RoBoSS: A Robust, Bounded, Sparse, and Smooth Loss Function for Supervised Learning

Mushir Akhtar, M. Tanveer,* *Senior Member, IEEE, Mohd. Arshad*

Abstract—In the domain of machine learning algorithms, the significance of the loss function is paramount, especially in supervised learning tasks. It serves as a fundamental pillar that profoundly influences the behavior and efficacy of supervised learning algorithms. Traditional loss functions, while widely used, often struggle to handle noisy and high-dimensional data, impede model interpretability, and lead to slow convergence during training. In this paper, we address the aforementioned constraints by proposing a novel robust, bounded, sparse, and smooth (RoBoSS) loss function for supervised learning. Further, we incorporate the RoBoSS loss function within the framework of support vector machine (SVM) and introduce a new robust algorithm named $\mathcal{L}_{rbss}$-SVM. For the theoretical analysis, the classification-calibrated property and generalization ability are also presented. These investigations are crucial for gaining deeper insights into the performance of the RoBoSS loss function in the classification tasks and its potential to generalize well to unseen data. To empirically demonstrate the effectiveness of the proposed $\mathcal{L}_{rbss}$-SVM, we evaluate it on 88 real-world UCI and KEEL datasets from diverse domains. Additionally, to exemplify the effectiveness of the proposed $\mathcal{L}_{rbss}$-SVM within the biomedical realm, we evaluated it on two medical datasets: the electroencephalogram (EEG) signal dataset and the breast cancer (BreaKHis) dataset. The numerical results substantiate the superiority of the proposed $\mathcal{L}_{rbss}$-SVM model, both in terms of its remarkable generalization performance and its efficiency in training time. The code of the proposed model is publicly available at https://github.com/mtanveer1/RoBoSS.

Index Terms—Supervised Machine Learning (SML), Classification, Loss Functions, Support Vector Machine (SVM), RoBoSS Loss Function.

I. Introduction and Motivation

DATA analysis tasks such as classification and regression fall under the umbrella of supervised machine learning (SML). SML is a powerful paradigm in machine learning wherein a model learns from labeled data to make predictions on unseen instances. Key to this process is the concept of loss functions, which quantify the discrepancy between predicted and actual outputs. Support Vector Machine (SVM) [1] represents an efficient SML algorithm. It is based on the concept of structural risk minimization (SRM) and has its roots in statistical learning theory (SLT) [2], providing it with a strong theoretical base and good generalization ability. In this paper, we undertake an in-depth examination of the interrelation between loss functions and the supervised learning algorithm, utilizing the framework of SVM.

This study is solely focused on the binary classification task. Let the training set be defined by $\{x_k, y_k\}_{k=1}^{n}$, where $x_k \in$

*Corresponding author
Mushir Akhtar, M. Tanveer and Mohd. Arshad are with the Department of Mathematics, Indian Institute of Technology Indore, Simrol, Indore, 453552, India (e-mail: phd2101241004@iiti.ac.in, mtanveer@iiti.ac.in, arshad@iiti.ac.in).

$\mathbb{R}^m$ indicates the sample vector and $y_k \in \{1, -1\}$ indicates the corresponding label of the class. The aim of SVM is to construct a decision hyperplane $w^\mathsf{T} x + b = 0$ with bias $b \in \mathbb{R}$ and weight vector $w \in \mathbb{R}^m$, which are estimated by training data. For a test data point $\widetilde{x}$, the corresponding class label $\widetilde{y}$ is predicted as 1 if $w^\mathsf{T}\widetilde{x} + b \geq 0$ and -1 otherwise. To obtain the optimal hyperplane, two situations can be considered in the input space: linearly separable and linearly inseparable training datasets.

For linearly separable situation, the optimal parameters w and b are obtained by addressing the following SVM model:

$$\min_{w,b} \ \frac{1}{2}\|w\|^2$$
$$\text{subject to} \quad y_k\left(w^\mathsf{T} x_k + b\right) \geq 1, \ \forall \ k = 1, 2, \dots, n. \quad (1)$$

The model (1) is termed as hard-margin SVM since it necessitates every training sample to be correctly classified.

For linearly inseparable situation, the widely used approach permit misclassification and penalize these violations by including the loss function in the objective function, which results in the following unconstrained optimization problem:

$$\min_{w,b} \ \frac{1}{2}\|w\|^2 + \frac{\gamma}{n}\sum_{k=1}^{n} \mathcal{L}\left(1 - y_k\left(w^\mathsf{T} x_k + b\right)\right), \quad (2)$$

where $\gamma > 0$ is a trade-off parameter and $\mathcal{L}(u)$ with $u := 1 - y_k\left(w^\mathsf{T} x_k + b\right)$ denotes the loss function. Since model (2) allows misclassification of samples, it is referred to as a soft-margin SVM model [1].

The loss function $\mathcal{L}(u)$ is an essential component of support vector machine, which controls the robustness and sparsity of SVM. The 0-1 loss function is defined as an ideal loss function [1] that assigns a fixed loss of 1 to all misclassified samples and no loss to correctly classified samples.

$$\mathcal{L}_{0-1}(u) = \begin{cases} 1, & u > 0, \\ 0, & u \leq 0. \end{cases} \quad (3)$$

However, solving SVM with 0-1 loss function is NP-hard [3, 4] since it is discontinuous and non-convex. For the development of SVM, a great deal of work has gone into constructing new loss functions to obtain new effective soft-margin SVM models. Here, we briefly reviewed a few renowned loss functions, which are sufficient to serve as inspiration for the rest of this paper.

The first soft-margin SVM model is hinge loss SVM ($\mathcal{L}_{hinge}$-SVM), which utilizes the hinge loss function
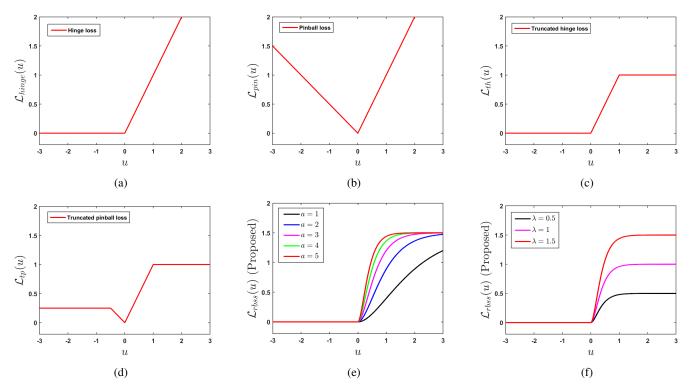
Fig. 1: (a) Hinge loss function. (b) Pinball loss function with $\tau = 0.5$. (c) Truncated hinge loss with $\delta = 1$. (d) Truncated pinball loss with $\tau = 0.5$, $\delta_1 = 1$, and $\delta_2 = 0.25$. (e) Proposed RoBoSS loss with fixed $\lambda = 1.5$ and different values of a. (f) Proposed RoBoSS loss with fixed $a = 5$ and different values of λ.

$\mathcal{L}_{hinge}(u)$ (see Fig. 1a), and is defined as:

$$\mathcal{L}_{hinge}(u) = \begin{cases} u, & u > 0, \\ 0, & u \leq 0. \end{cases} \tag{4}$$

The hinge loss function is convex, non-smooth, and unbounded. To improve the efficacy of $\mathcal{L}_{hinge}$-SVM, Huang et al. [5] first studied pinball loss SVM ($\mathcal{L}_{pin}$-SVM), which utilizes pinball loss function $\mathcal{L}_{pin}(u)$ (see Fig. 1b) and is defined as:

$$\mathcal{L}_{pin}(u) = \begin{cases} u, & u > 0, \\ -\tau u, & u \leq 0, \end{cases} \tag{5}$$

where $\tau \in [0, 1]$. For $\tau = 0$, the pinball loss function is reduced to the hinge loss function. For $\tau \in (0, 1]$, it also provides penalty to correctly classified samples, which diminishes the sparseness [6]. The pinball loss function is also convex, non-smooth, and unbounded. Some other convex loss functions are least square loss function [7], generalized hinge loss function [8], linex loss function [9], huberized pinball loss function [10], and so on.

The convexity of loss functions is acknowledged as highly preferable due to their computational advantages. Specifically, they possess unique optima, are easy to use, and can be efficiently optimized using convex optimization tools. However, the convex loss functions provide poor approximations of 0-1 loss function and exhibit a lack of robustness to outliers due to their boundlessness, which makes the corresponding classifier susceptible to being overly influenced or dominated by outliers [11]. To improve the robustness, various bounded

loss functions are suggested in the literature. In order to increase the robustness of $\mathcal{L}_{hinge}$-SVM, Wu and Liu [12] developed truncated hinge loss function $\mathcal{L}_{th}(u)$ (see Fig. 1c), which is defined as:

$$\mathcal{L}_{th}(u) = \begin{cases} \delta, & u \geq \delta, \\ u, & u \in (0, \delta), \\ 0, & u \leq 0, \end{cases} \tag{6}$$

where $\delta \geq 1$. It is non-convex, non-smooth, and bounded. Other relevant research focuses on the development of new algorithms for solving truncated hinge loss SVM, such as the branch and bound algorithm [13], the convex-concave procedure (CCCP) [14], and so on. To enhance the robustness and sparseness of $\mathcal{L}_{pin}$-SVM, Yang and Dong [15] proposed the truncated pinball loss function $\mathcal{L}_{tp}(u)$ (see Fig. 1d), and is defined as:

$$\mathcal{L}_{tp}(u) = \begin{cases} \delta_1, & u \geq \delta_1, \\ u, & u \in [0, \delta_1), \\ -\tau u, & u \in (-\delta_2/\tau, 0), \\ \delta_2, & u \leq -\delta_2/\tau, \end{cases} \tag{7}$$

where $\tau \in [0, 1]$, and $\delta_1, \delta_2 > 0$. It gives a fixed loss δ_1 for samples with $u \geq -\delta_1$, which enhances the robustness and a fixed loss δ_2 for samples with $u \leq -\delta_2/\tau$, which adds the sparseness to $\mathcal{L}_{pin}$-SVM. It is also non-convex, non-smooth, and bounded. The optimization of truncated pinball loss SVM is addressed by the popular and efficient CCCP algorithm. The non-convex and non-smooth nature of the

aforementioned loss functions poses significant challenges in terms of computational optimization for solving corresponding SVM models.

Motivated by the previous works, the main focus of this paper is to construct a new robust, bounded, sparse, and smooth loss function for supervised learning. To improve the robustness, sparsity, and smoothness of the aforementioned losses, we design a new loss function named RoBoSS loss function (see Fig. 1e and 1f), which is defined as:

$$\mathcal{L}_{rbss}(u) = \begin{cases} \lambda\Big(1 - (au+1)exp(-au)\Big), & u > 0, \\ 0, & u \leq 0, \end{cases} \quad (8)$$

where $a > 0$, $\lambda > 0$ are shape and bounding parameters, respectively. Table I compares the attributes of various state-of-the-art loss functions with the proposed RoBoSS loss and demonstrates that the proposed loss has all the desirable characteristics.

Further, we incorporate the proposed RoBoSS loss function in SVM and introduce a new robust SVM model termed as $\mathcal{L}_{rbss}$-SVM. By replacing $\mathcal{L}(\cdot)$ by $\mathcal{L}_{rbss}(\cdot)$ in (2) yields us to get the proposed $\mathcal{L}_{rbss}$-SVM model, which is given by

$$\min_{w,b} \frac{1}{2}\|w\|^2 + \frac{\gamma}{n}\sum_{k=1}^{n}\mathcal{L}_{rbss}\Big(1 - y_k\left(w^\mathsf{T}x_k + b\right)\Big). \quad (9)$$

The non-convex nature of the proposed loss function poses challenges for optimizing the $\mathcal{L}_{rbss}$-SVM by the Wolfe-dual method. However, the smoothness of $\mathcal{L}_{rbss}$-SVM allows us to employ the gradient-based algorithm to solve the model. In this paper, we utilize the Nestrov accelerated gradient (NAG) based framework to solve the optimization problem of $\mathcal{L}_{rbss}$-SVM. NAG is known for its low computational complexity and efficiency in handling large-scale problems [16]. The main contributions of this work can be summarized as follows:

- We introduce an innovative advancement in the realm of supervised learning: the RoBoSS (Robust, Bounded, Sparse, and Smooth) loss function.
- We delved into the theoretical aspects of the RoBoSS loss function and showed it possesses two crucial properties: classification-calibration and a bound on generalization error. These results not only emphasize the robustness of the RoBoSS loss function but also provide valuable insights into its performance and applicability.
- We integrate the proposed RoBoSS loss function into the framework of SVM and propose a novel SVM model coined as $\mathcal{L}_{rbss}$-SVM. The resulting $\mathcal{L}_{rbss}$-SVM model harnesses the inherent strengths of both the RoBoSS loss function and the SVM algorithm, leading to an advanced and versatile machine learning tool.
- We carried out the experiments on 88 UCI and KEEL benchmark datasets from diverse domains. The experimental outcomes validate the effectiveness of the $\mathcal{L}_{rbss}$-SVM model when compared to the baseline models.
- Furthermore, to showcase the prowess of the proposed $\mathcal{L}_{rbss}$-SVM model in the biomedical domain, we conducted additional experiments using two medical datasets: the electroencephalogram (EEG) signal dataset and the

breast cancer (BreaKHis) dataset. These experiments provide further evidence of the model's efficiency and applicability in real-world medical scenarios.

TABLE I: Illustrate the characteristics of different loss functions used for the classification task.

Loss function ↓\ Characteristic →	Robust	Sparse	Bounded	Convex	Smooth
Hinge loss	✗	✓	✗	✓	✗
Pinball loss	✗	✗	✗	✓	✗
Truncated hinge loss	✓	✓	✓	✗	✗
Truncated pinball loss	✓	✗	✓	✗	✗
Linex loss	✗	✗	✗	✓	✓
RoBoSS loss (Proposed)	✓	✓	✓	✗	✓

II. PROPOSED WORK

In this work, we introduce a significant advancement in the realm of supervised learning: a novel loss function that embodies robustness, boundedness, sparsity, and smoothness, termed as the RoBoSS loss function (see Fig. 1e and 1f). This innovative approach represents a substantial stride in optimizing the training process of machine learning models. The equation (8) depicts the mathematical representation of the proposed RoBoSS loss function introduced in this study. The RoBoSS loss function, as put forth in this work, exhibits the subsequent properties:

- It is robust and sparse. As it determines an upper bound λ and restricts the loss to stop raising for samples with $u > 0$ after a certain margin, which enhances the robustness, and it gives a fixed loss 0 for all samples with $u \leq 0$, which adds sparsity.
- It is non-convex, smooth, and bounded.
- It has two advantageous parameters, a and λ, which are respectively referred to as the shape parameter and bounding parameter. The shape parameter (a) determines the strength of the penalty. On the other hand, the bounding parameter (λ) sets the thresholds for loss values.
- For $\lambda = 1$, when $a \to +\infty$, it converges point-wise to the "$0 - 1$" loss function.

The RoBoSS loss function addresses multiple crucial aspects of supervised learning simultaneously. By encompassing robustness, it ensures the stability of the learning process even in the presence of outliers and noise. The bounded nature of the RoBoSS loss function restricts the impact of extreme values, preventing the loss from growing unbounded. Incorporating sparsity, the RoBoSS loss function promotes the utilization of only the relevant samples, resulting in parsimonious and interpretable models. Moreover, the RoBoSS loss function is designed with a focus on smoothness, facilitating a gradual and consistent optimization process. This smoothness property promotes avoiding abrupt changes during parameter updates, leading to more stable and efficient convergence during training.

Further, through the integration of the RoBoSS loss function (8) within the SVM framework, we propose a novel SVM model termed as $\mathcal{L}_{rbss}$-SVM. For simplification, throughout

the paper we use the terminology w for $[w^\mathsf{T}, b]$ and x_i for $[x_i, 1]^\mathsf{T}$. The formulation of $\mathcal{L}_{rbss}$-SVM is given by

$$\min_{w,\xi} \quad \frac{1}{2}\|w\|^2 + \frac{\gamma}{n}\sum_{k=1}^{n}\lambda\left(1 - (a\{\xi_k\}_+ + 1)exp(-a\{\xi_k\}_+)\right),$$
$$\text{subject to} \quad y_i\left(w^\mathsf{T}\psi(x_k)\right) = 1 - \xi_k, \ \forall \ k = 1, 2, \ldots, n, \tag{10}$$

where $\{\xi_k\}_+ = \xi_k$ if $\xi_k > 0$ and 0 otherwise, $\gamma > 0$ is the penalty parameter, a and λ are the loss parameters, and $\psi(\cdot)$ is feature mapping associated with the kernel function.

Since the dual problem of $\mathcal{L}_{rbss}$-SVM is challenging to optimize due to the non-convexity of the RoBoSS loss function, in this case, we use the representer theorem [17] for the non-linear $\mathcal{L}_{rbss}$-SVM. The corresponding solution can be stated as:

$$w = \sum_{k=1}^{n}\beta_k\psi(x_k), \tag{11}$$

where $\beta = (\beta_1, \ldots, \beta_n)^\mathsf{T}$ is the coeffecient vector. Substituting (11) into (10), we obtain

$$\min_{\beta} \ f(\beta) = \sum_{k=1}^{n}\sum_{j=1}^{n}\frac{1}{2}\beta_k\beta_j\mathcal{K}(x_k, x_j)$$
$$+ \frac{\gamma}{n}\sum_{k=1}^{n}\lambda\left(1 - (a\{\xi_k\}_+ + 1)exp(-a\{\xi_k\}_+)\right), \tag{12}$$

where $\xi_k = y_k\left(\sum_{j=1}^{n}\gamma_j\mathcal{K}(x_k, x_j)\right) - 1$, and $\mathcal{K}(x_k, x_j) = \left(\psi(x_k)\cdot\psi(x_j)\right)$ is the kernel function.

III. THEORETICAL ANALYSIS OF THE PROPOSED ROBOSS LOSS FUNCTION

Assume that the training data $z = \{x_k, y_k\}_{k=1}^{n}$ is drawn independently from a probability measure $\mathcal{P}$. The probability measure $\mathcal{P}$ is defined on $X \times Y$, where $X \subseteq \mathbb{R}^m$ represents the input space and $Y = \{-1, 1\}$ is the label space. The primary objective of the classification problem is to produce a binary classifier $\mathcal{C} : X \rightarrow Y$ that minimizes the associated risks. The risk of a classifier $\mathcal{C}$ is defined by

$$\mathcal{R}(\mathcal{C}) = \int_X \mathcal{P}(y \neq \mathcal{C}(x)|x)d\mathcal{P}_X,$$

where $\mathcal{P}(y|x)$ is the conditional distribution of $\mathcal{P}$ at x and $d\mathcal{P}_X$ is the marginal distribution of $\mathcal{P}$ on x. Further, $\mathcal{P}(y|x)$ is a binary distribution, which is given by $\mathrm{Prob}(y = 1|x)$ and $\mathrm{Prob}(y = -1|x)$. For simplification, we further use $P(x)$ and $1 - P(x)$ instead of $\mathrm{Prob}(y = 1|x)$ and $\mathrm{Prob}(y = -1|x)$, respectively. Define the Bayes classifier, for $P(x) \neq 1/2$, as

$$f_{\mathcal{C}}(x) = \begin{cases} 1, & P(x) > 1/2, \\ -1, & P(x) < 1/2. \end{cases} \tag{13}$$

One can verify that the Bayes classifier minimizes the classification risk, i.e.,

$$f_{\mathcal{C}} = \arg\min_{\mathcal{C}:X\rightarrow Y}\mathcal{R}(\mathcal{C}).$$

In practice, we are seeking a function $f:X \rightarrow \mathbb{R}$ to induce a binary classifier. In this case, the classification risk becomes $\int_{X\times Y}\mathcal{L}_{mis}(yf(x))d\mathcal{P}$, where $\mathcal{L}_{mis}(yf(x))$ is the misclassification loss defined as

$$\mathcal{L}_{mis}(yf(x)) = \begin{cases} 0, & yf(x) > 0, \\ 1, & yf(x) \leq 0. \end{cases} \tag{14}$$

Therefore, minimizing the misclassification error will result in a function whose sign corresponds to the Bayes classifier. Now, for any loss function $\mathcal{L}$, the expected risk of a classifier $f:X \rightarrow \mathbb{R}$ is defined as follows,

$$\mathcal{R}_{\mathcal{L},\mathcal{P}}(f) = \int_{X\times Y}\mathcal{L}(1 - yf(x))d\mathcal{P}. \tag{15}$$

The function $f_{\mathcal{L},\mathcal{P}}$, which minimizes the expected risk over all measurable functions, can be defined as

$$f_{\mathcal{L},\mathcal{P}}(x) = \arg\min_{f(x)\in\mathbb{R}}\int_Y\mathcal{L}(1 - yf(x))\,d\mathcal{P}(y|x), \ \forall x \in X. \tag{16}$$

Then, for the proposed RoBoSS loss ($\mathcal{L}_{rbss}(\cdot)$), we can obtain Theorem III.1, demonstrating that the RoBoSS loss is classification-calibrated [18]. It is a desirable property for a loss function and requires that the minimizer of expected risk has the same sign as the Bayes classifier.

Theorem III.1. *The proposed loss $\mathcal{L}_{rbss}(u)$ is classification-calibrated, i.e., $f_{\mathcal{L}_{rbss},\mathcal{P}}$ has the same sign as the Bayes classifier.*

Proof. After simple calculation, we obtain that

$$\int_Y \mathcal{L}_{rbss}(1 - yf(x))\,d\mathcal{P}(y|x)$$
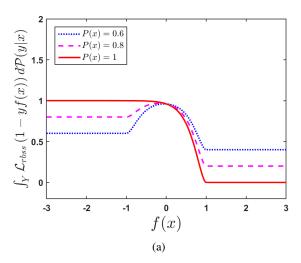$$= \mathcal{L}_{rbss}(1 - f(x))P(x) + \mathcal{L}_{rbss}(1 + f(x))(1 - P(x))$$
$$= \begin{cases} g_1(x)P(x), & f(x) \leq -1, \\ (g_1(x) - g_2(x))P(x) + g_2(x), & -1 < f(x) < 1, \\ g_2(x)(1 - P(x)), & f(x) \geq 1, \end{cases}$$

where $g_1(x) = \lambda\left(1 - (a(1 - f(x)) + 1)exp(-a(1 - f(x)))\right)$ and $g_2(x) = \lambda\left(1 - (a(1 + f(x)) + 1)exp(-a(1 + f(x)))\right)$.

Fig. 2a and 2b show the graph of $\int_Y\mathcal{L}_{rbss}(1 - yf(x))\,d\mathcal{P}(y|x)$ over $f(x)$ when $P(x) > 1/2$ and $P(x) < 1/2$, respectively. It is evident from Fig. 2 that, for $P(x) > 1/2$, the minimum value of $\int_Y\mathcal{L}_{rbss}(1 - yf(x))\,d\mathcal{P}(y|x)$ is obtained for the positive value of $f(x)$, and for $P(x) < 1/2$, the minimum value is obtained for the negative value of $f(x)$.

Hence, the proposed loss $\mathcal{L}_{rbss}(u)$ is classification-calibrated. $\square$

Further, we investigate the generalization ability of $\mathcal{L}_{rbss}$-SVM. First, we define the Rademacher complexity, which measures the complexity of a class of functions.

Definition III.1. Rademacher Complexity [19]
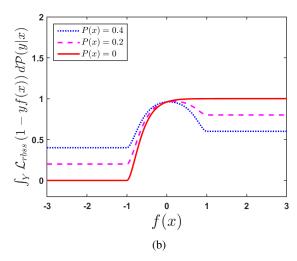Let $\mathcal{X} := \{x_1, x_2, \ldots, x_p\}$ be drawn independently from $d\mathcal{P}_X$

Fig. 2: Demonstrate the graph of $\int_Y \mathcal{L}_{rbss}\left(1 - yf(x)\right) d\mathcal{P}(y|x)$ over $f(x)$ for different values of $P(x)$. (a) For $P(x) > 1/2$ and (b) for $P(x) < 1/2$.

and $\mathcal{G}$ be a class of functions from X to $\mathbb{R}$. Define the random variable

$$\hat{R}_p(\mathcal{G}) := \mathbb{E}\left[\sup_{g \in \mathcal{G}} \left|\frac{2}{p}\sum_{k=1}^{p}\theta_k g\left(x_k\right)\right| \mid \mathcal{X}\right],$$

where $\theta_1, \theta_2, \ldots, \theta_p$ are independent discrete uniform $\{\pm1\}$-valued random variables. Then the Rademacher complexity of $\mathcal{G}$ is $R_p(\mathcal{G}) = \mathbb{E}\hat{R}_p(\mathcal{G})$.

Now, let the expected risk and empirical risk of RoBoSS loss be denoted by $\mathcal{R}(f_c)$ and $\mathcal{R}_z(f_c)$, respectively, and defined as

$$\mathcal{R}(f_c) = \int_{X \times Y} \mathcal{L}_{rbss}(1 - yf(x))d\mathcal{P},$$

$$\mathcal{R}_z(f_c) = \frac{1}{n}\sum_{k=1}^{n}\mathcal{L}_{rbss}(1 - yf(x)).$$

Then the generalization ability of $\mathcal{L}_{rbss}$-SVM can be stated as the convergence of $\mathcal{R}_z(f_c)$ to $\mathcal{R}(f_c)$ when the sample size n tends to infinity, where f_c is the classifier elicited by (10).

Theorem III.2. *Let f_c be the classifier produced by $\mathcal{L}_{rbss}$-SVM. Then for any $0 < \varepsilon < 1$, with confidence $1 - \varepsilon$, the following inequality holds*

$$\mathcal{R}(f_c) - \mathcal{R}_z(f_c) \leq \frac{4\lambda}{\sqrt{n\gamma}} + \sqrt{\frac{8\ln(1/\varepsilon)}{n}}.$$

Proof. For classifier f_c, obtained by (10) with the regularization parameter γ, we have

$$\gamma\left\|f_c^{\mathcal{L}_{rbss}}\right\|_{\mathcal{K}}^2 \leq \lambda^2,$$

which implies $\left\|f_c^{\mathcal{L}_{rbss}}\right\|_{\mathcal{K}} \leq \lambda/\sqrt{\gamma}$. Now, using theorem 8 in [19], for any $0 < \varepsilon < 1$, we have

$$\mathcal{R}\left(f_c^{\mathcal{L}_{rbss}}\right) - \mathcal{R}_{\mathbf{z}}\left(f_c^{\mathcal{L}_{rbss}}\right) \leq R_n(\mathcal{J}) + \sqrt{\frac{8\ln(1/\varepsilon)}{n}}, \quad (17)$$

where the set $\mathcal{J}$ is defined as

$$\mathcal{J} := \{j \mid j(x,y) := \phi(1 - yf(x)) - \phi(0), f \in \mathcal{J}_{\mathcal{K}},$$
$$\|f\|_{\mathcal{K}} \leq \lambda/\sqrt{\gamma}, (x,y) \in X \times Y\}.$$

Again, theorem 12 in [19] yields that

$$R_n(\mathcal{J}) \leq 2R_n\left(\mathcal{G}_\gamma\right) \text{ with}$$
$$\mathcal{G}_\gamma = \left\{f \mid f \in \mathcal{J}_{\mathcal{K}}, \|f\|_{\mathcal{K}} \leq \lambda\sqrt{\log\left(1 + \lambda^{-2}\right)/\gamma}\right\}.$$

Also from [20], we have

$$R_n\left(\mathcal{G}_\gamma\right) \leq \frac{2\lambda}{\sqrt{n\gamma}}. \quad (18)$$

Hence from (17) and (18), for any $0 < \varepsilon < 1$, we have

$$\mathcal{R}(f_c^{\mathcal{L}_{rbss}}) - \mathcal{R}_z(f_c^{\mathcal{L}_{rbss}}) \leq \frac{4\lambda}{\sqrt{n\gamma}} + \sqrt{\frac{8\ln(1/\varepsilon)}{n}}.$$

$\square$

IV. Optimization of $\mathcal{L}_{rbss}$-SVM

To solve the optimization problem (12), we adopt the framework based on the Nestrov accelerated gradient (NAG) algorithm. It is an extension of the stochastic gradient descent (SGD) method that incorporates momentum to accelerate convergence. In SGD, a small batch of samples (mini-batch) is used for each iteration during the training of a model. This approach offers several advantages, including reduced computational requirements and improved speed, particularly when dealing with large-scale problems. However, SGD has some drawbacks, such as getting stuck in local optima during its process of convergence due to the randomness of the mini-batch. To improve SGD, many researchers introduced accelerated variance in SGD [21, 22]. The momentum method [23] is a practical approach that helps SGD to accelerate in the relevant direction and dampen the oscillation. It does this by combining the update vector of the previous time step with the current update vector.

The NAG algorithm is an extension of the momentum method

that further improves convergence by incorporating a "look-ahead" mechanism [24]. It gives an approximation of the future position of the parameters and then calculates the gradient with respect to the approximate future position of the model parameters. One challenge for NAG is to choose an appropriate learning rate during the training. If the learning rate is set to a very low value, the algorithm's convergence speed becomes sluggish. On the contrary, using a high learning rate is likely to cause the algorithm to overshoot the optimal point or even fail to converge. An intuitive approach is to begin with a slightly higher learning rate and then gradually reduce it during the learning process according to a predefined schedule. Taking inspiration from the simulated annealing approach [25], we employ the exponential decay method for adjusting the learning rate as $\alpha_{new} = \alpha_{old} \exp(-\eta t)$, where η is a hyperparameter that determines the extent of the learning rate's decay at each iteration, while t represents the current iteration number.

Now, we solve (12) by employing the NAG-based framework. The method employed to solve (12) is thoroughly described in Algorithm 1. After obtaining the optimal β, the following decision function can be utilized to predict the label of a new sample x.

$$\hat{y} = \text{sign}(f(x)) = \text{sign}\left(\sum_{j=1}^{s} \beta_j \mathcal{K}(x_j, x)\right). \quad (19)$$

Algorithm 1 NAG-based algorithm to solve $\mathcal{L}_{rbss}$-SVM

Input:
The dataset: $\{x_k, y_k\}_{k=1}^{n}$, $y_k \in \{-1, 1\}$;
The parameters: Regularization parameter C, RoBoSS loss parameters λ and a, mini-batch size s, learning rate decay factor η, momentum parameter r, maximum iteration number N;
Initialize: model parameter β_0, velocity v_0, learning rate α;
Output:
The classifiers parameters: β;
1: Select s samples $\{x_k, y_k\}_{k=1}^{s}$ uniformly at random.
2: Computing ξ_k :

$$\xi_k = 1 - y_k \left(\sum_{j=1}^{s} \beta_j \mathcal{K}(x_k, x_j)\right), \ k = 1, \cdots, s; \quad (20)$$

3: Temporary update: $\widetilde{\beta}_t = \beta_t + r v_t$;
4: Compute gradient:

$$\nabla f(\beta_t) = \mathcal{K}\beta - \frac{\gamma}{s} \lambda \sum_{j=1}^{s} a^2 \xi_j \exp(-a\xi_j) y_j \mathcal{K}_j, \quad (21)$$

where $\mathcal{K}$ is the Gaussian kernel matrix.
5: Update velocity: $v_t = r v_{t-1} - \alpha_{t-1} \nabla f(\beta_t)$;
6: Update model parameter: $\beta_{t+1} = \beta_t + v_t$;
7: Update learning rate: $\alpha_{t+1} = \alpha_t \exp(-\eta t)$;
8: Update current iteration number: $t = t + 1$.
Until:
$t = N$.
Return: β_t.

V. EXPERIMENTAL RESULTS

This section discusses the results produced by the numerical experiment conducted in this study. We compare the proposed $\mathcal{L}_{rbss}$-SVM against four baseline loss function-based SVMs, namely $\mathcal{L}_{hinge}$-SVM [1], $\mathcal{L}_{pin}$-SVM [5], $\mathcal{L}_{linex}$-SVM [9], and $\mathcal{L}_{qtse}$-SVM [26].

A. Experimental setup and parameter selection

All the experiments are run utilizing MATLAB R2023a on window 10 running on a PC with configuration Intel(R) Xenon(R) Platinum 8260 CPU @ 2.30GHz with 256 GB of RAM. To map the input samples into a higher-dimensional space, the Gaussian kernel function is used. It is defined as $\kappa(x_k, x_j) = \exp\left(-\|x_k - x_j\|^2 / \sigma^2\right)$, where σ is the kernel parameter. Before training, each dataset is normalized in the interval $[-1, 1]$. For each model, the penalty parameter γ and kernel parameter σ are selected from the set $\{10^i \mid i = -6, -5, \ldots, 5, 6\}$. For $\mathcal{L}_{pin}$-SVM, the hyperparameter τ is selected from $\{0, 0.3, 0.5, 0.7, 0.9\}$. For $\mathcal{L}_{linex}$-SVM, and $\mathcal{L}_{qtse}$-SVM the range of loss parameter is taken the same as in [9] and [26], respectively. For the proposed $\mathcal{L}_{rbss}$-SVM the loss parameters a and λ are selected from the range $[0 : 0.1 : 5]$ and $[0.1 : 0.1 : 2]$, respectively. The parameters for the NAG-based algorithm are experimentally set as: (i) initial model parameter $\beta_0 = 0.01$, (ii) initial velocity $v_0 = 0.01$, (iii) initial learning rate $\alpha = 0.1$, (iv) learning decay factor $\eta = 0.1$, (v) momentum parameter $r = 0.6$, (vi) two distinct minibatch size configurations are used based on the size of the dataset: $s = 2^2$ for datasets with less than 100 samples and $s = 2^5$ for datasets with 100 or more samples, (vii) maximum iteration number $N = 1000$.

The choice of hyperparameters has a significant impact on the models' performance. In order to optimize them, we use k-fold ($k = 5$) cross-validation and grid search. In this, the dataset is randomly split into five non-overlapping subsets, one of which is designated as a test set and the other four as train sets. For each set of hyperparameters, we determined the testing accuracy for all five subsets separately. Then, for each hyperparameter set, we calculate the mean testing accuracy by taking the average of these five accuracy values. The best mean testing accuracy is chosen as the testing accuracy of the model.

The performance of the models is evaluated using the accuracy metric, which is defined as

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \times 100,$$

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively. To further analyze the model's performance, we also evaluate the rank and training time.

B. Evaluation on UCI and KEEL Datasets

Here, we discuss the experimental results on 88 real-world datasets from diverse domains downloaded from the UCI [27] and KEEL [28] repositories. Based on the sample size, we split the datasets into two categories: (D1) datasets with samples

TABLE II: The average classification accuracies, training times, and ranks of the proposed $\mathcal{L}_{rbss}$-SVM and baseline models on each 79 D1 category UCI and KEEL datasets.

Model	$\mathcal{L}_{hinge}$-SVM	$\mathcal{L}_{pin}$-SVM	$\mathcal{L}_{linex}$-SVM	$\mathcal{L}_{qtse}$-SVM	$\mathcal{L}_{rbss}$-SVM (Proposed)
Dataset samples, features	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time
acute inflammation 120,6	100±0, 0.0049	100±0, 0.0047	90±22.36, 0.0024	75±37.27, 0.0016	98.33±3.73, 0.001
balloons 16,4	53.33±50.55, 0.0079	80±18.26, 0.002	73.33±27.89, 0.0023	86.67±18.26, 0.0015	100±0, 0.0011
fertility 100,9	88±10.37, 0.0054	89±9.62, 0.0075	88±10.37, 0.0017	88±10.37, 0.0008	92±7.58, 0.0009
molec biol promoter 106,57	56.75±10.8, 0.0081	90.48±21.3, 0.0052	71.39±42.68, 0.0018	90.48±21.3, 0.0009	92.38±17.04, 0.0008
parkinsons 195,22	79.49±23.15, 0.0117	80±21.48, 0.0043	80±21.48, 0.0018	80±21.48, 0.0008	89.74±9.59, 0.0009
pittsburg bridges T OR D 102,7	86.14±13.95, 0.0078	86.14±13.95, 0.0029	86.14±13.95, 0.0028	86.14±13.95, 0.0008	92.14±5.69, 0.0008
breast cancer 286,9	70.18±44.62, 0.0047	70.18±44.62, 0.009	70.18±44.62, 0.0033	70.62±24.14, 0.0037	81.81±12.26, 0.0011
breast cancer wisc prog 198,33	76.35±8.95, 0.0036	76.35±8.95, 0.0057	77.35±8.32, 0.0047	76.35±8.95, 0.0016	78.82±7.94, 0.0011
congressional voting 435,16	62.07±3.04, 0.013	62.3±2.62, 0.0153	61.38±1.74, 0.0037	62.07±2.15, 0.0028	63.22±3.25, 0.0011
echocardiogram 131,10	75.44±11.83, 0.0042	77.01±7.85, 0.0051	77.04±7.35, 0.0036	75.58±11.04, 0.0017	85.44±9.63, 0.0011
haberman survival 306,3	73.49±8.48, 0.0046	73.49±8.48, 0.0088	73.82±8.2, 0.003	73.82±8.68, 0.003	76.11±8.71, 0.0011
hepatitis 155,19	83.23±11.27, 0.0039	83.87±7.21, 0.0035	81.94±8.1, 0.0036	81.29±11.5, 0.0017	89.03±8.1, 0.0011
horse colic 368,25	80.17±4.3, 0.0074	80.17±4.3, 0.0106	76.09±3.24, 0.0035	65.77±6.65, 0.0017	79.59±9.5, 0.0011
ionosphere 351,33	64.71±21.68, 0.0104	67.79±6.61, 0.028	81.78±9.51, 0.0045	69.61±23.45, 0.0021	72.56±4.84, 0.0012
planning 182,12	71.38±8.85, 0.0038	71.38±8.85, 0.0042	71.38±8.85, 0.0027	71.38±8.85, 0.0028	86.91±8.61, 0.0014
spect 265,22	64.15±6.67, 0.0043	65.28±6.62, 0.007	65.66±9.09, 0.0033	58.49±14.06, 0.0025	73.6±3.89, 0.0012
spectf 267,44	79.34±20.89, 0.0039	79.34±20.89, 0.0061	79.34±20.89, 0.0032	79.34±20.89, 0.0027	72.08±4.3, 0.0012
statlog heart 270,13	77.04±1.66, 0.0062	77.04±1.66, 0.0103	78.15±4.01, 0.0031	72.96±3.84, 0.0019	80.48±10.33, 0.0016
bupa or liver-disorders 345,6	71.88±3.64, 0.0074	71.88±3.64, 0.02	62.61±8.9, 0.0035	62.32±8.64, 0.003	80.37±1.66, 0.0011
cleve 297,13	78.77±7.4, 0.0078	78.77±7.4, 0.0165	76.07±5.26, 0.0036	74.36±8.95, 0.0026	69.28±3.75, 0.0014
crossplane130 130,2	70.77±8.85, 0.004	70.77±8.85, 0.0057	70±8.34, 0.003	64.62±9.96, 0.0017	80.79±4.34, 0.0012
crossplane150 150,2	62±12.16, 0.0038	62±12.16, 0.0081	67.33±7.6, 0.0029	66.67±8.82, 0.0017	74.62±5.83, 0.0012
ecoli-0-1-4-6vs5 280,6	97.5±2.71, 0.006	97.5±2.71, 0.0161	96.07±2.65, 0.0034	96.07±3.43, 0.0018	74±4.35, 0.0015
ecoli-0-1-4-7vs2-3-5-6 336,7	96.73±0.67, 0.0078	96.73±0.67, 0.0262	93.75±1.93, 0.0037	94.93±3.28, 0.0022	98.21±1.26, 0.0015
ecoli-0-1-4-7vs5-6 332,6	98.19±0.68, 0.0078	98.19±0.68, 0.0187	95.18±1.95, 0.0029	95.17±2.72, 0.0019	96.12±3.27, 0.0013
ecoli-0-1vs2-3-5 244,7	96.73±3.09, 0.0045	96.73±3.09, 0.0304	94.68±3.7, 0.0034	94.66±4.94, 0.0025	97.59±1.35, 0.0012
ecoli-0-1vs5 240,6	97.92±2.55, 0.0055	97.92±2.55, 0.0144	96.25±3.73, 0.0028	95.42±2.28, 0.0017	97.14±3.98, 0.0013
ecoli-0-2-3-4vs5 202,7	98.5±3.35, 0.004	98.5±3.35, 0.02	97.5±3.54, 0.0036	94.55±2.11, 0.003	98.75±1.14, 0.0011
ecoli-0-2-6-7vs3-5 224,7	96.44±5.79, 0.0042	96.44±5.79, 0.0112	91.99±6.39, 0.003	95.11±5.07, 0.0017	98±3.26, 0.0011
ecoli-0-3-4-6vs5 205,7	97.07±3.18, 0.004	97.07±3.18, 0.0151	95.61±4.01, 0.0031	94.63±2.04, 0.0017	96±5.53, 0.0011
ecoli-0-3-4-7vs5-6 257,7	97.66±0.89, 0.0052	97.66±0.89, 0.0129	94.95±2.6, 0.0037	94.53±2.58, 0.0029	97.56±2.44, 0.0011
ecoli-0-4-6vs5 203,6	97.05±2.67, 0.0037	97.05±2.67, 0.0206	95.57±3.61, 0.0019	94.07±3.31, 0.0017	97.65±1.64, 0.0011
ecoli-0-6-7vs3-5 222,7	96.4±3.02, 0.004	96.4±3.02, 0.0137	93.68±2.98, 0.0032	92.83±3.94, 0.0019	98.01±2.09, 0.0011
ecoli-0-6-7vs5 220,6	97.27±1.9, 0.0042	97.27±1.9, 0.011	93.64±6.31, 0.0028	94.55±4.13, 0.0015	96.87±2.99, 0.0011
ecoli0137vs26 311,7	96.15±3.3, 0.007	96.15±2.65, 0.0158	94.86±2.62, 0.0029	94.55±3.3, 0.0015	96.82±2.59, 0.0011
ecoli01vs5 240,7	98.33±1.74, 0.005	98.33±1.74, 0.015	99.17±1.14, 0.0038	98.75±1.86, 0.0015	97.12±2.35, 0.0011
ecoli3 336,7	92.85±3.87, 0.0084	92.85±3.87, 0.0267	93.75±3.23, 0.003	91.37±3.23, 0.0014	99.58±0.93, 0.0011
ecoli4 336,7	98.52±1.48, 0.0084	98.52±1.48, 0.0269	97.32±1.25, 0.0036	97.92±1.7, 0.0014	94.94±3.75, 0.0011
glass2 214,9	92.05±2.12, 0.0036	92.05±2.12, 0.0092	92.05±2.12, 0.0029	92.05±2.12, 0.0015	99.11±0.81, 0.0011
glass4 214,9	97.19±3.05, 0.005	97.19±3.05, 0.0139	96.25±2.7, 0.0034	96.27±2.64, 0.0015	93.91±3.58, 0.0011
glass5 214,9	96.73±2.07, 0.0053	96.73±2.07, 0.0184	95.79±1.95, 0.0029	96.74±2.65, 0.0015	98.14±1.95, 0.0011

TABLE II: The average classification accuracies, training times, and ranks of the proposed $\mathcal{L}_{rbss}$-SVM and baseline models on each 79 D1 category UCI and KEEL datasets (Continued).

Model Dataset samples, features	$\mathcal{L}_{hinge}$-SVM Acc. $\pm$ Std., time	$\mathcal{L}_{pin}$-SVM Acc. $\pm$ Std., time	$\mathcal{L}_{linex}$-SVM Acc. $\pm$ Std., time	$\mathcal{L}_{qtse}$-SVM Acc. $\pm$ Std., time	$\mathcal{L}_{rbss}$-SVM (Proposed) Acc. $\pm$ Std., time
haber 306,3	73.82±8.03, 0.0087	73.82±8.03, 0.0242	73.82±8.03, 0.0038	73.49±8.48, 0.0015	98.14±1.95, 0.0011
haberman 306,3	73.82±8.03, 0.0089	73.82±8.03, 0.0306	74.15±7.64, 0.0037	73.49±8.48, 0.0015	74.8±7.29, 0.0011
iono 351,33	80.91±5.76, 0.0079	80.91±5.76, 0.02	81.51±10.99, 0.0025	72.38±22.53, 0.0017	74.8±7.29, 0.0011
led7digit-0-2-4-5-6-7-8-9vs1 443,7	96.17±0.99, 0.0114	96.17±0.99, 0.0294	95.27±2.55, 0.0017	94.81±1.73, 0.0015	84.93±10.49, 0.0012
new-thyroid1 215,5	95.35±1.64, 0.0042	95.35±1.64, 0.0119	97.21±1.04, 0.0016	97.67±2.33, 0.0015	96.85±1.84, 0.0011
shuttle-6vs2-3 230,9	100±0, 0.005	100±0, 0.0176	100±0, 0.0017	100±0, 0.0015	99.53±1.04, 0.0011
votes 435,16	89.89±4.97, 0.023	90.11±5.37, 0.0316	92.41±3.69, 0.0024	85.75±7.48, 0.0015	100±0, 0.0011
wpbc 194,33	76.29±10.16, 0.0038	76.29±10.16, 0.0135	76.29±10.16, 0.0027	76.29±10.16, 0.0015	94.94±2.52, 0.0011
yeast1vs7 459,8	94.77±1.18, 0.0158	94.77±1.18, 0.0507	93.47±2.54, 0.0026	93.47±2.54, 0.0015	79.38±3.63, 0.0011
yeast2vs8 483,8	97.73±2.12, 0.0205	97.73±2.12, 0.0457	97.31±2.15, 0.0027	97.11±2.77, 0.0015	94.77±1.41, 0.0011
bank 4521,16	88.67±0.49, 3.1119	89.03±0.42, 4.9583	88.48±0.55, 0.0043	88.48±0.55, 0.0029	98.14±1.7, 0.0011
blood 748,4	76.64±13.29, 0.0687	76.64±13.29, 0.117	76.24±14.98, 0.0034	76.24±14.98, 0.0028	88.5±0.56, 0.0016
breast cancer wisc diag 569,30	79.44±3.43, 0.0138	81.54±5.8, 0.0221	84.27±4.69, 0.0038	81.69±6.92, 0.0023	78.51±11.76, 0.0012
chess krvkp 3196,36	72.3±27.33, 2.0588	75.77±23.19, 3.2068	58.7±14.93, 0.0047	75.77±23.19, 0.0023	87.42±6.62, 0.0012
credit approval 690,15	84.06±9.78, 0.0299	84.06±9.78, 0.0397	77.25±6.85, 0.003	76.38±13.81, 0.0025	75.99±23.24, 0.0014
cylinder bands 512,35	60.87±17.95, 0.0118	61.07±17.6, 0.0169	65±8.83, 0.0034	64.79±14.01, 0.0028	82.9±9.86, 0.0012
ilpd indian liver 583,9	71.35±5.09, 0.0774	71.35±5.09, 0.0235	71.35±5.09, 0.0032	71.35±5.09, 0.0028	68.32±10.01, 0.0012
mammographic 961,5	77.94±5.78, 0.0917	77.94±5.78, 0.138	73.15±2.91, 0.0034	71.08±2.65, 0.003	77.21±2.02, 0.0013
oocytes trisopterus nucleus 2f 912,25	67.99±6.85, 0.0684	67.99±6.85, 0.099	64.26±7.03, 0.0033	59.55±10.69, 0.0026	66.55±7.25, 0.0014
pima 768,8	70.58±2.36, 0.0476	70.58±2.36, 0.0641	65.24±5.69, 0.0035	65.1±5.95, 0.0026	69.66±4.88, 0.0013
monk1 556,6	51.79±3.06, 0.0164	52.15±3.1, 0.0697	51.97±3.91, 0.0015	52.5±4.21, 0.0015	53.04±5.14, 0.0013
monk3 556,6	50.72±1.55, 0.0225	50.9±1.42, 0.0476	51.44±2.22, 0.0016	52.16±3.72, 0.0015	53.06±5.3, 0.0011
checkerboard data 690,14	82.17±2.44, 0.0656	82.61±2.46, 0.0944	76.67±2.53, 0.0033	73.62±4.98, 0.0028	81.01±1.07, 0.0014
statlog australian credit 690,14	67.97±1.65, 0.0416	67.97±1.65, 0.0592	67.97±1.57, 0.0031	68.55±1.5, 0.0025	68.41±1.41, 0.0013
transfusion 748,4	77.3±12.01, 0.0323	77.3±12.01, 0.0897	76.51±14.55, 0.0027	76.24±14.98, 0.0015	78.64±10.99, 0.0012
vowel 988,10	95.54±2.14, 0.0604	95.54±2.14, 0.1765	94.43±2.42, 0.0026	93.01±5.2, 0.0017	95.95±3.23, 0.0013
yeast-0-2-5-6vs3-7-8-9 1004,8	93.22±2.26, 0.0937	93.22±2.26, 0.2803	90.73±3.25, 0.0025	90.54±3.12, 0.0016	92.43±2.06, 0.0012
yeast-0-2-5-7-9vs3-6-8 1004,8	96.22±0.9, 0.0932	96.22±0.9, 0.2532	93.92±1.59, 0.0028	93.13±2.2, 0.0017	95.72±0.44, 0.0012
yeast-0-3-5-9vs7-8 506,8	91.7±2.68, 0.0197	91.7±2.68, 0.1017	91.3±3.66, 0.0021	90.71±2.86, 0.0015	91.9±2.56, 0.0011
yeast-0-5-6-7-9vs4 528,8	92.42±1.35, 0.0216	92.42±1.35, 0.0554	90.72±1.81, 0.0028	90.34±1.82, 0.0015	93.18±1.43, 0.0012
titanic 2201,3	77.1±15.93, 0.4282	77.33±16.02, 0.5734	77.92±15.58, 0.0029	77.33±16.02, 0.0027	79.05±15.04, 0.0015
abalone9-18 731,7	95.36±3.31, 0.0326	95.36±3.31, 0.1077	94.4±4.5, 0.0042	94.95±4.5, 0.0025	95.9±3.82, 0.0014
aus 690,14	82.17±2.44, 0.0285	82.61±2.46, 0.0756	76.67±2.53, 0.0039	73.62±4.98, 0.0021	81.01±1.07, 0.0017
cmc 1473,9	69.54±18.95, 0.6173	79.99±22.08, 0.5047	81.62±20.33, 0.0032	81.69±20.36, 0.0017	81.62±20.33, 0.0014
ripley 1250,2	59.84±3.37, 0.27	59.84±3.37, 0.2029	59.84±3.37, 0.0032	59.84±3.37, 0.0017	60.4±3.12, 0.0013
yeast5 1484,8	97.57±2.03, 0.4438	97.57±2.03, 0.7824	97.03±2.45, 0.0036	97.17±2.41, 0.0017	97.78±1.3, 0.0014
ozone 2536,72	97.12±2.26, 0.3991	97.12±2.26, 0.4385	97.12±2.26, 0.0043	97.12±2.26, 0.0021	97.2±2.11, 0.002
spambase 4601,57	99.3±1.31, 96.8396	99.39±1.36, 1.6873	77.4±13.77, 0.0033	99.39±1.36, 0.0019	99.39±3.86, 0.0016
Avg. Acc. $\pm$ Avg. Std.	83.16±7.04	84.26±6.44	82.53±7.39	82.18±7.91	86.35±5.06
Avg. time	0.1304	0.1909	0.0031	0.0019	0.0012
Avg. rank	2.97	2.59	3.46	3.85	2.13

Here, Avg., Acc. and Std. are acronyms used for average, accuracy, and standard deviation, respectively.

TABLE III: The classification accuracies and training times of the proposed $\mathcal{L}_{rbss}$-SVM and baseline models on 9 D2 category UCI and KEEL datasets.

Model	$\mathcal{L}_{hinge}$-SVM	$\mathcal{L}_{pin}$-SVM	$\mathcal{L}_{linex}$-SVM	$\mathcal{L}_{qtse}$-SVM	$\mathcal{L}_{rbss}$-SVM (Proposed)
Dataset (samples, features)	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time
Musk2 (6598, 166)	80$\pm$44.72, 18.7863	84.59$\pm$34.46, 22.9542	84.59$\pm$34.46, 0.0048	81.02$\pm$17.85, 0.0023	84.59$\pm$34.46, 0.0029
Ringnorm (7400, 20)	50.5$\pm$1.29, 8.0734	50.95$\pm$0.88, 14.7853	51.15$\pm$0.62, 0.0033	51.03$\pm$1.03, 0.0019	52.22$\pm$0.9, 0.0018
Twonorm (7400, 20)	50.61$\pm$0.82, 5.0918	50.8$\pm$0.52, 28.8532	50.78$\pm$0.9, 0.0031	50.92$\pm$1.35, 0.0019	52.24$\pm$2.01, 0.0024
EEG Eye State (14980, 14)	55.12$\pm$25.92, 127.686	61.78$\pm$22.46, 192.8241	68.93$\pm$16.06, 0.0033	69.71$\pm$15.36, 0.002	71.2$\pm$13.68, 0.0017
Magic (19020,10)	82.84$\pm$9.8, 443.3522	82.88$\pm$9.72, 217.4496	65.3$\pm$25.25, 0.0043	95.16$\pm$10.82, 0.0021	95.16$\pm$33.91, 0.0023
Credit Default (30000, 23)	77.89$\pm$1.56, 247.5376	77.88$\pm$1.56, 1415.7059	77.88$\pm$1.56, 0.0062	77.88$\pm$1.56, 0.0034	77.88$\pm$1.56, 0.0069
Adult (48842, 14)	*	*	76.41$\pm$1.8, 0.0106	76.07$\pm$0.25, 0.0042	77.94$\pm$1.51, 0.0051
Connect4 (67557, 42)	*	*	75.38$\pm$3.78, 0.0118	75.38$\pm$3.78, 0.0057	75.4$\pm$3.75, 0.0082
Miniboone (130064, 50)	*	*	77.17$\pm$18.82, 0.0156	81.67$\pm$17.11, 0.0144	82.5$\pm$7.94, 0.0118
Avg Acc. $\pm$ Avg. Std.	66.16$\pm$14.02	68.15$\pm$11.6	69.73$\pm$11.47	73.2$\pm$7.68	74.35$\pm$11.08

Here, Avg., Acc. and Std. are acronyms used for average, accuracy, and standard deviation, respectively.

* Denote that the Matlab program encounters an "out of memory" error.

under or equal to 5000, and (D2) datasets with samples over 5000. There are 79 and 9 datasets in the D1 and D2 categories, respectively.

Table II presents the accuracy, training time, and rank of the models on 79 D1 category datasets. The results for the 9 D2 category datasets are presented in Table III.

C. Evaluation on Biomedical Datasets

In this section, we analyze the experimental results on publicly available biomedical datasets. Specifically, the electroencephalogram (EEG) signal dataset and the breast cancer (BreaKHis) dataset.

The EEG data [29] includes five sets: A, B, O, C, and S. Each contains 100 single-channel EEG signals that were sampled at 173.61 hertz with a duration of 23.6 seconds. The sets O and C stand for the subject's eyes open and closed signals, respectively. Sets A and B provide the EEG signal that represents the subject's interictal state. The seizure activity signal is contained in set S. The feature selection process is the same as opted in [30]. The experimental results on EEG datasets are presented in Table IV.

Further, we evaluate the models on BreaKHis dataset [31]. We used 1240 scans from the dataset at $400\times$ magnification. The scans can be classified as benign or malignant. The benign class has four subclasses: phyllodes tumor (PT), adenosis (AD), fibroadenoma (FD), and tubular adenoma (TA), with 115, 106, 237, and 130 scans, respectively. Whereas the subclasses of the malignant class are lobular carcinoma (LC), papillary carcinoma (PC), ductal carcinoma (DC), and mucinous carcinoma (MC) with 137, 138, 208, and 169 scans, respectively. For feature extraction, we employ the same process as in [32]. The experimental results on BreaKHis dataset are shown in Table V.

VI. STATISTICAL ANALYSIS OF RESULTS

To provide additional substantiation for the enhanced efficacy of the proposed $\mathcal{L}_{rbss}$-SVM model, we conducted a statistical analysis of the models. For this, we employed the Friedman test followed by the Nemenyi post hoc test to assess the relative performance of these models.

Friedman test: The Friedman test [33] is employed to statistically analyze the significance of the models. In this test, each model is ranked on each dataset separately, with the best-performing model securing rank 1, the second-best model getting rank 2, and so on. Under the null hypothesis, all the models are equivalent, i.e., the average rank of each model is equal. The Friedman statistic follows the chi-squared χ_F^2 distribution with $p - 1$ degrees of freedom (d.f.), where p denotes the number of models and is given by:

$$\chi_F^2 = \frac{12D}{p(p+1)} \left[\sum_e R_e^2 - \frac{p(p+1)^2}{4} \right], \qquad (22)$$

where D denotes the number of datasets and R_e is the mean rank of eth of the p models. The Friedman statistic is undesirably conservative and thus a better statistic is derived by Iman and Davenport [34] as:

$$F_F = \frac{(D-1)\chi_F^2}{D(p-1) - \chi_F^2}, \qquad (23)$$

which follows F distribution with $((p - 1), (p - 1)(D - 1))$ d.f.. From the statistical F-distribution table, at 5% level of significance, we find the value of $F((p - 1), (p - 1)(D - 1))$. If $F_F > F((p - 1), (p - 1)(D - 1))$, we reject the null hypothesis. In this case, substantial differences exist among the models. Table VI presents the results of the Friedman test on D1 category UCI and KEEL datasets, the EEG dataset, and the BreaKHis dataset. The outcomes demonstrate that significant differences exist among the proposed $\mathcal{L}_{rbss}$-SVM and baseline

TABLE IV: The classification accuracies, training times, and ranks of the proposed $\mathcal{L}_{rbss}$-SVM and baseline models on the EEG dataset.

Model	$\mathcal{L}_{hinge}$-SVM	$\mathcal{L}_{pin}$-SVM	$\mathcal{L}_{linex}$-SVM	$\mathcal{L}_{qtse}$-SVM	$\mathcal{L}_{rbss}$-SVM (Proposed)
Dataset	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time
BvsS-bhattacharyya100	69.5±5.97, 0.0049	70.5±8.55, 0.0051	73.5±7.83, 0.0015	55±1.77, 0.0017	77±6.47, 0.0016
BvsS-entropy100	69.5±5.97, 0.0055	70.5±8.55, 0.004	73.5±7.83, 0.0018	55±1.77, 0.0018	77±6.47, 0.0017
BvsS-roc50	72.5±3.54, 0.0051	73.5±4.87, 0.003	74.5±5.7, 0.0016	58±14.3, 0.0019	79±9.45, 0.0015
BvsS-ttest200	83.5±5.18, 0.0059	83.5±4.18, 0.0038	77±6.22, 0.0016	55±1.77, 0.0017	81±7.42, 0.0018
BvsS-wilcoxon100	76±2.24, 0.0048	76±2.24, 0.0034	79.5±4.11, 0.0019	55±1.77, 0.0017	81.5±6.27, 0.0013
BvsS-wilcoxon200	81.5±8.4, 0.0052	81.5±8.4, 0.0035	80±7.07, 0.0018	55±1.77, 0.0019	82.5±5.86, 0.0013
AvsS-bhattacharyya100	71.5±4.18, 0.0057	71.5±4.18, 0.0038	68±4.47, 0.0018	56±1.37, 0.0018	76±8.59, 0.0013
AvsS-bhattacharyya200	78±7.58, 0.0049	78±7.58, 0.004	70±8.29, 0.0016	55±1.77, 0.0018	79.5±4.11, 0.0013
AvsS-entropy100	71.5±4.18, 0.0052	71.5±4.18, 0.0037	68±4.47, 0.0018	56±1.37, 0.0017	76±8.59, 0.0012
AvsS-entropy200	78±7.58, 0.0049	78±7.58, 0.004	70±8.29, 0.0019	55±1.77, 0.0018	79.5±4.11, 0.0013
AvsS-ttest100	79±6.75, 0.0051	80±6.37, 0.0034	74.5±9.75, 0.0017	55±1.77, 0.0018	81±5.76, 0.0014
CvsB-roc200	77±4.11, 0.005	77±4.11, 0.0036	77.5±7.29, 0.0017	55±1.77, 0.0018	84.5±3.26, 0.0013
CvsB-wilcoxon200	84.5±4.11, 0.0047	86±5.18, 0.0031	86±8.4, 0.0015	55±1.77, 0.0017	90±5, 0.0013
CvsA-entropy200	77±7.58, 0.0056	77±7.58, 0.004	74±10.55, 0.0016	55±1.77, 0.0017	80±3.06, 0.0012
CvsA-entropy50	67.5±9.68, 0.0053	67.5±9.68, 0.0034	69±6.75, 0.0017	55±1.77, 0.0019	72±12.67, 0.0012
CvsA-roc150	77±5.42, 0.0051	77.5±3.95, 0.0039	79.5±4.81, 0.0016	55±1.77, 0.0018	85±7.29, 0.0012
CvsA-roc50	71.5±3.35, 0.0049	71.5±3.35, 0.0043	73.5±8.77, 0.0018	55±1.77, 0.0018	77±5.42, 0.0015
CvsA-ttest100	76.5±7.83, 0.0051	76.5±7.83, 0.0036	77±4.81, 0.0016	55±1.77, 0.0021	82±7.79, 0.0012
CvsA-ttest150	76.5±6.75, 0.0056	76.5±5.76, 0.0033	78.5±4.54, 0.0017	55±1.77, 0.0022	84±2.85, 0.0013
CvsA-ttest200	79.5±5.97, 0.005	79.5±5.97, 0.0036	79±3.79, 0.0016	55±1.77, 0.0022	85±5.59, 0.0013
CvsA-ttest50	74.5±7.79, 0.0051	74.5±7.79, 0.0039	75±9.01, 0.0017	55±9.84, 0.0021	79.5±5.42, 0.0022
CvsA-wilcoxon50	78.5±5.18, 0.0049	78.5±4.87, 0.0032	79.5±6.94, 0.0018	56±11.81, 0.002	84±6.98, 0.0016
CvsS-bhattacharyya100	60.5±6.94, 0.005	60.5±6.94, 0.0035	63.5±9.12, 0.0017	55±1.77, 0.0021	72±11.1, 0.0017
CvsS-bhattacharyya150	66.5±7.83, 0.0051	66.5±7.83, 0.0033	68.5±9.78, 0.0016	55±1.77, 0.0026	72±2.74, 0.0012
CvsS-entropy100	60.5±6.94, 0.0048	60.5±6.94, 0.0038	63.5±9.12, 0.0016	55±1.77, 0.0024	72±11.1, 0.0012
CvsS-entropy150	66.5±7.83, 0.0052	66.5±7.83, 0.0037	68.5±9.78, 0.0017	55±1.77, 0.0024	72±2.74, 0.0013
CvsS-ttest150	68±11.24, 0.0049	68±11.24, 0.0035	71±6.52, 0.0017	55±1.77, 0.0022	76.5±7.62, 0.0012
CvsS-ttest200	69.5±8.18, 0.0047	69.5±8.18, 0.0036	74.5±5.97, 0.0015	55±1.77, 0.0021	76.5±6.02, 0.0012
CvsS-wilcoxon100	64.5±8.91, 0.0048	64.5±7.79, 0.0031	68±10.37, 0.0017	55±1.77, 0.0021	73.5±5.48, 0.0013
OvsB-roc150	82±2.74, 0.005	82±2.74, 0.0037	79±1.37, 0.0016	55±1.77, 0.0023	86.5±3.79, 0.0013
OvsB-ttest50	71±4.87, 0.0051	73±6.22, 0.0029	77.5±7.91, 0.0019	55±1.77, 0.0021	81±8.02, 0.0014
OvsB-wilcoxon150	82±2.74, 0.0047	82±2.74, 0.0037	79±1.37, 0.0017	55±1.77, 0.0022	86.5±3.79, 0.0013
Avg Acc. $\pm$ Avg. Std.	73.8±6.17	74.05±6.29	74.06±6.91	55.19±2.7	79.42±6.28
Avg. time	0.0051	0.0036	0.0017	0.002	0.0014
Avg. rank	3.25	2.75	3	5	1

Here, Avg., Acc. and Std. are acronyms used for average, accuracy, and standard deviation, respectively.

TABLE V: The classification accuracies, training times, and ranks of the proposed $\mathcal{L}_{rbss}$-SVM and baseline models on the BreaKHis dataset.

Model	$\mathcal{L}_{hinge}$-SVM	$\mathcal{L}_{pin}$-SVM	$\mathcal{L}_{linex}$-SVM	$\mathcal{L}_{qtse}$-SVM	$\mathcal{L}_{rbss}$-SVM (Proposed)
Dataset	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time	Acc. $\pm$ Std., time
ADvsDC	66.87±4.75, 0.005	66.87±4.75, 0.0169	66.24±4.16, 0.0029	66.24±4.16, 0.0011	66.88±4.08, 0.0021
ADvsLC	57.21±6.49, 0.0033	57.21±6.49, 0.0216	58.89±7.25, 0.0036	57.61±5.18, 0.0011	62.59±5.98, 0.0017
ADvsMC	61.82±7.27, 0.0045	62.18±6.97, 0.0147	61.45±6.35, 0.0038	61.45±6.35, 0.0011	63.27±4.15, 0.0017
ADvsPC	56.54±6.82, 0.0045	57.76±4.97, 0.0174	57.76±4.97, 0.0031	57.36±4.69, 0.0011	61.91±5.68, 0.0021
FDvsDC	53.26±2.04, 0.0104	53.26±2.04, 0.0318	56.63±2.33, 0.0035	53.26±2.04, 0.0013	60±4.47, 0.0018
FDvsLC	66.84±3.76, 0.0077	66.84±3.76, 0.0128	66.3±5.85, 0.0037	63.37±3.51, 0.0012	66.57±5.02, 0.0016
FDvsMC	58.62±2.23, 0.0082	58.62±2.23, 0.0553	60.6±3.73, 0.0036	58.37±2.21, 0.0012	62.56±4.74, 0.0017
FDvsPC	63.2±3.96, 0.0071	63.2±3.96, 0.0281	64±3.89, 0.0038	63.2±3.96, 0.0011	66.93±5.45, 0.0016
PTvsDC	64.73±5.35, 0.0053	64.73±5.35, 0.017	64.41±4.85, 0.0034	64.41±4.85, 0.0012	66.26±2.45, 0.0015
PTvsLC	55.11±9.65, 0.0046	56.72±6.05, 0.0105	59.11±4.65, 0.0034	58.31±4.94, 0.0011	63.05±7.25, 0.0014
PTvsMC	59.5±6.39, 0.0045	59.5±6.39, 0.0234	59.5±6.39, 0.0036	59.5±6.39, 0.0011	62.31±4.44, 0.0014
PTvsPC	54.97±8.45, 0.0047	57.32±4.51, 0.0837	57.32±4.51, 0.0033	57.71±4.75, 0.0011	60.48±9.87, 0.0013
TAvsDC	64.21±6.99, 0.0065	64.21±6.99, 0.0473	61.55±5.54, 0.0036	61.55±5.54, 0.0011	63.64±5.92, 0.0016
TAvsLC	51.28±8.39, 0.0045	56.56±4.32, 0.0374	59.94±7.93, 0.0033	56.56±4.32, 0.0012	64.05±1.49, 0.0014
TAvsMC	59.85±6.71, 0.004	60.19±6.4, 0.0167	56.51±6.01, 0.0035	56.51±6.01, 0.0011	59.2±5.95, 0.0017
TAvsPC	54.47±6.54, 0.0053	56.35±4.15, 0.0107	56.35±4.15, 0.0078	58.23±4.76, 0.0011	62.3±3.51, 0.0018
Avg. Acc. $\pm$ Avg. Std.	59.28±5.99	60.09±4.96	60.41±5.16	59.6±4.6	63.25±5.03
Avg. time	0.0056	0.0278	0.0037	0.0011	0.0016
Avg. rank	3.59	2.97	3.22	3.84	1.37

Here, Avg., Acc. and Std. are acronyms used for average, accuracy, and standard deviation, respectively.

TABLE VI: Illustrate the results of the Friedman test on D1 category UCI and KEEL datasets, the EEG dataset, and the BreaKHis dataset.

Dataset	p	D	χ_F^2	F_F	$F((p-1),(p-1)(D-1))$	Significant difference (As per Friedman test)
D1 category dataset	5	79	58.776	17.823	2.4	Yes
EEG dataset	5	32	104	134.333	2.45	Yes
BreaKHis dataset	5	16	23.679	8.809	2.53	Yes

TABLE VII: Differences in the rankings of the proposed $\mathcal{L}_{rbss}$-SVM model against baseline models on D1 category UCI and KEEL datasets.

Model	Average rank	Rank difference	Significant difference (As per Nemenyi post hoc test)
$\mathcal{L}_{hinge}$-SVM	2.97	0.84	Yes
$\mathcal{L}_{pin}$-SVM	2.59	0.46	No
$\mathcal{L}_{linex}$-SVM	3.46	1.33	Yes
$\mathcal{L}_{qtse}$-SVM	3.85	1.72	Yes
$\mathcal{L}_{rbss}$-SVM (Proposed)	2.13	-	N/A

models.

Nemenyi post hoc test: In Nemenyi post hoc test [35], all models are compared pairwise. The performance of the two models is substantially different if the corresponding mean ranks differ by a certain threshold value (critical difference, $C.D.$). If the difference between comparing models mean ranks exceeds $C.D.$, the model with a higher mean rank is statistically better than the model with a lower mean rank. The $C.D.$ is calculated as:

$$C.D. = q_\alpha \sqrt{\frac{p(p+1)}{6D}}, \qquad (24)$$

where q_α are based on the studentized range statistic divided by $\sqrt{2}$ and called critical value for the two-tailed Nemenyi test. At 5 % level of significance, we can simply calculate that the values of $C.D.$ for D1 category UCI and KEEL datasets, the EEG dataset, and the BreaKHis dataset are 0.69, 1.08, and 1.52, respectively. Tables VII, VIII, and IX present the results of the Nemenyi post hoc test on D1 category UCI and KEEL datasets, the EEG dataset, and the BreaKHis dataset, respectively.

TABLE VIII: Differences in the rankings of the proposed $\mathcal{L}_{rbss}$-SVM model against baseline models on the EEG dataset.

Model	Average rank	Rank difference	Significant difference (As per Nemenyi post hoc test)
$\mathcal{L}_{hinge}$-SVM	3.25	2.25	Yes
$\mathcal{L}_{pin}$-SVM	2.75	1.75	Yes
$\mathcal{L}_{linex}$-SVM	3	2	Yes
$\mathcal{L}_{qtse}$-SVM	5	4	Yes
$\mathcal{L}_{rbss}$-SVM (Proposed)	1	-	N/A

TABLE IX: Differences in the rankings of the proposed $\mathcal{L}_{rbss}$-SVM model against baseline models on the BreaKHis dataset.

Model	Average rank	Rank difference	Significant difference (As per Nemenyi post hoc test)
$\mathcal{L}_{hinge}$-SVM	3.59	2.22	Yes
$\mathcal{L}_{pin}$-SVM	2.97	1.6	Yes
$\mathcal{L}_{linex}$-SVM	3.22	1.85	Yes
$\mathcal{L}_{qtse}$-SVM	3.84	2.47	Yes
$\mathcal{L}_{rbss}$-SVM (Proposed)	1.37	-	N/A

VII. CONCLUSIONS AND FUTURE WORK

In conclusion, this paper introduced a novel and innovative loss function, RoBoSS, designed to address critical challenges in supervised learning paradigms. The RoBoSS loss function is characterized by its robustness, boundedness, sparsity, and smoothness, making it a promising tool for enhancing the performance of various machine learning tasks. The theoretical analysis of the RoBoSS loss function demonstrates its remarkable properties, including classification-calibration and a rigorous generalization error bound. These theoretical insights establish RoBoSS as a reliable choice for constructing robust models in supervised learning scenarios. Furthermore, by incorporating the RoBoSS loss function into the SVM framework, we proposed a novel $\mathcal{L}_{rbss}$-SVM model. This new model not only inherits the well-known strengths of traditional SVM but also significantly bolsters their robustness and performance. The empirical results on a diverse range of datasets, including UCI, KEEL, EEG, and breast cancer datasets, decisively support the effectiveness of the proposed $\mathcal{L}_{rbss}$-SVM model.

In the future, the RoBoSS loss function can be integrated into various other supervised learning algorithms, thereby expanding its applicability and unveiling its efficacy across diverse domains. The code of the proposed model is publicly available at https://github.com/mtanveer1/RoBoSS.

REFERENCES

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[2] V. Vapnik, *The nature of statistical learning theory*. Springer Science & Business Media, 1999.

[3] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, 1995.

[4] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theoretical Computer Science*, vol. 209, no. 1-2, pp. 237–260, 1998.

[5] X. Huang, L. Shi, and J. A. Suykens, "Support vector machine classifier with pinball loss," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 984–997, 2013.

[6] M. Tanveer, S. Sharma, R. Rastogi, and P. Anand, "Sparse support vector machine with pinball loss," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 2, p. e3820, 2021.

[7] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, pp. 293–300, 1999.

[8] P. L. Bartlett and M. H. Wegkamp, "Classification with a reject option using a hinge loss." *Journal of Machine Learning Research*, vol. 9, no. 8, 2008.

[9] Y. Ma, Q. Zhang, D. Li, and Y. Tian, "Linex support vector machine for large-scale classification," *IEEE Access*, vol. 7, pp. 70 319–70 331, 2019.

[10] W. Zhu, Y. Song, and Y. Xiao, "Support vector machine classifier with huberized pinball loss," *Engineering Ap-*

plications of Artificial Intelligence, vol. 91, p. 103635, 2020.

[11] L. Zhao, M. Mammadov, and J. Yearwood, "From convex to nonconvex: a loss function analysis for binary classification," in *2010 IEEE International Conference on Data Mining Workshops*.   IEEE, 2010, pp. 1281–1288.

[12] Y. Wu and Y. Liu, "Robust truncated hinge loss support vector machines," *Journal of the American Statistical Association*, vol. 102, no. 479, pp. 974–983, 2007.

[13] J. P. Brooks, "Support vector machines with the ramp loss and the hard margin loss," *Operations Research*, vol. 59, no. 2, pp. 467–479, 2011.

[14] X. Shen, G. C. Tseng, X. Zhang, and W. H. Wong, "On ψ-learning," *Journal of the American Statistical Association*, vol. 98, no. 463, pp. 724–734, 2003.

[15] L. Yang and H. Dong, "Support vector machine with truncated pinball loss and its application in pattern recognition," *Chemometrics and Intelligent Laboratory Systems*, vol. 177, pp. 89–99, 2018.

[16] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[17] F. Dinuzzo and B. Schölkopf, "The representer theorem for hilbert spaces: a necessary and sufficient condition," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[18] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006.

[19] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 463–482, 2002.

[20] S. Mendelson, "A few notes on statistical learning theory," in *Advanced Lectures on Machine Learning: Machine Learning Summer School 2002 Canberra, Australia, February 11–22, 2002 Revised Lectures*. Springer, 2003, pp. 1–40.

[21] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[22] H. Yuan and T. Ma, "Federated accelerated stochastic gradient descent," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5332–5344, 2020.

[23] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.

[24] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence o (1/k2)," in *Dokl. Akad. Nauk. SSSR*, vol. 269, no. 3, 1983, p. 543.

[25] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[26] X. Zhao, S. Fu, Y. Tian, and K. Zhao, "Asymmetric and robust loss function driven least squares support vector machine," *Knowledge-Based Systems*, vol. 258, p. 109990, 2022.

[27] D. Dua, C. Graff *et al.*, "UCI machine learning repository," 2017.

[28] J. Derrac, S. Garcia, L. Sanchez, and F. Herrera, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Mult. Valued Logic Soft Comput*, vol. 17, 2015.

[29] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Physical Review E*, vol. 64, no. 6, p. 061907, 2001.

[30] M. Ganaie, A. Kumari, A. K. Malik, and M. Tanveer, "Eeg signal classification using improved intuitionistic fuzzy twin support vector machines," *Neural Computing and Applications*, pp. 1–17, 2022, https://doi.org/10.1007/s00521-022-07655-x.

[31] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "A dataset for breast cancer histopathological image classification," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 7, pp. 1455–1462, 2015.

[32] C. Gautam, P. K. Mishra, A. Tiwari, B. Richhariya, H. M. Pandey, S. Wang, M. Tanveer, and for the Alzheimer's Disease Neuroimaging Initiative, "Minimum variance-embedded deep kernel regularized least squares method for one-class classification and its applications to biomedical data," *Neural Networks*, vol. 123, pp. 191–216, 2020.

[33] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.

[34] R. L. Iman and J. M. Davenport, "Approximations of the critical region of the fbietkan statistic," *Communications in Statistics-Theory and Methods*, vol. 9, no. 6, pp. 571–595, 1980.

[35] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.