AIMS Mathematics, 11(5): 13683–13709.
DOI: 10.3934/math.2026564
Received: 30 January 2026
Revised: 20 April 2026
Accepted: 28 April 2026
Published: 15 May 2026

https://www.aimspress.com/journal/Math

Research article

Generalization analysis of tuning-free, Markov-ensemble SVM with distributed applications

Hongwei Jiang[1], **Yujing Yang**[1], **Bin Zou**[2,*] **and Jie Xu**[3]

[1] School of Science, Shenyang University of Technology, Shenyang 110870, China
[2] Faculty of Mathematics and Statistics, Hubei Key Laboratory of Applied Mathematics, Hubei University, Wuhan 430062, China
[3] Faculty of Computer Science, Hubei University, Wuhan 430062, China

* **Correspondence:** Email: zoubin0502@hubu.edu.cn.

Abstract: Although support vector machine (SVM) is an important algorithm, known hyperparameter tuning methods are typically time-consuming and susceptible to the influence of noise samples in large datasets. In particular, the selection of SVM hyperparameters is especially challenging in distributed learning. Therefore, this paper proposed a novel linear kernel SVM based on non-hyperparameter tuning. Its core idea was to train multiple SVM models using different regularization hyperparameters, and then integrate them into a final SVM model. To further increase the diversity of the resulting SVM models, Markov sampling was employed to generate different training subsets prior to training each SVM model. This paper derived the SVM based on non-hyperparameter tuning (SNHT) algorithm and proved its consistency. As an application, SNHT was applied to distributed learning. The performance of SNHT was validated through experiments on benchmark datasets.

Keywords: SVM; non-hyperparameter tuning; Markov sampling; generalization bounds; distributed learning
Mathematics Subject Classification: 65C40, 68Q32

1. Introduction

Support vector machine (SVM) [1] is an important learning algorithm. It not only exhibits strong learning performance and is widely applied in the field of machine learning, such as in cyber-attack detection [2], medical disease research [3], and numerous other domains [4–8], but also possesses robust theoretical results regarding consistency and convergence rates. For example, Zhang [9] investigated how to approach the optimal Bayesian error rate of SVM as closely as possible. Steinwart [10] combined techniques from stochastic processes, approximation theory and functional

analysis to prove the universal consistency of SVMs. Chen et al. [11] provided probably approximately correct (PAC) error analysis for SVM classification algorithm and q-norm soft-margin classifier. Steinwart et al. [12] studied the learning ability of SVM for α-mixture processes. Xu et al. [13] considered the generalization issue of SVM with Markov resampling. All of the theoretical studies mentioned above are typically based on kernel methods, which map the original feature space to a new functional space and transform the inner product operation of vectors in the new functional space into the computation of the kernel function. The main drawback of SVM using kernel techniques is that they result in high training and classification costs [14]. Since the kernel function of an SVM requires all training samples, its computational complexity is proportional to the square of the training set size. The classification cost is relatively high. As the kernel function of an SVM requires each support vector, the number of support vectors may be very large. Furthermore, due to the need for a large number of training samples and the requirement for high-speed processing in online prediction applications, kernel SVM has not been widely adopted [15]. Zhang [9] pointed out that any nonlinear function can be approximated by a linear model with appropriate (nonlinear) features. Consequently, many methods aim to approximate feature vectors using linear SVM. For example, Chang et al. [16] applied linear SVM to classification problems by explicitly constructing a low-order polynomial feature space. Litayem et al. [17] reduced the size of input vectors by constructing locally sensitive hashes and applying these to accelerate the prediction phase of linear SVM. Kafai and Eshghi [14] proposed the CROification algorithm by combining linear SVM with CRO (concomitant rank order) feature map. Jiao et al. [18] demonstrated that the proximal gradient method exhibits a linear convergence rate in linear sparse SVM.

Hyperparameter selection is a central issue in model development, aiming to strike an optimal balance between model complexity, data features, and computational resources. Neglecting hyperparameter tuning may result in models performing far below their potential, or even failing entirely. Therefore, whether through empirical heuristic methods or systematic automated search methods, careful hyperparameter selection remains a crucial step in building efficient and robust models. For linear SVMs, the hyperparameter is the regularization parameter. Methods such as cross-validation [19] and genetic algorithms [20] are commonly used to tune the optimal hyperparameter, ensuring that the SVM achieves optimal performance. Anguita et al. [21] improved the cross-validation method and proposed a sample-based SVM model selection method, which allows the training set to be utilized for both classifier training and hyperparameter tuning. Hu et al. [22] treated the SVM parameter selection problem as a composite optimization problem and utilized chaotic optimization to search for SVM parameters without needing to consider the SVM's dimension and complexity. Popov and Sautin [23] analyzed the impact of different grid types on SVM hyperparameter selection. Zhai et al. [24] proposed a global and efficient hyperparameter optimization scheme for robust SVM based on solution paths. Shao et al. [25] optimized the hyperparameters of SVMs using the Polar Lights Optimization (PLO) algorithm. Furthermore, Yuan et al. [26] proposed the Moose ox Optimizer (MO), a novel meta-heuristic optimization algorithm that has demonstrated excellent optimization performance on benchmark functions and engineering problems, providing an efficient solution for hyperparameter tuning of models such as SVM.

The main drawbacks of the aforementioned hyperparameter selection methods can be summarized as follows. First, these methods are typically time-consuming when dealing with large training datasets, implying that they are not suitable for scenarios requiring rapid prediction. Second, the

quality of big data is often low. In particular, noisy samples can interfere with SVM hyperparameter selection, thereby affecting model performance. In addition, due to the rapid growth in data volumes, big data is frequently stored using distributed storage methods. This implies that hyperparameter selection for SVM also faces more severe challenges. Therefore, this article explores linear SVM and proposes a new linear SVM based on non-hyperparameter tuning (SNHT). The core idea of this paper is to integrate multiple independently trained hyperparameter models through a weighted ensemble approach, thereby reducing reliance on any single parameter. By leveraging model diversity, this approach reduces variance and bias, balances overfitting and underfitting, and enhances the model's robustness against noisy data and outliers. Specifically, SNHT trains multiple linear SVM base models with different regularization hyperparameters, then employs AdaBoost weights to integrate them into a final model. To increase the diversity of these base models, we apply a Markov resampling method, which draws different sets of Markov chain samples from the given data based on information obtained from the previous model. To comprehensively investigate the properties of SNHT, we first establish a generalization error bound for SNHT and prove the convergence rate of the SNHT algorithm. As an application, we utilize SNHT to address distributed classification learning. The main contributions are summarized below.

• We propose an SNHT algorithm based on non-hyperparameter tuning. It constructs multiple base models with fixed regularization parameters and extracts information-rich subsets from the full training dataset, thereby achieving robust performance against noise.

• Theoretically, we establish a generalization bound for the proposed SNHT algorithm based on Markov chain samples and obtain a fast convergence rate.

• Experiments on real-world datasets validate the theoretical conclusions in this paper and demonstrate that the SNHT algorithm exhibits excellent noise resistance.

The rest of this paper is organized as follows. Section 2 presents the relevant definitions and notation. Section 3 describes the new SNHT algorithm. Section 4 analyzes the generalization bound of SNHT. Section 5 presents the experimental study of the SNHT algorithm. Section 6 offers some discussion and applies the proposed algorithm to distributed learning. Finally, Section 7 concludes the paper.

2. Preliminaries

In this section, some related concepts and notations used throughout this article are given.

2.1. SVM algorithm

Suppose that the random variable $Z = X \times Y$ is drawn from a metric space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ with a distribution ρ. Let $\mathcal{X} \subset \mathbb{R}^d$ be a compact feature vector space and the corresponding output space be defined as $\mathcal{Y} = \{-1, 1\}$. Define the sign function as $\text{sgn}(h) = 1$ if $h \geq 0$ and $\text{sgn}(h) = -1$ otherwise. Given a sample set $S = \mathbf{z} = z_i = (x_i, y_i)_{i=1}^n$, the SVM classifier $\text{sgn}(h)$ is determined by minimizing the following regularized objective over the hypothesis space $\mathcal{H}_K$:

$$h_{\mathbf{z},\lambda} = \arg\min_{h \in \mathcal{H}_K} \{\mathcal{R}_S(h) + \lambda \|h\|_K^2\}, \tag{2.1}$$

where λ is a regularization hyperparameter, $\mathcal{R}_S(h) = \frac{1}{n}\sum_{i=1}^n \ell(h, z_i)$ is the empirical risk, and $\ell(h, z) = (1 - yh(x))_+$ is the hinge loss [1]. The expectation risk is defined as $\mathcal{R}(h) = \mathbb{E}[\ell(h, z)]$. Here, $\mathcal{H}_K$ is

a linear function hypothesis space. In this paper, $\mathcal{H}_K$ is a reproducing kernel Hilbert space (RKHS) generated by a Mercer kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ [27]. Specifically, for the linear kernel, $\mathcal{H}_K$ is a linear function hypothesis space [28], and $\|h\|_K = \|h\|_\rho$, where $\|\cdot\|_\rho$ is the L_2-distance induced by the marginal distribution ρ_x. Let $\kappa = \sup_{x\in\mathcal{X}} \sqrt{K(x,x)}$, and let $C(\mathcal{X})$ denote the space of continuous functions on $\mathcal{X}$ with the norm $\|\cdot\|_\infty$. Then, the above reproducing property tells us that $\|h\|_\infty \leq \kappa\|h\|_K$, $\forall h \in \mathcal{H}_K$ [28].

Different from the classical SVM, in this paper SVM is used as a base classification model with a given regularization parameter. The training samples for each base classifier are uniformly ergodic Markov chain (u.e.M.c.) examples. The definition of u.e.M.c. is as follows.

2.2. U.e.M.c.

Let $(\mathcal{Z}, \mathcal{B})$ be a measurable space. A Markov chain is a sequence of random variables $\{Z_t\}_{t\geq 1}$ with the transition probabilities $\mathrm{P}^m(\mathcal{A} \mid z_i)$ defined as follows: for $\mathcal{A} \in \mathcal{B}$, $z_i \in \mathcal{Z}$,

$$\mathrm{P}^m(\mathcal{A} \mid z_i) := \mathrm{P}\{Z_{m+i} \in \mathcal{A} \mid Z_j, j < i, Z_i = z_i\}.$$

Here, $\mathrm{P}^m(\mathcal{A} \mid z_i)$ denotes the transition probability that the state z_{m+i} will belong to the set $\mathcal{A}$ after m time steps, starting from the initial state z_i at time i. The fact that the transition probability does not depend on the values of Z_j prior to time i is the Markov property. That is, $\mathrm{P}^m(\mathcal{A} \mid z_i) = \mathrm{P}\{Z_{m+i} \in \mathcal{A} \mid Z_i = z_i\}$. This is expressed as "given the present, the future is conditionally independent of the past". For two probabilities ν_1, ν_2 on the space $(\mathcal{Z}, \mathcal{B})$, the total variation distance between them is defined as

$$\| \nu_1 - \nu_2 \|_{TV} = \sup_{\mathcal{A}\in\mathcal{B}} | \nu_1(\mathcal{A}) - \nu_2(\mathcal{A}) | .$$

Thus, the definition of u.e.M.c. can be stated as follows [13].

Definition 2.1. *[13] A Markov chain $\{Z_t\}_{t\geq 1}$ is uniformly ergodic if there are two constants $0 < \gamma_0 < \infty$ and $0 < \varrho < 1$ such that*

$$\| \mathrm{P}^m(\cdot \mid z) - \pi(\cdot) \|_{TV} \leq \gamma_0 \varrho^m, \forall m \geq 1, m \in \mathbb{N},$$

where $\pi(\cdot)$ is the stationary distribution of $\{Z_t\}_{t\geq 1}$.

Remark 2.1. *If the state space of a Markov chain $\{Z_t\}_{t\geq 1}$ is finite, and the transition probability between any two states is positive, then the Markov chain $\{Z_t\}_{t\geq 1}$ is u.e.M.c. [27].*

3. Algorithm and computational complexity

In this section, we present a new linear kernel SVM algorithm based on non-hyperparameter tuning and analyze its computational complexity.

3.1. SNHT algorithm

Given a training set $\mathbf{z}$ of size n, let T be the number of base models cand $\Lambda = \{\lambda_t\}_{t=1}^T$ be the set of regularization hyperparameters corresponding to each base model. The proposed SNHT algorithm can then be described as follows.

Algorithm 1 SNHT

Input: $\mathbf{z}$, T, Λ
Output: $f_{\mathbf{z},\lambda} = \sum_{t=1}^{T} \alpha_t h_{\mathbf{z},\lambda_t}$

Draw randomly samples $S_0 = \{z_i\}_{i=1}^{N_0}$ from $\mathbf{z}$, train S_0 by FLD algorithm and obtain a classification discriminant $h_{\mathbf{z},\lambda_0}$, draw randomly a sample z from $\mathbf{z}$ and $z_1 \leftarrow z$, let $t \leftarrow 1$
while $t \leq T$ **do** $i \leftarrow 1$, $N_t \leftarrow 0$
  while $i \leq n$ **do** $i \leftarrow i + 1$
    Draw another sample z_i from $\mathbf{z}$,
    $p_t^i \leftarrow \min\{1, e^{-\ell(h_{\mathbf{z},\lambda_{t-1}}, z_i)}/e^{-\ell(h_{\mathbf{z},\lambda_{t-1}}, z_{i-1})}\}$
    if $p_t^i \equiv 1$ and $y_i y_{i-1} = 1$ **then**
      $p_t^i \leftarrow e^{-y_i h_{\mathbf{z},\lambda_{t-1}}}/e^{-y_{i-1} h_{\mathbf{z},\lambda_{t-1}}}$
    end if
    if rand(1) $< p_t^i$ **then**
      $S_t \leftarrow z_i$, $N_t = N_t + 1$
    else
      z_i was refused
    end if
  end while
  Obtain Markov chain $S_t = \{z_i\}_{i=1}^{N_t}$, train S_t by Algorithm (2.1) with λ_t and obtain a model $h_{\mathbf{z},\lambda_t}$. Calculate
  $e_t \leftarrow \mathrm{P}(Y \neq \mathrm{sgn}(h_{\mathbf{z},\lambda_t}(X)) \mid \mathbf{z})$, $\alpha_t \leftarrow (1/2) * \log((1 - e_t)/e_t)$, $t \leftarrow t + 1$
end while
For $t = 1$ to T do
  $w = \sum \alpha_t$, $\alpha_t = \alpha_t / w$

Remark 3.1. *(1) Many experiments of machine learning indicate that the noise examples not only lead to an increase in the amount of storage space, but also affect the accuracy of learning. According to the statistical learning theory [1], we know that the most "important" examples for classification problems are the examples close to the interface of two classes data. Therefore, in this article we introduce the idea of Markov sampling to sample a small number of training examples from this given data and then these Markov chain samples are used to train the base classifiers.*

(2) The Markov sampling method proposed in the SNHT is inspired by the Markov chain Monte Carlo (MCMC) method [29]. MCMC usually calculates transition probability based on the distribution characteristics of training samples or based on a randomly fixed model. In this paper, the transition probabilities p_t^i $(1 \leq t \leq T)$ used to generate examples in S_t are constructed based on the previous model h_{t-1}. The model and sampling distribution are updated iteratively with each other, forming a dynamic distribution, which is different from the conventional MCMC method. In addition, the preliminary model h_0 is constructed using the Fisher Linear Discriminant (FLD) [30] algorithm, since it requires no manual tuning of hyperparameters (such as regularization parameters, learning rate, etc.). This is consistent with the idea of avoiding hyperparameter tuning in this paper, and also distinguishes our method from the sampling approach in [13].

(3) For $1 \leq t \leq T$, inspired by the AdaBoost ensemble algorithm, the proposed SNHT algorithm assigns a weight α_t to each base model using the update rule $\alpha_t \leftarrow (1/2) * \log((1 - e_t)/e_t)$, where e_t is the error of the base model h_{t-1}. In this way, base models with higher accuracy are assigned larger weights, while weaker models obtain smaller weights. This strategy aims to enhance the diversity of base models, so that the final combined SVM classifier achieves stronger generalization performance. In contrast, the SVM methods presented in [13] and [1] only train a single model on the given dataset, and the hyperparameters of that model are determined via manual tuning.

3.2. Computational complexity

The SNHT algorithm consists of two stages: training an initial model in the first stage, and training multiple base models based on Markov samples in the second stage. Let T be the number of base models, n be the training set size, $N_t (0 \leq t \leq T)$ be the training subset size, and d be the feature dimension. The computational complexity of the first stage is $O(N_0 d)$, while that of the second stage is $O(\sum_{1\leq t\leq T} N_t d)$. Set $N_{max} = \max_{1\leq t\leq T} N_t$. Then, the overall complexity of the SNHT algorithm is $O((T + 1)N_{max}d)$, where the subset size $N_{max} = \max_t N_t \ll n$.

4. Estimating of generalization ability

In this section, we first estimate the generalization bound of $f_{\mathbf{z},\lambda} = \sum_{t=1}^{T} \alpha_t h_{\mathbf{z},\lambda_t}$ in Algorithm 1, and prove its consistency. Readers may refer to Reference [1, 31–33] for further theoretical details on the algorithm's generalization performance.

The property of classifier f is measured by the misclassification rate $L(f)$, which is defined by the probability of the event $\{f(X) \neq Y\}$, $L(f) = \mathrm{P}\{f(X) \neq Y\}$. The Bayes classifier is $f_c := \mathrm{sgn}(f_\rho)$ [34], which satisfies the optimal Bayes risk $L^* = L(f_c) = \inf_f L(f)$. Here the regression function of the distribution ρ is $f_\rho = \int y d\rho(y|x), x \in \mathcal{X}$. For the classifier $\mathrm{sgn}(f_{\mathbf{z},\lambda})$ obtained by Algorithm 1, we expect that $L(\mathrm{sgn}(f_{\mathbf{z},\lambda}))$ can become arbitrarily close to the optimal Bayes risk L^*, so the excess classification rate $L(\mathrm{sgn}(f_{\mathbf{z},\lambda})) - L^*$ is our estimation target. According to [9], we have

$$L(\mathrm{sgn}(f_{\mathbf{z},\lambda})) - L^* \leq \mathcal{R}(f_{\mathbf{z},\lambda}) - \mathcal{R}(f_\rho). \tag{4.1}$$

It follows from inequality (4.1) that we need to prove that the excess generalization error $\mathcal{R}(f_{\mathbf{z},\lambda}) - \mathcal{R}(f_\rho)$ converges to 0 in order to prove that the obtained classifier is consistent. Inspired by the idea from [32], we have the following error decomposition for $\mathcal{R}(f_{\mathbf{z},\lambda}) - \mathcal{R}(f_\rho)$.

Proposition 4.1. *Let* $h_{\lambda_t} = \arg\min_{h\in\mathcal{H}_K} \{\mathcal{R}(h) + \lambda_t \|h\|_K^2\}$, $\mathcal{R}(f_{\mathbf{z},\lambda}) - \mathcal{R}(f_\rho)$ *be decomposed as*

$$\mathcal{R}(f_{\mathbf{z},\lambda}) - \mathcal{R}(f_\rho) \leq \sum_{t=1}^{T} \alpha_t \{\mathcal{R}(f_{\mathbf{z},\lambda_t}) - \mathcal{R}(f_\rho)\} \leq \sum_{t=1}^{T} \alpha_t \{\Delta_t + \mathcal{D}(\lambda_t)\},$$

where $\Delta_t = \mathcal{R}(h_{\mathbf{z},\lambda_t}) - \mathcal{R}_{S_t}(h_{\mathbf{z},\lambda_t}) + \mathcal{R}_{S_t}(h_{\lambda_t}) - \mathcal{R}(h_{\lambda_t})$, $\mathcal{D}(\lambda_t) = \mathcal{R}(h_{\lambda_t}) - \mathcal{R}(f_\rho) + \lambda_t \|h_{\lambda_t}\|_K^2$.

The detail proofs of Proposition 4.1 are presented in Appendix A. In Proposition 4.1, Δ_t is the sample error, and $\mathcal{D}(\lambda_t)$ is the regularization error [11].

Definition 4.1. *[32] The function* f_ρ *is called to be approximated by* $\mathcal{H}_K$ *if for two constants* $0 < s \leq 1$ *and* $C_s > 0$, $\mathcal{D}(\lambda_t) \leq C_s \lambda_t^s$, $\forall \lambda_t > 0$ *for any* $1 \leq t \leq T$.

In this article, we assume that $|\ f_\rho(x)\ | \leq M$ for $x \in \mathcal{X}$ and a constant M [32]. In Proposition 4.1, the sample error Δ_t can be expressed as

$$\mathbb{E}(\xi_{t,1}) - \frac{1}{N_t}\sum_{i=1}^{N_t} \xi_{t,1}(z_i) + \frac{1}{N_t}\sum_{i=1}^{N_t} \xi_{t,2}(z_i) - \mathbb{E}(\xi_{t,2}), z_i \in S_t,$$

where $\xi_{t,1} = \ell(h_{\mathbf{z},\lambda_t}) - \ell(f_\rho), \xi_{t,2} = \ell(h_{\lambda_t}) - \ell(f_\rho)$.

The last term $\xi_{t,2}$ is a fixed random variable that can be estimated by probability inequalities. The first term $\xi_{t,1}$ is changing with the sample z runing over a set of functions, and should not be considered as a fixed function. To establish the generalization ability of the SNHT algorithm, we need the main tools as follows.

4.1. Main tools

Since the bound of the sample error is related to the capacity of the space $\mathcal{H}_K$, the concept of covering number is introduced below.

Definition 4.2. *[32] Let $\mathcal{G}$ be a subset of the metric space. For any $\eta > 0$, l is the number of balls with radius η covering $\mathcal{G}$, then the covering number $\mathcal{N}(\mathcal{G}, \eta)$ of $\mathcal{G}$ is the minimal integer $l \in \mathbb{N}$.*

Let $\mathcal{B}_R = \{h \in \mathcal{H}_K : \|h\|_K \leq R, R > 0\}$ be a ball. The covering number of $\mathcal{B}_1$ with the metric $\|\cdot\|_\infty$ is denoted by $\mathcal{N}(\mathcal{B}_1, \eta), \eta > 0$, then we have the following result.

Definition 4.3. *[32] There exists an exponent $r > 0$ if there is some $C_r > 0$ such that $\ln \mathcal{N}(\mathcal{B}_1, \eta) \leq C_r(1/\eta)^r, \ \forall\ \eta > 0$.*

Remark 4.1. *By [33], if K is $C^{2n/r}$ on a subset $\mathcal{X}$ of $\mathbb{R}^n$, Definition 4.3 is valid. In particular, for a C^∞ kernel (such as Gaussians), Definition 4.3 holds true for any $r > 0$.*

The following lemma, which is the Bernstein inequality with u.e.M.c. samples, plays an important role in our proof.

Lemma 4.1. *[27] Suppose that $\{z_i\}_{i=1}^m$ is u.e.M.c. sample and $\mathcal{G}$ is a countable class of bounded measurable functions. Let $0 \leq g(z) \leq C$ for any $g \in \mathcal{G}$ and any z. We have that for any $\varepsilon > 0$,*

$$P\Big\{ \mid \frac{1}{m}\sum_{i=1}^{m} g(z_i) - \mathbb{E}(g) \mid \geq \varepsilon \Big\} \leq 2\exp\{\frac{-m\varepsilon^2}{56C \parallel \Gamma' \parallel^2 \mathbb{E}(g)}\},$$

where $\parallel \Gamma' \parallel = \sqrt{2\gamma_0}/(1 - \sqrt{\varrho})$.

Lastly, before presenting the main conclusions, we also need the following Propositions 4.2 and 4.3, which will be proven in Appendix A.

Proposition 4.2. *Let $S_t = \{z_i\}_{i=1}^{N_t} (1 \leq t \leq T)$ be an u.e.M.c. sample. Then for any $\eta \in (0, 1)$, we have that with confidence $1 - \eta$,*

$$\frac{1}{N_t}\sum_{i=1}^{N_t} \xi_{t,2}(z_i) - \mathbb{E}(\xi_{t,2}) \leq \frac{1}{2}\mathcal{D}(\lambda_t) + \frac{56\ln(1/\eta)\|\Gamma'\|^2}{N_t}(\kappa\sqrt{\mathcal{D}(\lambda_t)/\lambda_t} + M).$$

Proposition 4.3. *Let $S_t = \{z_i\}_{i=1}^{N_t} (1 \leq t \leq T)$ be an u.e.M.c. sample and $R \geq 0$. Then, for any $\eta \in (0, 1)$, inequality*

$$\mathbb{E}(\xi_{t,1}) - \frac{1}{N_t}\sum_{i=1}^{N_t} \xi_{t,1}(z_i) \leq \frac{1}{2}[\mathcal{R}(h_{\mathbf{z},\lambda_t}) - \mathcal{R}(f_\rho)] + \frac{896(\kappa R + M)\|\Gamma'\|^2\{C_r(4RN_t^\tau)^r + \ln(1/\eta)\}}{N_t}$$

holds with confidence $1 - \eta$ provided that $N_t \geq (896(\kappa R + M)\|\Gamma'\|^2\{C_r(4R)^r + \ln(1/\eta)\})^{1/(1-\tau-\tau r)}$.

4.2. Main results

Our main results on the generalization bound of SNHT can be given as follows.

Theorem 4.1. *Let* $\mathbf{z} = \{z_i = (x_i, y_i)\}$ *be a training set,* $S_t = \{z_i\}_{i=1}^{N_t} (1 \le t \le T)$ *be an u.e.M.c. sample and* $R \le M/\kappa$. *Then, for any* $\eta \in (0, 1)$, *the inequality*

$$\mathcal{R}(f_{\mathbf{z},\lambda}) - \mathcal{R}(f_\rho) \le \sum_{t=1}^{T} \alpha_t \Big\{ \frac{3584 M^{r+1} C_r \|\Gamma'\|^2 4^r \kappa^{-r}}{N_t^{1-\tau r}} + 3\mathcal{D}(\lambda_t) + \frac{112 \ln(2T/\eta) \|\Gamma'\|^2 (\kappa \sqrt{\mathcal{D}(\lambda_t)/\lambda_t} + 33M)}{N_t} \Big\}$$

is valid with confidence $1 - \eta$ *provided that* $N_t \ge N^* := (1792M\|\Gamma'\|^2(C_r(4M/\kappa)^r + \ln(2T/\eta)))^{1/(1-\tau-\tau r)}$.

By Theorem 4.1 and Definition 4.1, we get the convergence rate of the SNHT algorithm with u.e.M.c. samples as follows.

Theorem 4.2. *By Theorem 4.1, take* $\lambda_t = (N_t t)^{-\beta}$ *for* $1 < \beta < \frac{1}{1-s}$. *Then, for any* $\eta \in (0, 1)$, *the inequality*

$$L(\mathrm{sgn}(f_{\mathbf{z},\lambda})) - L^* \le C\Big(\frac{1}{N}\Big)^{1-(1-s)\beta}$$

is valid with confidence $1 - \eta$ *provided that* $N = \min_t N_t \ge \max\{N^*, T\}$. *Here,* $N^* = (1792M\|\Gamma'\|^2(C_r(4M/\kappa)^r + \ln(2T/\eta)))^{1/(1-\tau-\tau r)}$, $C = 3696c'M + 3C_s + 3584M^{r+1}C_r\|\Gamma'\|^2 4^r\kappa^{-r} + 112c'\kappa C_s^{1/2}$ *is a constant.*

We will prove Theorems 4.1 and 4.2 in Appendix B.

Remark 4.2. *(1) Since* $N \to +\infty$ *as* $n \to +\infty$. *Thus, by Theorem 4.2, we can conclude that* $L(\mathrm{sgn}(f_{\mathbf{z},\lambda})) - L^* \to 0$ *as* $n \to +\infty$, *which implies that the proposed SNHT algorithm is consistent. As* $s \to 1$, *we have* $1 - (1 - s)\beta$ *is arbitrarily close to* 1. *Thus, by Theorem 4.2, we have that the convergence rate of* $L(\mathrm{sgn}(f_{\mathbf{z},\lambda})) - L^*$ *can be close to* $O(1/N)$. *Compared with the results of cross-validation tuning parameters in [13] for u.e.M.c. sequences and [35, 36] for i.i.d. samples, we can find that this article achieves the same optimal learning rate as that in [13, 35, 36].*

(2) Compared to the results established in [13, 35, 36] with Theorem 4.2, we can find that although [13, 35, 36] and this article focuses on the generalization bounds of SVM, and many steps in our proof technique are similar to those of [13, 35, 36], the difference between [13, 35, 36] and Theorem 4.2 is obvious: the results established in [13, 35, 36] are based on a fixed regularization hyperparameter in order to estimate the optimal rate. Meanwhile Theorem 4.2 is not based on a fixed regularization parameter, since in this paper we do not choose the regularization hyperparameter.

5. Numerical experiments

In this section, we compare the SNHT with 6 algorithms: the classical SVM [1] (SCV), the SVM based on Markov sampling introduced in [13] (SCVM), Light Gradient Boosting Machine (LGBM) [37], Simple Multilayer Perceptron (MLP) [38], Random Forest (RF) [39].

The performances of the above 6 algorithms are measured by accuracy, precision, recall and F_1 score, which are defined as follows:

$$\text{accuracy} = \frac{n_{tp} + n_{tn}}{n_{tp} + n_{fp} + n_{tn} + n_{fn}}, \text{recall} = \frac{n_{tp}}{n_{tp} + n_{fn}},$$
$$\text{precision} = \frac{n_{tp}}{n_{tp} + n_{fp}}, \qquad F_1 = \frac{2\,\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}},$$

where n_{tn} and n_{tp} are the sizes of true negative and true positive; n_{fn} and n_{fp} are the sizes of false negative and false positive, respectively.

5.1. Datasets and experimental setup

The experiments adopt 9 real-world binary classification benchmark datasets: Skin, Seismic, W7a, Human Activities and Postural Transitions (HAPT), and TV-News are from the University of California, Irvine (UCI)*, and German, Acoustic, Covtype, and Aloi are from Libsvm†. We divide each dataset into a train set S_{train} and a test set S_{test}. These datasets are described in Table 1.

Table 1. 9 real-world datasets.

Dataset	$\#S_{train}$	$\#S_{test}$	# Input dimension
Skin	163371	81686	3
German	70000	30000	20
Seismic	73896	24632	50
Acoustic	73896	24632	50
Covtype	435759	145253	54
Aloi	72000	36000	128
W7a	33166	16583	300
HAPT	7767	2589	561
TV-news	86457	43228	4125

In this article, the hyperparameters of SCV and SCVM are determined by grid search with the 5-fold cross-validation method. Due to the time consumption and high memory requirements of cross-validation, we randomly select 30000 samples from the given training set for cross-validation if its size exceeds 30000, otherwise we use all the training set. For the proposed SNHT algorithm, we take $T = 7$ and assign different values of hyperparameter λ for the base linear SVM by $\lambda_1 = 0.0005, \lambda_2 = 0.005, \cdots, \lambda_T = 500$. In addition, all data is normalized in the preprocessing step to avoid the influence of numerical range on characteristic attributes and facilitate numerical calculations. The LGBM, MLP, and RF algorithms are implemented with their default parameter settings.

Comparative experiments of SNHT with SCV and SCVM algorithms are implemented on a PC with 2.20 GHz CPU and 32GB RAM using Matlab R2020a. Experiments comparing SNHT with LGBM, MLP, and RF are implemented on a PC with 3.00 GHz CPU and 32GB RAM using Python.

*https://archive.ics.uci.edu/ml/index.php
†https://www.csie.ntu.edu.tw/ ~cjlin/libsvmtools/datasets/

5.2. Experimental results

Now we state simply our experiments as follows:

(1) Train these 6 algorithms on the training set S_{train}, then test them on the test set S_{test} and compute the corresponding accuracy, precision, recall, and F_1 score.

(2) Combine S_{train} and S_{test}, and randomly redivide them into new S'_{train} and S'_{test} (with sizes matching the original sets).

(3) Repeat steps (1)–(2) for k times and calculate the average values of accuracy, precision, recall and F_1 score and the average time (sampling and training) for the 6 algorithms.

5.2.1. Compare with SCV and SCVM

We first present the comparative experimental results of the three algorithms: SNHT, SCV, and SCVM. In Tables 2 and 3, we set $T = 7$, $n = 8000$, where n denotes the training set size (except $n = 7767$ for HAPT dataset since the size of HAPT dataset is less than 8000). All these results are based on $k = 50$.

Tables 2 and 3 show that for $n = 8000$, $T = 7$, the means of accuracy, time, and F_1 score of the SNHT algorithm are better than that of SCV, SCVM. However, the mean precision of SNHT is significantly lower than that of SCV/SCVM on the W7a dataset. There are two main reasons for this. First, precision and recall are a pair of mutually constrained metrics: improving precision usually reduces recall, and vice versa. Consequently, when SCVM achieves extremely high precision, it leads to an extremely low recall (e.g., only 5.14% on the W7a dataset). This also explains why SNHT achieves slightly higher precision on some datasets (e.g., Seismic) but has slightly lower mean recall on those same datasets. Second, the class ratio of positive to negative samples in the W7a dataset is severely imbalanced (1:32). This leads to SCV/SCVM achieving much higher precision on the positive class than SNHT. In contrast, SNHT maintains a more balanced trade-off between precision and recall, thus achieving the highest F_1 score across all datasets. The F_1 score is more indicative of an algorithm's overall classification performance in real-world scenarios.

Table 2. Experimental results of SCV, SCVM, and SNHT algorithms for $n = 8000$, $T = 7$.

Dataset	Accuracy (%)			Train time(s)		
	SCV	SCVM	SNHT	SCV	SCVM	SNHT
Skin	93.14±0.25	94.01±0.29	94.40±0.09	12138	12148	42
German	77.09±0.41	77.16±0.47	78.52±0.44	50591	50642	169
Seismic	83.03±0.35	83.24±0.38	84.55±0.39	176019	176070	789
Acoustic	73.30±0.38	74.01±0.47	75.76±0.34	233617	233660	942
Covtype	71.73±0.28	72.04±0.39	72.84±0.26	75646	75769	465
Aloi	63.03±0.37	62.73±0.42	64.05±0.30	117101	117215	6564
W7a	97.59±0.14	97.18±0.12	97.91±0.14	21737	21756	86
HAPT	99.67±0.11	99.52±0.16	99.76±0.10	70	106	36
TV-news	88.47±0.14	88.17±0.15	88.87±0.15	45572	45774	568

Table 3. Experimental results of SCV, SCVM, and SNHT algorithms for $n = 8000$, $T = 7$.

Dataset	Precision (%)			Recall (%)			F_1 score (%)		
	SCV	SCVM	SNHT	SCV	SCVM	SNHT	SCV	SCVM	SNHT
Skin	77.30	77.78	78.75	94.77	99.61	99.98	85.15	87.35	88.10
German	66.22	66.58	66.52	48.19	47.91	57.21	55.77	55.70	61.47
Seismic	76.51	77.18	79.85	95.35	94.41	92.50	84.90	84.92	85.68
Acoustic	68.73	69.72	72.15	85.56	84.96	83.96	76.22	76.58	77.60
Covtype	75.11	75.77	77.61	75.90	75.43	74.03	75.50	75.58	75.77
Aloi	63.88	63.42	64.82	60.27	60.61	61.57	61.96	61.86	63.12
W7a	95.44	99.89	75.76	19.89	05.14	43.87	32.72	7.99	55.33
HAPT	99.27	99.10	99.42	98.66	97.87	99.06	98.96	98.48	99.24
TV-news	89.95	89.72	90.79	92.11	91.87	91.77	91.01	90.77	91.27

To analyze whether there exist statistically significant differences among SNHT, SCV, and SCVM, Tables 4 and 5 present the Wilcoxon signed-rank test [40] based on the experimental results given in Tables 2 and 3.

Table 4. Wilcoxon tests of SNHT and SCV.

Comparison	R_+	R_-	Hypothesis($\alpha = 0.05$)	Selected
Accuracy	45	0	Rejected	SNHT
Precision	36	9	Not Rejected	SNHT
Recall	29	16	Not Rejected	SNHT
F_1 score	45	0	Rejected	SNHT

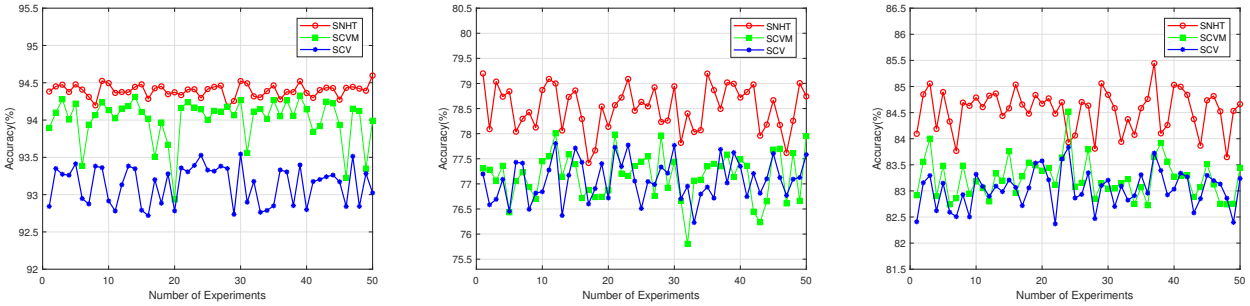
Table 5. Wilcoxon tests of SNHT and SCVM.

Comparison	R_+	R_-	Hypothesis($\alpha = 0.05$)	Selected
Accuracy	45	0	Rejected	SNHT
Precision	35	10	Not Rejected	SNHT
Recall	27	18	Not Rejected	SNHT
F_1 score	45	0	Rejected	SNHT

Tables 4 and 5 show that there is no significant difference between SNHT and SCV/SCVM in terms of precision and recall, but there exists a significant difference for accuracy and F_1 score.

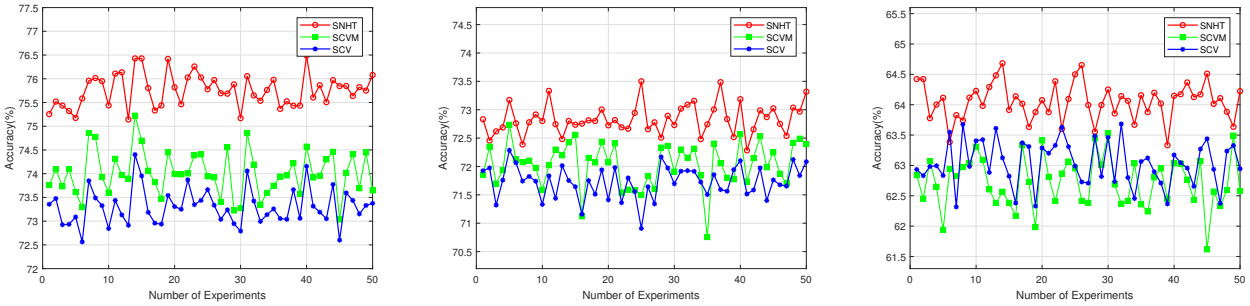
To better demonstrate the learning performance of the proposed SNHT algorithm, Figures 1–3 present the accuracy of SCV, SCVM, and SNHT over 50 repeated experiments. The horizontal axis represents the number of repeated experiments,and the vertical axis represents accuracy.

Figures 1–3 show that almost all of the 50 experimental accuracies of SNHT are better than those of SCV and SCVM, except for at most 3 times on the HAPT dataset.



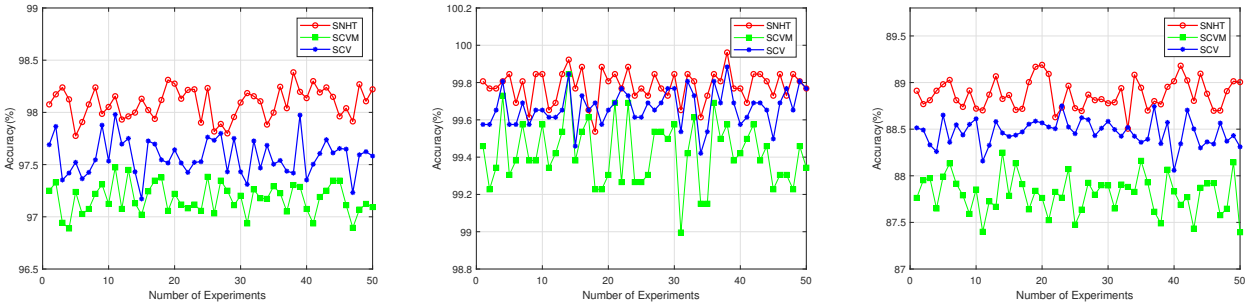
(a) (b) (c)

Figure 1. 50-times repeated experimental accuracies (%) of SCV, SCVM, and SNHT for $n = 8000$, $T = 7$. (a) Skin; (b) German; (c) Seismic.



(a) (b) (c)

Figure 2. 50-times repeated experimental accuracies (%) of SCV, SCVM, and SNHT for $n = 8000$, $T = 7$. (a) Acoustic; (b) Covtype; (c) Aloi.



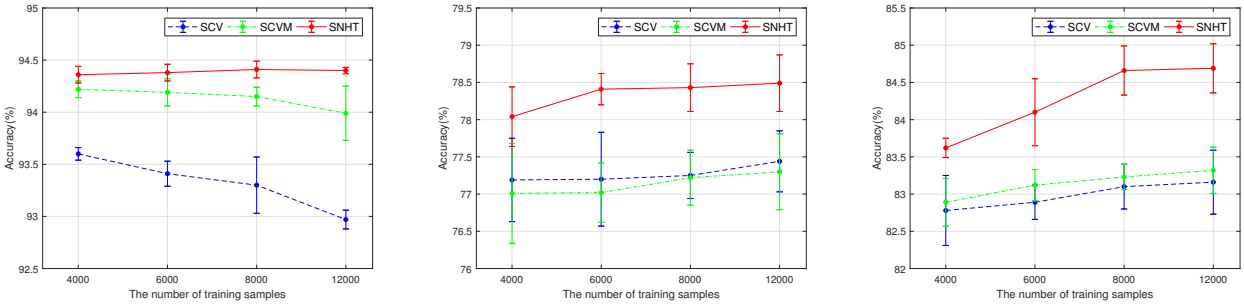
(a) (b) (c)

Figure 3. 50-times repeated experimental accuracies (%) of SCV, SCVM, and SNHT for $n = 8000$, $T = 7$. (a) W7a; (b) HAPT; (c) TV-news.

In Figures 4–9, we compare the average accuracy and the average training time of SNHT with the other two algorithms for different n with $T = 7$. The horizontal axis of Figures 4–6 is the value of n

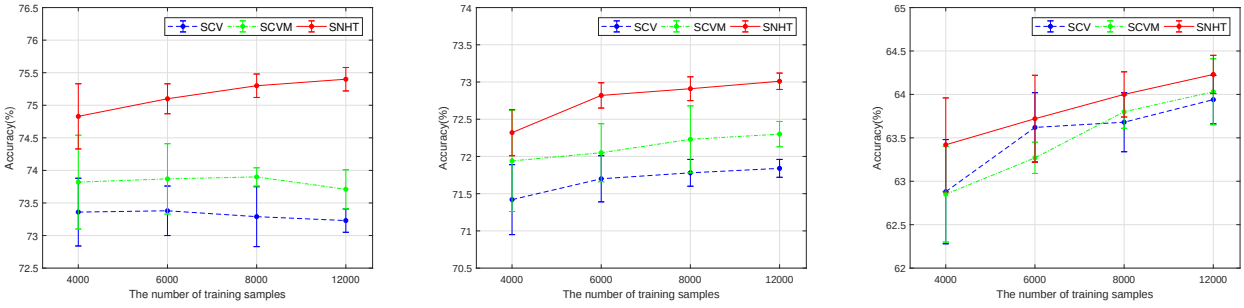
while the vertical axis is accuracy. In Figures 7–9, "SCV/10", "SCVM/10" represent one-tenth of the average training time for the SCV and SCVM algorithms, respectively, and "SCV", "SCVM", "SNHT" represent the average training time for SCV, SCVM, and SNHT, respectively. All results are based on $k = 50$.

From Figures 4–9, we can see that for different n with $T = 7$, all the accuracies of SNHT are better than those of the SCV and SCVM algorithms, and the training time of the SNHT algorithm is much less than that of SCV and SCVM algorithms.



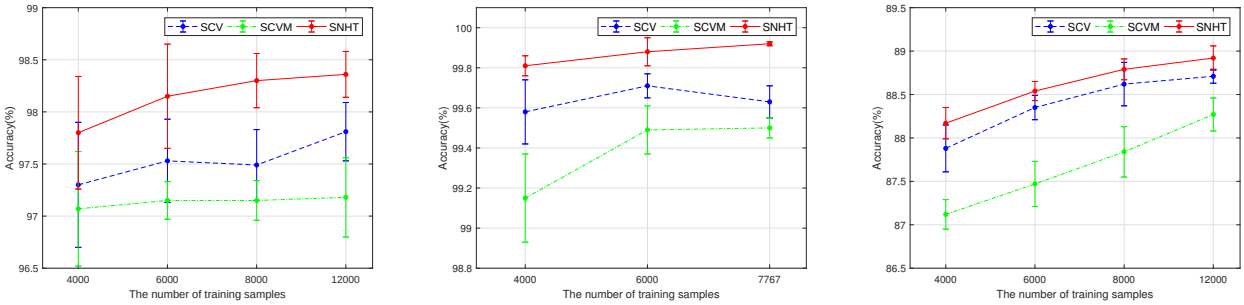
(a) (b) (c)

Figure 4. Accuracies (%) of SCV, SCVM, and SNHT for different n with $T = 7$. (a) Skin; (b) German; (c) Seismic.



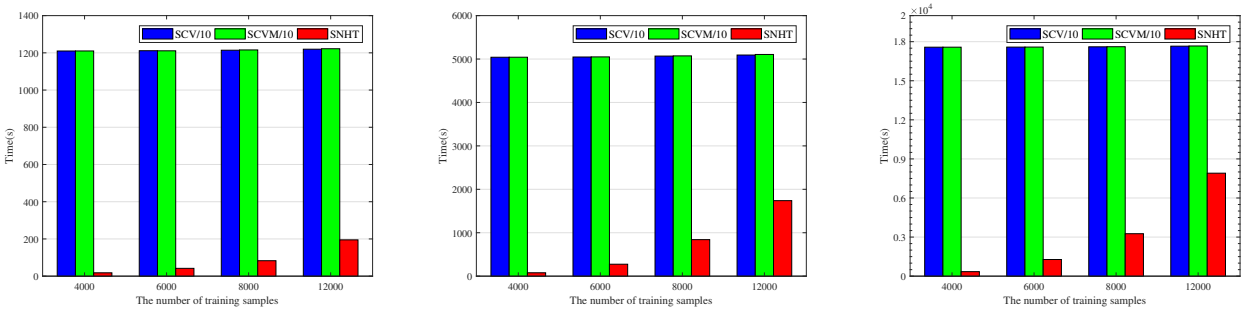
(a) (b) (c)

Figure 5. Accuracies (%) of SCV, SCVM, and SNHT for different n with $T = 7$. (a) Acoustic; (b) Covtype; (c) Aloi.



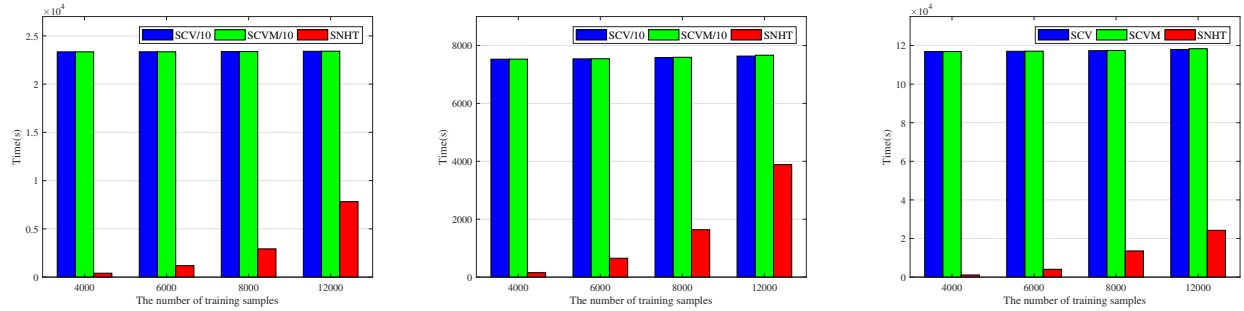
(a) (b) (c)

Figure 6. Accuracies (%) of SCV, SCVM, and SNHT for different n with $T = 7$. (a) W7a; (b) HAPT; (c) TV-news.



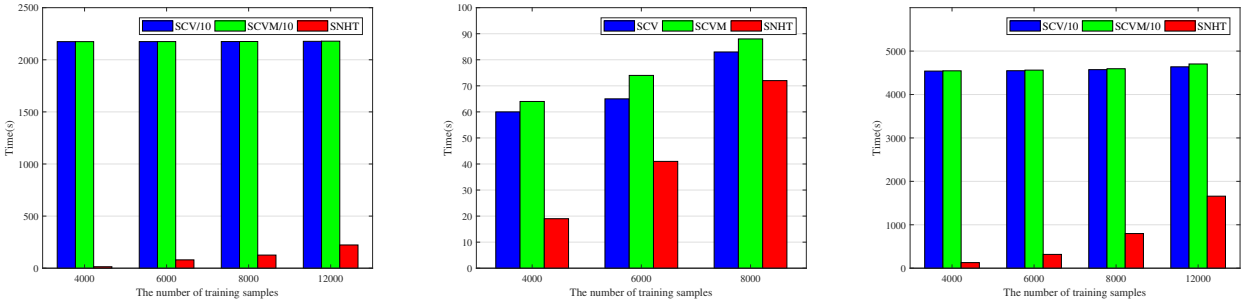
(a) (b) (c)

Figure 7. Train time (s) of SCV, SCVM, and SNHT for different n with $T = 7$. (a) Skin; (b) German; (c) Seismic.



(a) (b) (c)

Figure 8. Train time (s) of SCV, SCVM, and SNHT for different n with $T = 7$. (a) Acoustic; (b) Covtype; (c) Aloi.



(a) (b) (c)

Figure 9. Train time (s) of SCV, SCVM, and SNHT for different n with $T = 7$. (a) W7a; (b) HAPT; (c) TV-news.

5.2.2. Compare with LGBM, MLP, and RF

In this section, we present the comparative experimental results of the four algorithms: SNHT, LGBM, MLP, and RF. In Tables 6 and 7, we set the same parameters $T = 7$, $n = 8000$ (except $n = 7767$ for HAPT dataset since the size of HAPT dataset is less than 8000). All these results are based on $k = 50$.

Table 6. Experimental results of SNHT, LGBM, MLP, and RF algorithms for $n = 8000$, $T = 7$.

Dataset	Accuracy (%)				Train time(s)			
	LGBM	MLP	RF	SNHT	LGBM	MLP	RF	SNHT
Skin	99.79±0.05	99.80±0.01	99.81±0.01	94.38±0.01	5	135	26	203
German	99.79±0.01	99.62±0.05	100±0	78.61±0.39	4	128	16	604
Seismic	84.91±0.03	84.90±0.08	84.97±0.07	84.47±0.22	8	160	293	643
Acoustic	78.74±0.06	76.55±0.07	78.90±0.03	75.71±0.43	8	170	236	991
Covtype	80.10±0.08	77.18±0.28	81.83±0	72.86±0.27	5	184	44	244
Aloi	80.95±0.69	76.91±0.55	87.88±0.2	64.06±0.21	12	886	63	1410
W7a	98.10±0.01	98.34±0.14	98.57±0.06	97.95±0.04	20	859	168	560
HAPT	99.85±0	99.98±0.02	99.65±0	99.92±0.04	66	158	393	116
TV-news	91.30±0.06	85.42±1.02	89.21±0.04	88.85±0.14	62	3057	209	3425

To clearly compare the performance of the SNHT algorithm, the optimal results among all methods are marked in bold and the suboptimal results achieved by the SNHT algorithm are underlined in Tables 6 and 7. Tables 6 and 7 show that SNHT only achieves the highest recall on the Seismic, Acoustic, and HAPT datasets, and ranks first in F_1 score on Seismic. Additionally, its training time on HAPT is better than those of RF and MLP, highlighting its unique advantage in recall-sensitive tasks. LGBM has a significant advantage in training efficiency across all datasets, being much faster than other algorithms. RF achieves the highest accuracy and F_1 score on six datasets (Skin, German, Acoustic, Covtype, Aloi, W7a), and demonstrating strong generalization ability and stability.

MLP achieves the highest accuracy and F_1 score on the HAPT dataset, showing strong potential for specific tasks.

Table 7. Experimental results of SNHT, LGBM, MLP, and RF algorithms for $n = 8000$, $T = 7$.

Dataset	Precision (%)				Recall (%)				F_1 score (%)			
	LGBM	MLP	RF	SNHT	LGBM	MLP	RF	SNHT	LGBM	MLP	RF	SNHT
Skin	99.32	99.10	99.31	78.94	99.69	99.95	99.76	99.96	99.51	99.52	99.54	88.22
German	99.82	99.82	100	66.31	99.48	98.89	100	43.38	99.65	99.35	100	52.45
Seismic	81.96	81.27	83.14	79.93	89.50	90.68	87.72	92.58	85.57	85.72	85.37	85.79
Acoustic	76.88	74.40	76.41	72.33	82.12	80.94	83.54	83.66	79.42	77.53	79.82	77.58
Covtype	81.70	77.98	83.06	71.24	84.08	83.82	85.78	80.61	82.87	80.80	84.40	75.64
Aloi	80.57	74.63	87.52	63.08	81.55	81.57	88.33	63.28	81.06	77.95	87.93	63.18
W7a	85.99	83.73	90.19	76.88	46.36	58.32	60.56	48.19	60.24	68.75	72.46	59.24
HAPT	99.75	99.88	99.75	100	99.26	100	98.04	99.51	99.51	99.94	98.89	99.75
TV-news	92.30	91.54	88.96	88.69	94.14	84.85	94.74	93.00	93.21	88.07	91.76	90.80

5.3. Robustness comparison

To evaluate the robustness of the proposed SNHT algorithm against SCV, SCVM, LGBM, MLP, and RF on both clean and Gaussian noise data, we define the relative percentage decrease as follows:

$RD(\%) = \frac{prediction_{Clean} - prediction_{Noisy}}{prediction_{Clean}} \times 100.$

Let $r_0 \in [0, 1]$ denote the noise ratio for the dataset, and $q \in [0, 1]$ denote the noise intensity coefficient. For each feature vector $\mathbf{x}$, we add Gaussian noise $\epsilon \sim N(0, \sigma^2)$, where $\sigma = q \times \text{std}(\mathbf{x})$ and std($\mathbf{x}$) stands for the standard deviation of $\mathbf{x}$. Here, "RD-Acc" and "RD-F_1" denote the relative percentage decreases in accuracy and F_1 score, respectively.

Table 8 presents the experimental results on the TV-news dataset under different noise ratios r_0 (15%, 20%, 25%) with $q = 0.5$. Table 9 presents the experimental results on the Acoustic dataset under different q (0.1, 0.4, 0.8) with $r_0 = 20\%$.

Tables 8 and 9 show that the addition of noise results in a positive relative percentage decrease for all algorithms, meaning that both accuracy and F_1 score have declined. Furthermore, across almost noise levels, the rate of performance decline for SNHT remains lower than that of other algorithms, except at a noise ratio of 15% on the TV-news dataset. This fully demonstrates that SNHT possesses superior robustness.

Table 8. Robustness Comparison for TV-news with different r_0.

Algorithm	15%		20%		25%	
	RD-Acc(%)	RD-F_1(%)	RD-Acc(%)	RD-F_1(%)	RD-Acc(%)	RD-F_1(%)
SCV	0.46	0.36	0.69	0.63	0.68	0.65
SCVM	25.80	12.05	26.37	12.32	26.49	12.41
LGBM	0.72	0.54	0.75	0.56	0.68	0.51
MLP	6.46	5.06	6.25	4.55	6.59	4.81
RF	1.92	1.42	2.50	1.84	3.03	2.23
SNHT	0.74	0.54	0.67	0.45	0.64	0.45

Table 9. Robustness Comparison for Acoustic with different q.

Algorithm	0.1		0.4		0.8	
	RD-Acc(%)	RD-F_1(%)	RD-Acc(%)	RD-F_1(%)	RD-Acc(%)	RD-F_1(%)
SCV	0.34	0.83	0.70	1.30	1.25	1.73
SCVM	6.56	39.70	20.64	96.91	20.66	96.65
LGBM	1.36	1.41	1.94	2.19	2.36	2.38
MLP	0.87	1.43	2.53	2.95	3.70	4.21
RF	1.46	1.50	2.14	2.44	2.19	2.43
SNHT	0.14	0.17	0.24	0.47	0.61	0.72

6. Discussions and explanations

In this section, we discuss the choice of T, α_t, and apply SNHT to distributed classification learning.

6.1. Choices of T

Table 10 shows different values of T and their corresponding ranges of the regularization hyperparameter λ.

Table 10. The range of λ for different T.

T	The range Λ of λ
3	$\{0.05, 0.5, 5\}$
5	$\{0.005, 0.05, 0.5, 5, 50\}$
7	$\{0.0005, 0.005, 0.05, 0.5, 5, 50, 500\}$
9	$\{0.00005, 0.0005, 0.005, 0.05, 0.5, 5, 50, 500, 5000\}$

Table 11 presents the experimental results on accuracy (Acc) and training time (Time) of the SNHT algorithm based on German, Covtype, and Aloi datasets for different T with $n = 8000$.

Table 11. Experimental results of SNHT for different T with $n = 8000$.

T	German		Covtype		Aloi	
	Acc (%)	Time (s)	Acc (%)	Time (s)	Acc (%)	Time (s)
3	77.62	161	72.16	317	63.74	4879
5	77.98	349	72.64	727	63.92	9605
7	78.18	585	72.89	1291	64.00	13447
9	78.33	831	73.16	1950	64.12	16439

From Table 11, we can find that the accuracy and the training time of SNHT both have a tendency to increase as T increases. Therefore, to balance accuracy and training time, we choose $T = 7$ and the corresponding regularized hyperparameter for the results presented in the last section.

6.2. Choices of weight α_t

In SNHT, the weight of each base model follows the AdaBoost updating rule: $\alpha_t = \frac{1}{2}\log((1-e_t)/e_t)$ for $t = 1, \ldots, T$, where e_t denotes the error of the base model h_t. To demonstrate the advantage of this weighting scheme, we compare it against two alternatives: equal weight ($\alpha_t = 1/T$) and reciprocal error weight ($\alpha_t = 1/e_t$). For brevity, the corresponding accuracies are denoted as Acc-E (equal weight), Acc-R (reciprocal error weight), and Acc-S (SNHT weight), respectively.

Table 12 presents the average accuracy of the SNHT algorithm on the Acoustic and Covtype datasets under different weight settings for different values of T, with $n = 2000$ and $k = 5$.

Table 12. Acc (%) of SNHT for different α_t with $n = 2000$.

T	Acoustic			Covtype		
	Acc-E	Acc-R	Acc-S	Acc-E	Acc-R	Acc-S
3	73.24	73.24	73.24	67.11	67.11	67.11
5	73.47	73.47	73.47	67.99	67.99	70.35
7	73.64	73.64	73.76	68.02	68.02	70.64
9	73.74	73.75	73.80	68.12	70.64	70.82

From Table 12, we can find that the accuracy of SNHT weight is higher than that of the equal weight and reciprocal error strategies for different values of T.

Table 13 presents the average accuracy of the SNHT algorithm on the Covtype and Aloi datasets under different weight settings for different values of n, with $T = 7$ and $k = 5$.

Table 13. Acc (%) of SNHT for different α_t with $T = 7$.

n	Covtype			Aloi		
	Acc-E	Acc-R	Acc-S	Acc-E	Acc-R	Acc-S
2000	68.02	68.02	70.64	62.82	62.82	63.51
4000	69.64	69.64	71.53	63.38	63.38	63.89
8000	71.15	71.15	71.72	63.75	63.75	64.00
10000	71.30	71.30	71.78	63.80	63.80	64.10

6.3. Applications to distributed learning

SNHT is applied to the distributed learning for classification. Let $\widehat{T}$ be the number of training dataset blocks in the distributed learning framework, that is, $S_{train} = \bigcup_{i=1}^{\widehat{T}} S^i_{train}$. The proposed SNHT and SCVM algorithms are applied to each dataset S^i_{train} ($1 \le i \le \widehat{T}$). In Table 14, "D-SNHT" and "D-SCVM" denote the results of SNHT and SCVM under the distributed learning framework, respectively. All experimental results are based on $T = 7$, $n = |S^i_{train}|$ (the size of each block), and 50 repeated experiments. The training times presented in Table 14 are averages.

Table 14. The experimental results of D-SNHT and D-SCVM for $T = 7$.

Dataset	$\widehat{T}$	Accuracy (%)		Precision (%)		Recall (%)		F_1 score (%)		Time (s)	
		D-SCVM	D-SNHT	D-SCVM	D-SNHT	D-SCVM	D-SNHT	D-SCVM	D-SNHT	D-SCVM	D-SNHT
Skin	20	94.17±0.06	94.38±0.06	78.07	78.69	99.97	100	87.67	88.07	13193	1673
German	8	77.13±0.32	78.45±0.39	66.39	65.67	48.01	58.88	55.71	62.08	52441	6104
Seismic	9	83.71±0.22	84.68±0.29	77.84	79.25	94.22	93.97	85.25	85.98	178782	24168
Acoustic	9	74.08±0.24	75.24±0.22	69.58	71.12	84.91	85.49	76.71	77.40	235701	24561
Covtype	50	72.53±0.10	72.86±0.09	76.64	78.25	74.93	72.95	75.78	75.50	102425	80799
Aloi	8	64.63±0.25	64.85±0.27	65.42	65.39	62.10	63.15	63.71	64.24	120141	f65606
W7a	4	97.30±0.12	98.27±0.10	97.25	85.71	08.75	49.42	16.04	62.62	21835	690
HAPT	1	99.44±0.15	99.77±0.08	98.99	99.28	97.45	99.26	98.21	99.27	101	40
TV-news	10	88.95±0.13	89.17±0.13	90.56	91.19	92.18	91.79	91.36	91.49	49952	13791

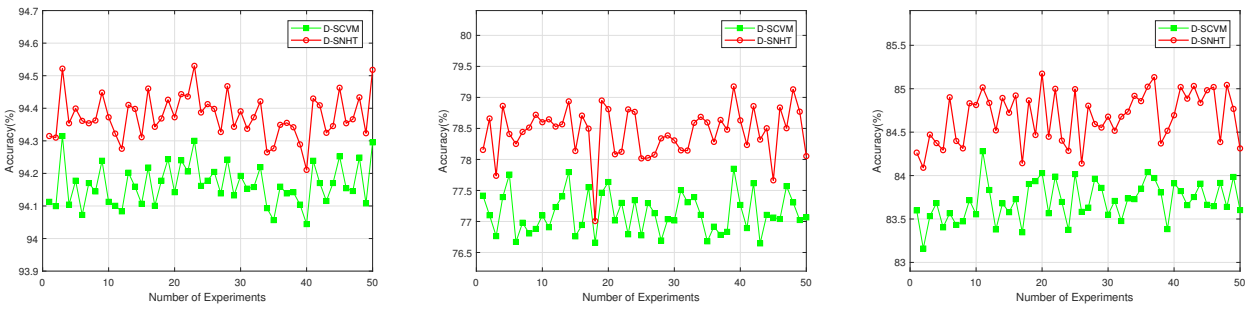
Table 14 shows that accuracy and F_1 of D-SNHT are better than those of D-SCVM, except for the F_1 score on the Covtype dataset. In addition, the training time of D-SNHT is much less than that of D-SCVM.

To analyze whether there exist statistically significant differences between the D-SNHT and D-SCVM algorithms, we apply the Wilcoxon signed-rank test [40] to the average accuracy, precision, recall and F_1 score shown in Table 14, with the corresponding results presented in Table 15.

Table 15. Wilcoxon tests of D-SNHT and D-SCVM.

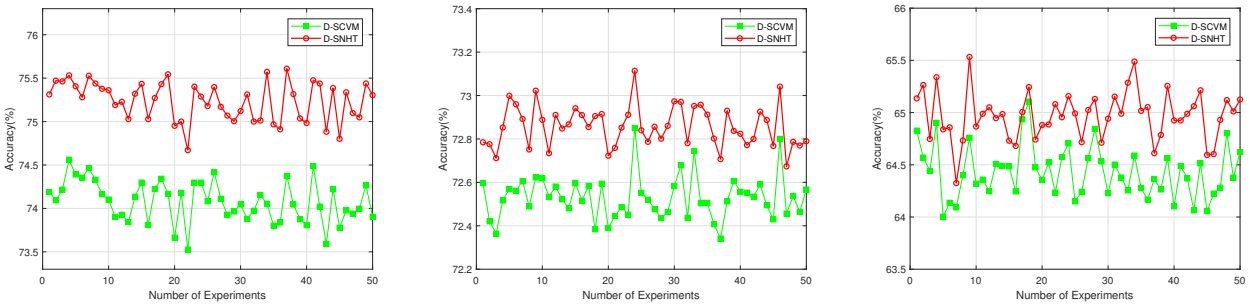
Comparison	R_+	R_-	Hypothesis($\alpha = 0.05$)	Selected
Accuracy	45	0	Rejected	D-SNHT
Precision	30	15	Not Rejected	D-SNHT
Recall	33	12	Not Rejected	D-SNHT
F_1 score	43	2	Rejected	D-SNHT

Table 15 indicates that there is no significant difference between D-SNHT and D-SCVM for precision and recall, but there is a significant difference for accuracy and F_1 score. Figures 10–12 compare the accuracy of D-SNHT and D-SCVM over 50 repeated experiments.



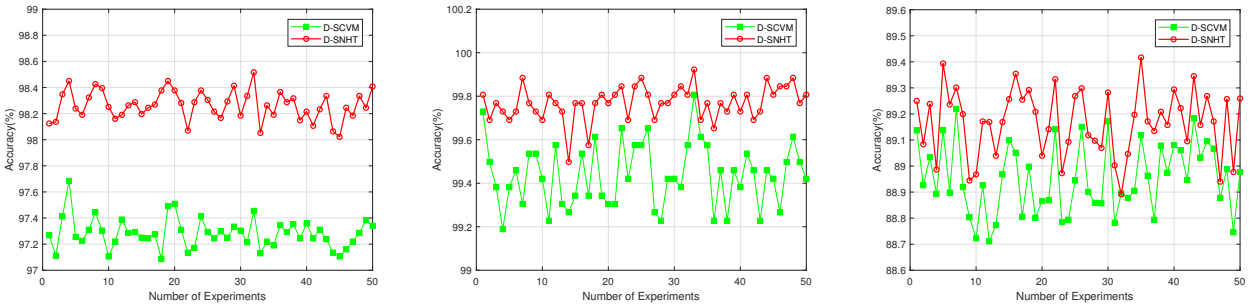
(a) (b) (c)

Figure 10. 50-times experimental Accuracies of D-SCVM and D-SNHT for $T = 7$. (a) Skin, $\widehat{T} = 20$; (b) German, $\widehat{T} = 8$; (c) Seismic, $\widehat{T} = 9$.



(a) (b) (c)

Figure 11. 50-times experimental Accuracies of D-SCVM and D-SNHT for $T = 7$. (a) Acoustic, $\widehat{T} = 9$; (b) Covtype, $\widehat{T} = 40$; (c) Aloi, $\widehat{T} = 8$.



(a) (b) (c)

Figure 12. 50-times experimental Accuracies of D-SCVM and D-SNHT for $T = 7$. (a) W7a, $\widehat{T} = 4$; (b) HAPT, $\widehat{T} = 1$; (c) TV-news, $\widehat{T} = 10$.

As shown in Figures 10–12, almost all of the 50 experimental accuracies of the D-SNHT algorithm are higher than those of the D-SCVM algorithm.

6.4. Explanations of learning performance

In this section, we give some explanations on the learning performance of the proposed algorithm. Hyperparameter selection requires balancing data characteristics and resource constraints. The proposed method integrates multiple independently trained hyperparameter models (each with different hyperparameters) through weighted ensemble methods, reducing reliance on a single parameter. By leveraging model diversity, it lowers variance and bias, balances overfitting and underfitting, and enhances the model's stability against noisy data and outliers. However, this approach risks compromising the generalization performance of the algorithm. To address this, we introduce Markov sampling for each independent SVM model. Since the examples that are close to the interface of two classes of data are the most "important" examples for classification problems and the number of such examples is smaller compared to the size of the total training set, in Algorithm 1 we define the acceptance probability p_t^i based on the last classification function h_t, and then we use these probabilities

to accept the training examples in the Markov resampling process, so that the "important" examples can be accepted with high probabilities. In other words, in Algorithm 1, these "important" examples for classification problems are drawn, which is the reason that the misclassification rates of the proposed SNHT algorithm are smaller than those of SVM algorithms based on regularized hyperparameter tuning. It is also the reason that we introduce the idea of Markov sampling. In addition, since the number of examples used to obtain the base classification functions is significantly smaller compared to the size of the given training set, the total sampling and training time of the proposed SNHT algorithm is shorter than that of SVM algorithms based on regularized hyperparameter tuning.

7. Conclusions

To address the challenges brought by hyperparameter learning in SVM models, this paper proposes a novel linear kernel SVM based on non-regularized hyperparameter tuning (SNHT). By training multiple SVM models with different regularization parameters and ensembling them to obtain the final model, the proposed method avoids overfitting or underfitting caused by fixed regularization parameters. Markov sampling is adopted to generate diverse training subsets to improve model diversity. The consistency of the algorithm is theoretically proved. Extensive numerical experiments demonstrate that, compared with SVM variant algorithms (SCV/SCVM), the proposed SNHT algorithm not only achieves higher classification accuracy but also requires less training time. Compared with existing parameter-free or low-parameter algorithms (such as LGBM, MLP, and RF), the proposed SNHT algorithm exhibits excellent recall characteristics only on low-dimensional datasets. For recall-prioritized tasks, SNHT can still serve as an effective alternative. Under almost all noise levels, the rate of performance degradation of SNHT is lower than that of other algorithms. This result fully demonstrates that SNHT possesses superior robustness.

Based on the current research, it should be noted that the SNHT algorithm still has limitations and shortcomings. For example, the SNHT algorithm currently only supports the linear kernel, and its generalization performance under nonlinear kernels has not been sufficiently studied. Moreover, the algorithm is only designed for binary classification tasks, and its adaptability to multi-class scenarios needs to be further verified. Future research will focus on solving the above issues to further improve the generalization performance of the algorithm and its adaptability to real-world scenarios. First, regarding kernel optimization, we will explore the performance of SNHT under nonlinear kernels, and theoretically analyze the influence of different kernel functions (such as Radial Basis Function (RBF) kernel and polynomial kernel) on the generalization ability of the algorithm, so as to overcome the limitation that the linear kernel is difficult to fit nonlinear data distributions and broaden the application scope of the algorithm. Second, in terms of multi-classification applications, we will extend the algorithm to multi-class scenarios in follow-up work. Specifically, the one-vs-one (OvO) or one-vs-all (OvA) strategy will be adopted, and the performance of SNHT will be verified on tasks such as multi-class text classification and multi-label image recognition, aiming to alleviate class imbalance and ambiguous decision boundaries in multi-class settings.

Author contributions

Hongwei Jiang: Methodology, software, investigation, writing-Original draft. Yujing Yang: Data curation, formal analysis, visualization. Bin Zou: Conceptualization, writing-Reviewing and editing, supervision. Jie Xu: Software, validation, supervision. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors gratefully acknowledge the financial support from the National Key Research and Development Program of China (No. 2020YFA0714200).

Conflict of interest

The authors declare no conflicts of interest or financial relationships relevant to this work.

References

1. V. N. Vapnik, *Statistical learning theory*, New York: Wiley, 1998.
2. H. Güney, A fast-optimizing and adaptable intrusion detection system based on progressively optimized support vector machines, *Concurr. Comput. Pract. Exp.*, **37** (2025), e70156. https://doi.org/10.1002/cpe.70156
3. K. Ramu, S. Patthi, Y. N. Prajapati, J. V. N. Ramesh, S. Banerjee, K. B. V. Rao, et al., Hybrid CNN-SVM model for enhanced early detection of Chronic kidney disease, *Biomed. Signal Process. Control*, **100** (2025), 107084. https://doi.org/10.1016/j.bspc.2024.107084
4. J. K. Myilvahanan, N. M. Sundaram, Support vector machine-based stock market prediction using long short-term memory and convolutional neural network with aquila circle inspired optimization, *Netw. Comput. Neural Syst.*, **36** (2025), 1185–1220. https://doi.org/10.1080/0954898X.2024.2358957
5. G. H. Huang, T. G. Xue, W. H. Chen, L. L. Huang, Q. Dai, J. Y. Jiang, SVM-LncRNAPro: An SVM-based method for predicting long noncoding RNA promoters, *IET Syst. Biol*, **19** (2025), e70013. https://doi.org/10.1049/syb2.70013
6. C. Singh, N. Jain, N. Adlakha, K. R. Pardasani, Type-2 fuzzy support vector machine model for conformational epitope prediction, *Netw. Model. Anal. Health Inform. Bioinform.*, **14** (2025), 4. https://doi.org/10.1007/s13721-024-00498-7

7. H. J. Lin, H. S. H. Chung, C. X. Lin, D. Xie, Q. L. Deng, M. C. Lyu, et al., Improved fault diagnosis capability in CHBMCs: Counter design for multiple OC switches via an E-SVM unit, *IEEE Trans. Power Electron.*, **41** (2026), 2358–2376. https://doi.org/10.1109/TPEL.2025.3618228
8. A. Singhal, Seema, A. M. Saeed, R. Tiwari, A. Chaudhary, Hybrid fractional thermoelastic-machine learning framework for heat and mass transfer in skin tissue: Enhanced simulations using Atangana-Baleanu, Cattaneo-Vernotte models, and KNN-SVM classifiers, *Int. Commun. Heat Mass Transf.*, **171** (2026), 110074. https://doi.org/10.1016/j.icheatmasstransfer.2025.110074
9. T. Zhang, Statistical behavior and consistency of classification methods based on convex risk minimization, *Ann. Statist.*, **32** (2004), 56–85. https://doi.org/10.1214/aos/1079120130
10. I. Steinwart, Consistency of support vector machines and other regularized kernel classifiers, *IEEE Trans. Inf. Theory*, **51** (2005), 128–142. https://doi.org/10.1109/TIT.2004.839514
11. D. R. Chen, Q. Wu, Y. M. Ying, D. X. Zhou, Support vector machine soft margin classifiers: Error analysis, *J. Mach. Learn. Res.*, **5** (2004), 1143–1175.
12. I. Steinwart, A. Christmann, Fast learning from non-i.i.d. observations, In: *Proceedings of the 22nd international conference on neural information processing systems*, 2009, 1768–1776.
13. J. Xu, Y. Y. Tang, B. Zou, Z. B. Xu, L. Q. Li, Y. Lu, B. et al., The generalization ability of SVM classification based on Markov sampling, *IEEE Trans. Cybern.*, **45** (2015), 1169–1179. https://doi.org/10.1109/TCYB.2014.2346536
14. M. Kafai, K. Eshghi, CROification: Accurate kernel classification with the efficiency of sparse linear SVM, *IEEE Trans. Pattern Anal. Mach. Intell.*, **41** (2019), 34–48. https://doi.org/10.1109/TPAMI.2017.2785313
15. Y. Kong, Y. Fu, Max-margin action prediction machine, *IEEE Trans. Pattern Anal. Mach. Intell.*, **38** (2016), 1844–1858. https://doi.org/10.1109/TPAMI.2015.2491928
16. Y. W. Chang, C. J. Hsieh, K. W. Chang, M. Ringgaard, C. J. Lin, Training and testing low-degree polynomial data mappings via linear SVM, *J. Mach. Learn. Res.*, **11** (2010), 1471–1490.
17. S. Litayem, A. Joly, N. Boujemaa, Hash-based support vector machines approximation for large scale prediction, In: *Proceedings of the British machine vision conference*, 2012, 86.1–86.11. https://doi.org/10.5244/C.26.86
18. X. Q. Jiao, H. Lian, J. M. Liu, Y. Y. Zhang, Linear convergence of proximal gradient method for linear sparse SVM, *Neural Netw.*, **194** (2026), 108162. https://doi.org/10.1016/j.neunet.2025.108162
19. C. R. Rao, Y. Wu, Liner model selection by cross-validation, *J. Stat. Plan. Infer.*, **128** (2005), 231–240. https://doi.org/10.1016/j.jspi.2003.10.004
20. H. Li, Q. X. Huang, C. Wang, An early warning model for student status based on genetic algorithm-optimized radial basis kernel support vector machine, *J. Inf. Process. Syst.*, **20** (2024), 263–272. https://doi.org/10.3745/JIPS.02.0213
21. D. Anguita, A. Ghio, L. Oneto, S. Ridella, In-sample model selection for support vector machines, In: *The 2011 international joint conference on neural networks*, 2011, 1154–1161. https://doi.org/10.1109/IJCNN.2011.6033354

22. Y. X. Hu, H. T. Zhang, Chaos optimization method of SVM parameters selection for chaotic time series forecasting, *Phys. Procedia*, **25** (2012), 588–594. https://doi.org/10.1016/j.phpro.2012.03.130

23. A. Popov, A. Sautin, Selection of support vector machines parameters for regression using nested grids, In: *2008 Third international forum on strategic technologies*, 2008, 329–331. https://doi.org/10.1109/IFOST.2008.4602974

24. Z. Zhai, B. Gu, C. Deng, H. Huang, Global model selection via solution paths for robust support vector machine, *IEEE Trans. Pattern Anal. Mach. Intell.*, **47** (2025), 1331–1347. https://doi.org/10.1109/TPAMI.2023.3346765

25. J. C. Shao, X. N. Zhou, Q. K. Shao, H. L. Chen, B. J. Pan, A novel lymph node metastasis prediction method for gastric cancer: Enhanced support vector machine with polar lights optimization, *Biomed. Signal Process. Control*, **111** (2026), 108349. https://doi.org/10.1016/j.bspc.2025.108349

26. Y. L. Yuan, G. Y. Chong, J. J. Ren, W. Zhao, Y. A. Li, Z. X. Wang, et al., Musk ox optimizer (MO): A novel optimization algorithm and its application. *Cluster Comput.*, **28** (2025), 1041. https://doi.org/10.1007/s10586-025-05735-w

27. J. J. Zeng, Y. Duan, D. Wang, B. Zou, Y. Yin, J. Xu, Generalization performance of Lagrangian support vector machine based on Markov sampling, *J. Stat. Plan. Infer.*, **214** (2021), 89–104. https://doi.org/10.1016/j.jspi.2020.09.001

28. Z. Wang, T. Liang, B. Zou, Y. L. Cai, J. Xu, X. G. You, Incremental Fisher linear discriminant based on data denoising, *Knowl.-Based Syst.*, **237** (2022), 107799. https://doi.org/10.1016/j.knosys.2021.107799

29. S. Geman, D. Geman, Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Mach. Intell.*, **6** (1984), 721–741. https://doi.org/10.1109/TPAMI.1984.4767596

30. R. A. Fisher, The use of multiple measures in taxonomic problems, *Ann. Eugen.*, **7** (1936), 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

31. F. Cucker, S. Smale, On the mathematical foundations of learning, *Bull. Amer. Math. Soc.*, **39** (2002), 1–49. https://doi.org/10.1090/S0273-0979-01-00923-5

32. Q. Wu, Y. M. Ying, D. X. Zhou, Learning rates of least-square regularized regression, *Found. Comput. Math.*, **6** (2006), 171–192. https://doi.org/10.1007/s10208-004-0155-9

33. D. X. Zhou, Capacity of reproducing kernel spaces in learning theory, *IEEE Trans. Inf. Theory*, **49** (2003), 1743–1752. https://doi.org/10.1109/TIT.2003.813564

34. L. Devroye, L. Györfi, G. Lugosi, *A probabilistic theory of pattern recognition*, New York: Springer, 1996. https://doi.org/10.1007/978-1-4612-0711-5

35. H. Z. Tong, D. R. Chen, L. Z. Peng, Analysis of support vector machine regression, *Found. Comput. Math.*, **9** (2009), 243–257. https://doi.org/10.1007/s10208-008-9026-0

36. I. Steinwart, C. Scovel, Fast rates for support vector machines using Gaussian kernels, *Ann. Statist.*, **35** (2007), 575–607. https://doi.org/10.1214/009053606000001226

37. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, et al., LightGBM: A highly efficient gradient boosting decision tree, In: *Proceedings of the 31st international conference on neural information processing systems*, 2017, 3149–3157.

38. D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, *Nature*, **323** (1986), 533–536. https://doi.org/10.1038/323533a0

39. L. Breiman, Random Forests, *Mach. Learn.*, **45** (2001), 5–32. https://doi.org/10.1023/A:1010933404324

40. F. Wilcoxon, Individual comparisons by ranking methods, *Biom. Bull.*, **1** (1945), 80–83. https://doi.org/10.2307/3001968

Appendix A. Proof of proposition

Proof of Proposition 4.1. By Jensen's inequality, by $f_{\mathbf{z},\lambda} = \sum_{t=1}^{T} \alpha_t h_{\mathbf{z},\lambda_t}$, $\sum_{t=1}^{T} \alpha_t = 1$, we decompose the excess generalization error as follows:

$$\mathcal{R}(f_{\mathbf{z},\lambda}) - \mathcal{R}(f_\rho) \leq \sum_{t=1}^{T} \alpha_t \{\mathcal{R}(h_{\mathbf{z},\lambda_t}) - \mathcal{R}(f_\rho)\},$$

where

$$\begin{aligned}\mathcal{R}(f_{\mathbf{z},\lambda}) - \mathcal{R}(f_\rho) \leq \sum_{t=1}^{T} \alpha_t \{&\Delta_t + \mathcal{R}(h_{\lambda_t}) - \mathcal{R}(f_\rho) + \lambda_t \|h_{\lambda_t}\|_K^2 \\ &+ \mathcal{R}_{S_t}(h_{\mathbf{z},\lambda_t}) + \lambda_t \|h_{\mathbf{z},\lambda_t}\|_K^2 - \mathcal{R}_{S_t}(h_{\lambda_t}) - \lambda_t \|h_{\lambda_t}\|_K^2\}.\end{aligned}$$

From the definition of the function $h_{\mathbf{z},\lambda_t} (1 \leq t \leq T)$, we have $\mathcal{R}_{S_t}(h_{\mathbf{z},\lambda_t}) + \lambda_t \|h_{\mathbf{z},\lambda_t}\|_K^2 - \mathcal{R}_{S_t}(h_{\lambda_t}) - \lambda_t \|h_{\lambda_t}\|_K^2 \leq 0$. Noticing $\mathcal{D}(\lambda_t) = \mathcal{R}(h_{\lambda_t}) - \mathcal{R}(f_\rho) + \lambda_t \|h_{\lambda_t}\|_K^2$, the proof is completed. ■

Proof of Proposition 4.2. For any $h_{\lambda_t} \in \mathcal{H}_K$, we have $\lambda_t \|h_{\lambda_t}\|_K^2 \leq \mathcal{E}(h_{\lambda_t}) - \mathcal{E}(f_\rho) + \lambda_t \|h_{\lambda_t}\|_K^2 = \mathcal{D}(\lambda_t)$. It follows that $\|h_{\lambda_t}\|_\infty \leq \kappa \|h_{\lambda_t}\|_K \leq \kappa \sqrt{\mathcal{D}(\lambda_t)/\lambda_t}$. By the assumption $| f_\rho | \leq M$ and $\xi_{t,2} = \ell(h_{\lambda_t}) - \ell(f_\rho)$, we have that $| \xi_{t,2} | = | (1 - y h_{\lambda_t})_+ - (1 - y f_\rho)_+ | \leq d_t := \kappa \sqrt{\mathcal{D}(\lambda_t)/\lambda_t} + M$ and $E(\xi_{t,2}) \leq d_t$. By Lemma 4.1, we can conclude that for any $\varepsilon > 0$,

$$\mathrm{P}\Big\{ \frac{\frac{1}{N_t} \sum_{i=1}^{N_t} \xi_{t,2}(z_i) - \mathbb{E}(\xi_{t,2})}{\sqrt{\mathbb{E}(\xi_{t,2}) + \varepsilon}} \geq \sqrt{\varepsilon} \Big\} \leq \exp\Big\{ \frac{-N_t \varepsilon}{56 \|\Gamma'\|^2 d_t} \Big\}.$$

For any $\eta \in (0, 1)$, let $\eta = \exp\Big\{ \frac{-N_t \varepsilon}{56 \|\Gamma'\|^2 d_t} \Big\}$, and we have

$$\varepsilon = \frac{56 \ln(1/\eta) \|\Gamma'\|^2 d_t}{N_t}.$$

Using $\sqrt{\varepsilon} \sqrt{\mathbb{E}(\xi_{t,1}) + \varepsilon} \leq \frac{1}{2} \mathbb{E}(\xi_{t,1}) + \varepsilon$, the proof is completed. ■

Proof of Proposition 4.3. Let $\mathcal{G} = \{\mathrm{g}(z) = (1 - yh)_+ - (1 - y f_\rho)_+ : h \in \mathcal{B}_R\}$, and we have $\|h\|_\infty \leq \kappa \|h\|_K \leq \kappa R$, $0 < \mathrm{g}(z) \leq \kappa R + M$. By Lemma 4.1, we conclude that for any $\varepsilon > 0$,

$$\mathrm{P}\Big\{ \sup_{g \in \mathcal{G}} \frac{\mathbb{E}(\mathrm{g}) - \frac{1}{N_t} \sum_{i=1}^{N_t} \mathrm{g}(z_i)}{\sqrt{\mathbb{E}(\mathrm{g}) + \varepsilon}} \geq \sqrt{\varepsilon} \Big\} \leq \mathcal{N}(\mathcal{B}_R, \frac{\varepsilon}{4}) \exp\Big\{ \frac{-N_t \varepsilon}{896(\kappa R + M) \|\Gamma'\|^2} \Big\}.$$

By the definition of $h_{\mathbf{z},\lambda_t}$, we have

$$\mathrm{P}\Big\{\frac{\mathbb{E}(\xi_{t,1}) - \frac{1}{N_t}\sum_{i=1}^{N_t}\xi_{t,1}(z_i)}{\sqrt{\mathbb{E}(\xi_{t,1}) + \varepsilon}} \geq \sqrt{\varepsilon}\Big\} \leq \exp\{C_r(\frac{4R}{\varepsilon})^r + \frac{-N_t\varepsilon}{896(\kappa R + M)\|\Gamma'\|^2}\}.$$

Thus, for some positive constant $\tau > 0$ and $\varepsilon \geq N_t^{-\tau}$, from Definition 4.3, we have

$$\mathrm{P}\Big\{\frac{\mathbb{E}(\xi_{t,1}) - \frac{1}{N_t}\sum_{i=1}^{N_t}\xi_{t,1}(z_i)}{\sqrt{\mathbb{E}(\xi_{t,1}) + \varepsilon}} \geq \sqrt{\varepsilon}\Big\} \leq \exp\{C_r(4RN_t^{\tau})^r + \frac{-N_t\varepsilon}{896(\kappa R + M)\|\Gamma'\|^2}\}. \tag{A.1}$$

For the same η, let the righthand side of (A.1) be η, and by $\sqrt{\varepsilon}\sqrt{\mathbb{E}(\xi_{t,1}) + \varepsilon} \leq \frac{1}{2}\mathbb{E}(\xi_{t,1}) + \varepsilon$, if $N_t \geq (896(\kappa R + M)\|\Gamma'\|^2\{C_r(4R)^r + \ln(1/\eta)\})^{1/(1-\tau-\tau r)}$, inequality

$$\begin{aligned}\mathbb{E}(\xi_{t,1}) - \frac{1}{N_t}\sum_{i=1}^{N_t}\xi_{t,1}(z_i) \leq& \frac{1}{2}[\mathcal{R}(h_{\mathbf{z},\lambda_t}) - \mathcal{R}(f_\rho)] \\ &+ \frac{896(\kappa R + M)\|\Gamma'\|^2\{C_r(4RN_t^{\tau})^r + \ln(1/\eta)\}}{N_t}\end{aligned}$$

is valid with confidence for at least $1 - \eta$. The proof is completed. ■

Appendix B. Proof of theorem

Proof of Theorem 4.1. Take $R \leq M/\kappa$, and by Propositions 4.2-4.3, we get that with confidence $1 - 2\eta$, and the inequality

$$\begin{aligned}\mathcal{R}(h_{\mathbf{z},\lambda_t}) - \mathcal{R}(f_\rho) \leq& \frac{3584M^{r+1}C_r\|\Gamma'\|^2 4^r\kappa^{-r}}{N_t^{1-\tau r}} + 3\mathcal{D}(\lambda_t) \\ &+ \frac{112\ln(1/\eta)\|\Gamma'\|^2(\kappa\sqrt{\mathcal{D}(\lambda_t)/\lambda_t} + 33M)}{N_t}\end{aligned}$$

holds provided that $N_t \geq (1792M\|\Gamma'\|^2(C_r(4M/\kappa)^r + \ln(2/\eta)))^{1/(1-\tau-\tau r)}$. By Proposition 4.1, we have

$$\begin{aligned}\mathcal{R}(f_{\mathbf{z},\lambda}) - \mathcal{R}(f_\rho) \leq \sum_{t=1}^{T}\alpha_t\Big\{&\frac{3584M^{r+1}C_r\|\Gamma'\|^2 4^r\kappa^{-r}}{N_t^{1-\tau r}} + 3\mathcal{D}(\lambda_t) \\ &+ \frac{112\ln(2T/\eta)\|\Gamma'\|^2(\kappa\sqrt{\mathcal{D}(\lambda_t)/\lambda_t} + 33M)}{N_t}\Big\}\end{aligned}$$

holds with confidence $1 - \eta$, as $N_t \geq N^* := (1792M\|\Gamma'\|^2(C_r(4M/\kappa)^r + \ln(2T/\eta)))^{1/(1-\tau-\tau r)}$. The proof is completed. ■

Proof of Theorem 4.2. Take $\lambda_t = (tN_t)^{-\beta}$ with $\beta > 0$, and by Theorem 4.1 and the assumption $\mathcal{D}(\lambda_t) \leq C_s\lambda_t^s$, we have

$$\mathcal{R}(f_{\mathbf{z},\lambda}) - \mathcal{R}(f_\rho) \leq \sum_{t=1}^{T}\alpha_t\Big\{\frac{3584M^{r+1}C_r\|\Gamma'\|^2 4^r\kappa^{-r}}{N_t^{1-\tau r}} + 3C_s(tN_t)^{-s\beta}$$

$$+\frac{112\ln(2T/\eta)\|\Gamma'\|^2(\kappa C_s^{1/2}(tN_t)^{(1-s)\beta/2}+33M)}{N_t}\Big\}.$$

Let $c'=\ln(2T/\eta)\|\Gamma'\|^2$ and $N=\min_t N_t$, and by $\sum_{t=1}^{T}\alpha_t=1$, we can get

$$\mathcal{R}(f_{\mathbf{z},\lambda})-\mathcal{R}(f_\rho)\leq\frac{3584M^{r+1}C_r\|\Gamma'\|^2 4^r\kappa^{-r}}{N^{1-\tau r}}+3C_s(N)^{-s\beta}+\frac{112c'(\kappa C_s^{1/2}(NT)^{(1-s)\beta/2}+33M)}{N}.$$

If $T\leq N$, for any $0<\tau<\frac{(1-s)\beta}{r}$, $1<\beta<\frac{1}{1-s}$, we have

$$\mathcal{R}(f_{\mathbf{z},\lambda})-\mathcal{R}(f_\rho)\leq(c_1+c_2+c_3+3C_s)(N)^{(1-s)\beta-1},$$

where $c_1=3584M^{r+1}C_r\|\Gamma'\|^2 4^r\kappa^{-r}, c_2=3696c'M, c_3=112c'\kappa C_s^{1/2}$. Using inequality (4.1), we conclude that for any $\eta\in(0,1)$, the inequality

$$L(\mathrm{sgn}(f_{\mathbf{z},\lambda}))-L^*\leq C\Big(\frac{1}{N}\Big)^{1-(1-s)\beta}$$

is valid with confidence for at least $1-\eta$ provided that $N\geq\max\{N^*,T\}$. Here, $C=c_1+c_2+3C_s+c_3$ is a constant. The proof is completed. ■

AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)