM are focused. These are shown in Figure 1.3.

(a) **Least Square SVM**

Toward this direction, SVM was first extended to 'Least Square SVM (LS-SVM) [18].' This variant was proposed to ease the implementation of conventional SVM. This extension made use of the square loss function. The optimization problem

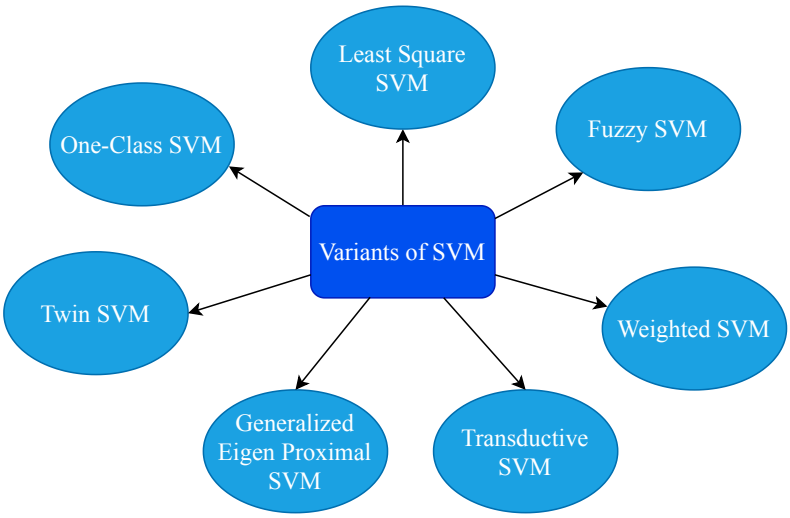

Figure 1.3: Variants of SVM [1]

for LS-SVM is given by [18]

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + \frac{\lambda}{2}\sum_{i=1}^{m}\xi_i^2 \\ \text{s.t.} \quad & y_i(w^T x_i + b) = 1 - \xi_i, \ i = 1, \ldots, m. \end{aligned} \tag{1.3}$$

LS-SVM was proposed to make the optimization problem easier to solve. The solution of LS-SVM is obtained after constructing the Lagrangian function which yield a linear system to get the solution instead of solving quadratic programming as proposed in SVM [18]. This is shown in Appendix A.

This variant of SVM was proposed in 1999. The main aim behind this variant is to make the implementation of SVM easier. The difference between SVM implementation and LS-SVM is that the conventional SVM implements the quadratic formulation while LS-SVM implements linear equations.

(b) **Fuzzy SVM (FSVM)**

SVM was later merged with fuzzy logic in such a manner that a fuzzy membership value $\sigma \leq s_i \leq 1$ $(\sigma \geq 0)$ was also attached with the dataset $\{x_i, y_i, s_i\}$, $i =$

$1, \ldots, m$ where m is the number of data points. FSVM creates a decision boundary according to importance of the data points in the training set. In 2002, fuzzy membership was firstly applied to SVM, where different data points were assigned with various fuzzy membership values to contribute to the decision surface [19]. The optimization problem corresponding to FSVM is

$$\begin{aligned} \min_{w,b,\xi_i} \quad & \frac{1}{2}\|w\|^2 + \lambda \sum_{i=1}^{m} s_i \xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, m \\ & \xi_i \geq 0, \ i = 1, \ldots, m. \end{aligned} \tag{1.4}$$

The method of FSVM can be used to classify data points with noise or outliers. Although FSVM 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 [20].

Fuzzy logic was used with the classification problems, and regression problems [21]. Fuzzy logic was also used in designing other classifiers using SVM or SVR (in case of regression) [21]. The ideas of fuzzy neural networks and fuzzy SVM were combined to form a new FSVM [22] which was more robust than these two. That method of using fuzzy membership with different classifiers was used in many applications. FSVM was used to evaluate credit risk [23], for multi-class text categorization [24], etc.

Hence, FSVM was proposed to reformulate SVM in such a way that it assigns fuzzy membership values to different data points so that these can make different contributions to the decision boundary [19].

(c) **Weighted SVM**

The following variant of SVM proposed is weighted SVM. The 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 weight assignment task was done using a fuzzy clustering algorithm, and kernel-based possibilistic C-means algorithm [25]. The optimization problem obtained after assigning weights W_i is given by

$$\begin{aligned} \min_{w,b,\xi} \ & \frac{1}{2}\|w\|^2 + \frac{\lambda}{2}\sum_{i=1}^{m} W_i \xi_i \\ \text{s.t.} \ & y_i(w^T x_i + b) \geq 1 - \xi_i, \ i = 1, \ldots, m. \end{aligned} \tag{1.5}$$

It should be noted here that the weighted SVM was proposed to improve the outlier sensitivity problem of SVM. The complexities of training and testing can also be adjusted using the pruning method according to the outliers present in the dataset. This method yielded a higher classification rate than standard SVM [25].

(d) **Transductive SVM (TSVM)**

The above-discussed variants come under the category of supervised ML. Transductive SVM is the semi-supervised variant of SVM [26]. In this model, a significant number of unlabeled samples are used during training. Hence the name 'transductive' is given to this algorithm. Let there is a set $\mathbb{L}$ of labeled samples, L given by

$$\mathbb{L} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_L, y_L)\}, \ x \in \mathbb{R}^d, \tag{1.6}$$

with labels $y \in \{-1, +1\}$ and a set $\mathbb{U}$ of unlabeled samples, U, given by

$$\mathbb{U} = \{x_{L+1}, x_{L+2}, \ldots, x_{L+U}\}, \ x \in \mathbb{R}^d. \tag{1.7}$$

The task is to find a suitable hyperplane characterized by $H = (w, b)$. New labels are assigned using these optimal weight vector, w, and bias term, b, in

$$f_H(x) = w^T x + b. \tag{1.8}$$

To obtain w and b, the optimization problem considered for TSVM is [27]

$$\begin{aligned} \min_{w,b} \ & \frac{1}{2}\|w\|^2 + \lambda \sum_{i=1}^{L} \xi_i + \lambda^* \sum_{j=L+1}^{L+U} \xi_j \\ \text{s.t.} \ & y_i(w^T x_i + b) \geq 1 - \xi_i, \ i = 1, 2, \ldots, L, \\ & |w^T x_j + b| \geq 1 - \xi_j, \ j = L+1, L+2, \ldots, L+U. \end{aligned} \tag{1.9}$$

Hence, TSVM was proposed for semi-supervised learning. Through this variant of SVM, the labels are assigned to the unlabeled samples.

(e) **Generalized Eigen Proximal SVM (GEPSVM)**

Another variant is generalized Eigen proximal SVM originated from proximal SVMs (PSVM) [28]. PSVMs are based on getting two parallel planes close to the datasets of the respective classes but as far as possible. 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 [29].

Since its evolution, researchers are having a keen interest in developing various variants of it. Because of GEPSVMs, a twin support vector machine (TWSVM) was introduced, and this model has helped a lot in the growth of machine learning.

GEPSVM has two main variants for classification: one is regularized GEPSVM [30] and the other one is improved GEPSVM given by Shao et al. [31]. The opti-

mization problem includes replacing the generalized Eigen decomposition with the conventional Eigen value decomposition. Both the improved versions of GEPSVM helped in reducing the computational time.

(f) **Twin Support Vector Machine (TWSVM)**

GEPSVM introduced TWSVM to the world of machine learning, which was faster than the widely used model, SVM, and showed good generalization. TWSVM was initially proposed for two-class classification [32]. The algorithm defined two hyperplanes, one for each class. It classified points, according to which hyperplane a given point is closest to [32]. TWSVM1(equation corresponding to the first hyperplane) is defined by [32]

$$\begin{aligned} \min_{w_1,b_1} \quad & \frac{1}{2}(x_1w_1 + e_1b_1)^T(x_1w_1 + e_1b_1) + \lambda e_2^T\xi \\ \text{s.t.} \quad & -(x_2w_1 + e_2b_1) + \xi \geq e_2,\ \xi \geq 0. \end{aligned} \tag{1.10}$$

Similarly, TWSVM2 is defined by

$$\begin{aligned} \min_{w_2,b_2} \quad & \frac{1}{2}(x_2w_2 + e_2b_2)^T(x_2w_2 + e_2b_2) + \lambda^* e_1^T\xi \\ \text{s.t.} \quad & -(x_1w_2 + e_1b_2) + \xi \geq e_1,\ \xi \geq 0, \end{aligned} \tag{1.11}$$

where x_1, x_2 are the training points belong to class 1 and class 2 respectively, w_1, w_2 and b_1, b_2 are the weight vectors and bias terms corresponding to TWSVM1 and TWSVM2, respectively. λ, λ^* are the regularization parameters and ξ is the term used to indicate slackness while classifying the points. Minimization of the first term in both the above equations indicates 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 the other class. In (1.10) and (1.11), e_1 and e_2 are the vectors of ones 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. [32] also proved that TWSVM is four times faster than SVM. The effect of outliers or noise was not discussed in that work, but various robust variants of TWSVM were also proposed because of its low computational time.

TWSVM was extended for regression problems as well. Both the models had some significant differences [33] except one similarity of getting two non-parallel planes around the data points. The equations for two hyperplanes proposed for twin support vector regression (TSVR) are:

$$\begin{aligned} \min_{w_1,b_1} \ & \frac{1}{2}(Y - e\varepsilon_1 - (x_1w_1 + eb_1))^T(Y - e\varepsilon_1 - (x_1w_1 + eb_1)) + \lambda e^T\xi \\ \text{s.t.} \ & Y - (x_1w_1 + eb_1) \geq e\varepsilon_1 - \xi, \ \xi \geq 0, \end{aligned} \tag{1.12}$$

and

$$\begin{aligned} \min_{w_2,b_2} \ & \frac{1}{2}(Y + e\varepsilon_2 - (x_2w_2 + eb_2))^T(Y + e\varepsilon_2 - (x_2w_2 + eb_2)) + \lambda^* e^T\xi^* \\ \text{s.t.} \ & (x_2w_2 + eb_2) - Y \geq e\varepsilon_2 - \xi^*, \ \xi^* \geq 0, \end{aligned} \tag{1.13}$$

where $\lambda, \lambda^*, \varepsilon_1, \varepsilon_2 \geq 0$ are the parameters used and ξ, ξ^* are the slack vectors.

(g) **One-class SVM (OCSVM)**

It is a supervised variant of SVM, which was proposed to handle the issue of class imbalance. Only one class is considered in this classification, and the model is trained based on that class. During the testing phase, the test set is evaluated based on the model trained and results in whether the data points of the test set belong to that class or not [34].

For the dataset $[x_1, x_2, \ldots, x_m] \in \mathbb{R}^{m\times n}$, the optimization problem corresponding to OCSVM is given by [34]

$$\min_{w,\xi,\rho} \ \frac{1}{2}\|w\|^2 + \frac{1}{m\lambda}\sum_{i=1}^{m}\xi_i - \rho$$
$$\text{s.t.} \ \ w.\phi(x_i) \geq \rho - \xi_i, \ \xi_i \geq 0, \tag{1.14}$$

where m are the data points in the data set, λ is the regularization parameter, ξ_i is the slack variable for point x_i that allows it to locate outside the decision boundary. The equation of the decision boundary is given by

$$f(x) = w.\phi(x) - \rho \tag{1.15}$$

where ϕ is the mapping function with $x \in \mathbb{R}^m$.

The above-discussed variants have their pros and cons, and these are described in Table 1.1.

Table 1.1: Summary of Variants of SVM

Variants	**Pros**	**Cons**
LS-SVM [18]	Convert inequality constraints to the equality constraints	Lost Sparseness
	Easy to solve optimization problem	Lost robustness

FSVM [19]	Classifies according to the fuzzy membership values assigned to the data points More feasible in reducing the effects of noise than SVM	How to adaptively determine a suitable model of fuzzy membership function
Weighted SVM [25]	Provides higher classification rate than SVM	Weight assignment is a problem
	Less sensitive to outliers	Computational complexity is higher
TSVM [26]	Considers even unlabeled data points Can also be used for semi-supervised learning	Optimization problem is relatively harder to solve
GEPSVM [28]	Computational time is low Solves two equations for non-parallel planes instead of one complex quadratic problem	Still outlier sensitive

TWSVM [32]	Four times faster than SVM Can handle class imbalance problem	Needs to compute the inverse of matrices in standard TWSVM Standard TWSVM/TSVR are outlier sensitive
	Its variants nicely handles outlier sensitivity problem	Lack of sparseness
OCSVM [34]	Handles class imbalance Train the classifier using only patterns belonging to target class	Outlier sensitive

Although these models are proposed to handle various SVM issues, like the complex optimization technique, class imbalance and more computational time, they inherit few SVM properties due to hinge loss function in them. In the following subsection, we briefly explain the hinge loss function and its properties.

1.2 Hinge Loss Function

Hinge loss function is a convex loss function defined as

$$H_1(z) = \max(0, 1 - z) \tag{1.16}$$

where $z = y(w^T x + b)$. This loss function is conventionally used with SVM [9]. The plot for the hinge loss function is shown in Figure 1.4. In Figure 1.4, z is plotted on

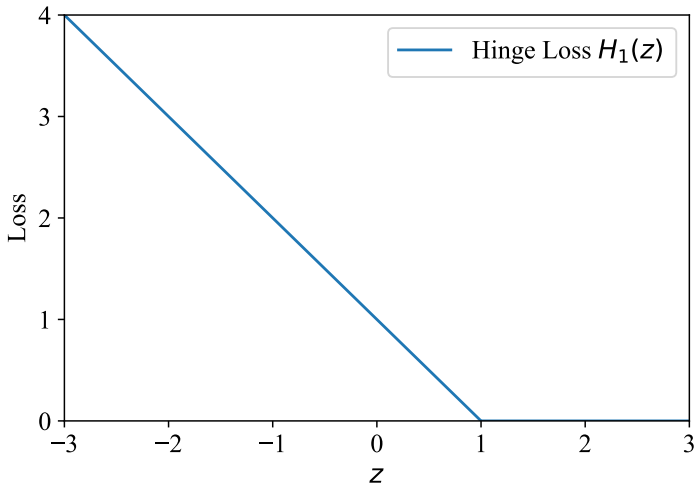

Figure 1.4: Hinge Loss Function

x-axis and 'Loss' is plotted on the y-axis. From this plot, it can be observed that this function is convex.

Properties of Hinge Loss Function

Besides convexity, it can be observed from Figure 1.4 that the function is monotonous, continuous, non-differentiable, unbounded, and hence sensitive to noise [1]. Due to hinge loss function in the objective formulation of SVM, SVM also carries these properties. Since the above-discussed variants are generated from the parent SVM, they also use the conventional hinge loss function in their formulation, and hence, the variants are also sensitive to noise and outliers.

In this thesis, the sensitivity of SVM and its variants towards noise and outliers is the primary concern. In the next section, the robust statistics in machine learning,

particularly in SVM and its variants, are introduced.

1.3 Robust Statistics

The term 'robustness' means the capability of performing under a wide range of conditions. In this thesis, robust machine learning is focused. Robust machine learning refers to the algorithm's ability to cope with noise in the data or uncertainty in the parameters. The robustness can be used in two contexts: robustness towards noise in the data (robust statistics) and robustness towards uncertainty (robust optimization).

Although SVM has proved its superiority by classifying well in various real-world applications, the performance gets affected when there is some uncertainty in the data set, model uncertainty, data uncertainty, or it can be parametric uncertainty [35]. SVM is highly sensitive to these uncertainties [35]. It leads to the introduction of robust optimization. Robust optimization handles the model training so that the model's performance can be least affected despite these uncertainties.

Similarly, SVM and its variants are highly sensitive to noise and outliers in the data set. This is due to the presence of the unbounded hinge loss function in their formulation. This is the point where 'robust statistics' comes into the picture.

Robust Statistics is deployed to ensure the algorithm's excellent performance corresponding to data obtained from various sources under a probabilistic distribution. Now the data usually obtained from various sources may contain outliers in it. These outliers can directly affect the algorithm's performance, so to tackle these problems, various techniques were used to make the algorithm robust [1]. The generalized framework of robust statistics is shown in Figure 1.5.

From Figure 1.5, it can be observed that when the training data corrupted by noise or outliers are passed to SVM or its variants, some robust statistical techniques are applied to get a robust model. After applying the robust statistical techniques, the new formulation is optimized by selecting the suitable optimization technique. This

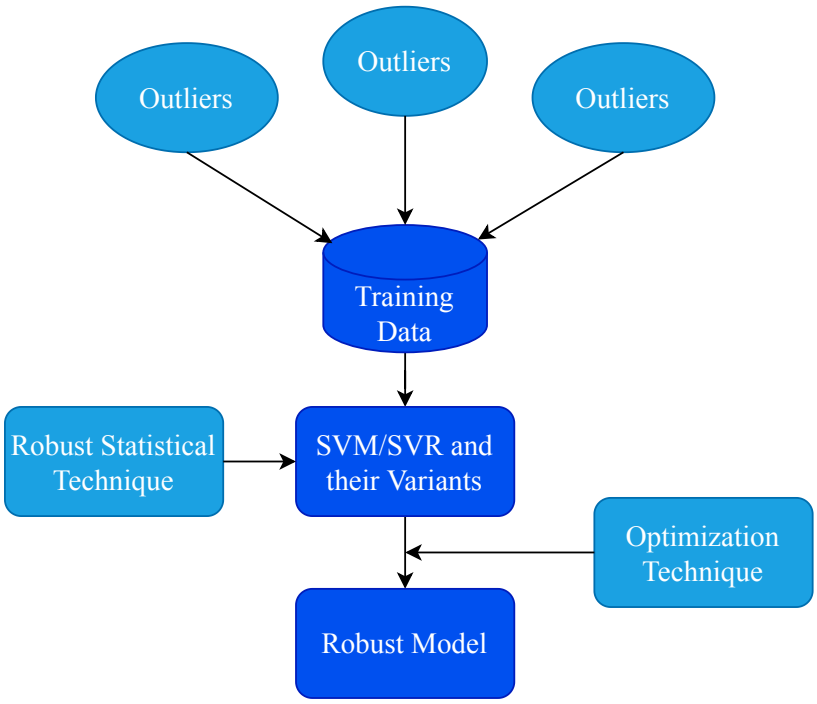

Figure 1.5: Generalized Framework of Robust Statistics [1]

produces a robust model which is resistant to noise or outliers and also which is time-efficient.

Robust statistical techniques are of two types: the outliers are first detected from the training set and then removed. The second robust statistical technique includes using robust estimators that can handle outliers' presence without removing them [1]. The first technique leads to the loss of information. In this thesis, the second technique is considered, where the outliers are not removed from the training set.

The robust loss functions for SVM and its variants are proposed to handle noise or outliers in the data sets. This leads to the generation of new objective functions. These formulations are optimized using suitable optimization techniques. The convergence proof of the proposed robust models is also provided. The robust formulations are also applied to many real-world applications. In the next section, the main motivations behind this work are described.

1.4 Motivation

Several machine learning researchers have recently focused their attention on imparting robustness to the models. Since machine learning is essentially a data-driven approach, the resulting model should be robust against noisy data. Specifically, concerning SVMs, it is observed that many robust variants [25, 36, 37] of SVM are proposed to make the model robust to noise and outliers. The researches reported in this literature motivated us to add robustness to SVM and its variants.

From the theory of robust statistics, it has been observed that the optimal selection of robust statistical and optimization techniques leads to the generation of a model that is robust towards noise or outliers. This motivated us to use robust statistics in SVM and its variants.

1.5 Research Contributions

The significant research contributions are listed below:

(a) The first contribution is the addition of robustness in the conventional SVM. The framework is proposed to handle classification tasks. Towards this direction, a robust loss function rescaled α-hinge loss function is proposed. The regularizer used in this robust formulation is the non-smooth L_2 regularizer, and the final objective function is optimized using the primal-dual proximal (PDprox) technique. The proposed approach also added sparsity to the model. This work also showed how the different values of the regularization parameter affect the accuracy of the proposed approach.

(b) The following contribution is the addition of robustness to TSVR. The model is made robust against Gaussian and uniform noise in the data set. In this work, the rescaled hinge loss function is used with a regularization term. It also helped in overcoming the issues of overfitting. The final optimization problem is solved using

the half-quadratic optimization technique. The dual of the proposed objective function is also formulated, which is further implemented using the quadprog() function in MATLAB.

(c) The above two contributions come under the category of supervised ML. The next is in the field of semi-supervised learning, in which the robustness is added to TSVM. In this literature, the truncated pinball loss function is used with TSVM and named as $\overline{\text{pin}}$-TSVM. The TSVM is made robust to label noise in the data set. Concave-convex procedure (CCCP) is used to solve the non-convex problem, further optimized using SGD. The dual of this problem is also formulated and implemented using mlcv_quadprog. The model is proposed for both small and large real-world data sets.

(d) The robust approach, $\overline{\text{pin}}$-TSVM, is also applied to a real-world application of detecting COVID-19 infected patients using chest X-ray images. In this work, the data set is generated by collecting chest X-ray images from various sources. The data set consists of chest X-ray images of normal patients, patients with the bacterial infection, and the COVID-19 infected patients. The pre-trained VGG19 model is used to extract features from the data set. These features are then used to train the proposed model, $\overline{\text{pin}}$-TSVM. This approach also helped in assigning labels to the unlabeled samples.

(e) The following contribution is the detection of diabetic retinopathy (DR) using eye fundus images. In this literature, TWSVM is used in this application to reduce the computational time. This is for the first time that TWSVM has been used for diabetic retinopathy (DR) detection. The pinball loss function is also in this formulation to get robust DR detection.

1.6 Organization of the Thesis

The rest of the thesis is organized as follows. The next chapter discusses the related work on robust statistics based on SVM and its variants published in reputed journals and conferences. In Chapter 3, the first contribution towards robust SVM, RSVM-PDProx, is described. In this chapter, the formulation of rescaled α-hinge loss function and the detailed formulation of the proposed approach, RSVM-PDProx, are given. The next chapter, Chapter 4, is based on the second contribution in this thesis. In this chapter, the proposed approach, Res-TSVR, is discussed. In Chapter 5, the next contribution, which is in semi-supervised learning, is described. In this chapter, robustness is added to TSVM and the proposed formulation of TSVM, $\overline{\text{pin}}$-TSVM, is mentioned. The formulation is also applied to the detection of COVID-19 infected patients ising chest X-ray images. In Chapter 6, another real-world application which is DR detection using eye fundus images, is considered. Chapter 7 concludes the above discussed contributions in this thesis. The possible future scope in this research is also mentioned in this chapter.