Information Sciences 753 (2026) 123615

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A semi-supervised deep forest framework based on margin distribution optimization for tabular data

Shen-Huan Lyu [a, b, d], Jia-Le Xu [a], Yi-Xiao He [c,*], Yanyan Wang [a, d], Baoliu Ye [d,*], Qingfu Zhang [b]

[a] Key Laboratory of Water Big Data Technology of Ministry of Water Resources, College of Computer Science and Software Engineering, Hohai University, Nanjing, 211100, China
[b] Department of Computer Science, City University of Hong Kong, 518057, Hong Kong, China
[c] School of Artificial Intelligence and Information Technology, Nanjing University of Chinese Medicine, Nanjing, 210023, China
[d] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210023, China

HIGHLIGHTS

- Semi-supervised deep forest for tabular data via margin distribution.
- Maximizes labeled margins and minimizes unlabeled margin variance jointly.
- Theoretical analysis proves that margin distribution improves generalization.
- SSDF outperforms semi-supervised baselines with limited labeled samples.

ARTICLE INFO

Keywords:
Semi-supervised learning
Deep forest
Margin theory

ABSTRACT

Deep Forest (DF) is a non-differentiable deep learning model based on decision tree ensembles. As an alternative to deep neural networks, it demonstrates superior suitability for dealing with structured high-dimensional data while inherently offering interpretability. However, like many deep learning paradigms, DF often requires a substantial amount of labeled data to achieve optimal performance, posing a significant challenge in real-world scenarios where labeled samples are scarce. To address this limitation, this paper proposes a novel semi-supervised learning framework for DF, focusing on optimizing the margin distribution of both labeled and unlabeled samples. We introduce a new method to maximize the average margin of labeled data and minimize the margin variance of unlabeled data, thereby enhancing the model's generalization capability theoretically. Extensive experiments on various datasets demonstrate that our proposed semi-supervised Deep Forest (SSDF) can outperform existing semi-supervised baselines under conditions of limited labeled data.

1. Introduction

Deep learning has achieved remarkable success across various domains, particularly in computer vision [1] and natural language processing [2]. The field primarily focuses on deep neural network architectures, i.e., sophisticated models comprising multiple layers of parameterized nonlinear differentiable modules trained by backpropagation [3]. However, recognizing the inherent limitations

* Corresponding authors.
Email addresses: lvsh@hhu.edu.cn (S.-H. Lyu), xujl@hhu.edu.cn (J.-L. Xu), heyx@njucm.edu.cn (Y.-X. He), yanyan.wang@hhu.edu.cn (Y. Wang), yebl@nju.edu.cn (B. Ye), qingfu.zhang@cityu.edu.hk (Q. Zhang).

https://doi.org/10.1016/j.ins.2026.123615
Received 29 October 2025; Received in revised form 8 May 2026; Accepted 8 May 2026
Available online 9 May 2026

of differentiable models in processing structured sparse high-dimensional data, Zhou and Feng [4] propose a novel deep forest framework named gcForest (multi-Grained Cascade Forest). This groundbreaking approach leverages decision tree ensembles [5] rather than neural networks, utilizing non-differentiable modules without backpropagation training.

While deep forest has demonstrated remarkable performance in structured data processing [6], its reliance on large-scale labeled datasets remains a critical bottleneck, especially in domains where expert annotations are costly or privacy-sensitive. For instance, user reluctance to manually label preferred commodities results in extremely sparse annotated datasets due to excessive labeling costs [7]. Similarly, in medical diagnostics [8], stringent data privacy regulations fundamentally restrict the accessibility of labeled patient cases, as clinical annotations often contain sensitive protected health information. Therefore, the practical implementations of deep forest rely on integrating semi-supervised learning methods [9].

Semi-supervised learning methodologies aim to empower classifiers to exploit latent structures within unlabeled data for performance enhancement autonomously. Existing semi-supervised tree ensembles (e.g., Co-training [10], Tri-training [11]) primarily leverage diversity among multiple classifiers to improve confidence in pseudo-labels generated for unlabeled samples. However, these conventional approaches typically focus on single-layer model architectures, relying on data randomness to maintain classifier diversity [12], thus failing to harness the hierarchical feature transformation capability inherent in deep forest models. Moreover, maintaining sufficient classifier diversity becomes increasingly challenging as the number of classifiers scales up.

In contrast, deep forest with cascade architecture demonstrates unique advantages in representation learning. This hierarchical structure consists of multiple levels, each containing an ensemble of decision tree forests [5,13], i.e., an ensemble of ensembles. Each cascade level receives enhanced features generated from its preceding level as input. This more complex model structure enhances its representational capacity but also increases its susceptibility to overfitting when labeled samples are insufficient. Moreover, due to the sequential nature of the cascade architecture, directly employing pseudo-labels from unlabeled data for self-learning may cause estimation errors to accumulate across layers, degrading feature representation quality. To address these challenges, this paper proposes a semi-supervised deep forest framework based on margin distribution optimization. An ensemble pruning module is integrated into each layer of the deep forest to maximize the margin mean of labeled samples and minimize the margin variance of unlabeled samples, thereby reducing overfitting risk and mitigating noise accumulation while improving classifier alignment with the distribution of unlabeled data. The main contributions of this work are summarized as follows:

- We address the critical challenge of Deep Forest's performance degradation under scenarios of labeled sample scarcity, providing a robust solution for data-limited environments.
- We propose a novel semi-supervised learning framework for Deep Forest (SSDF), specifically introducing a margin distribution optimization strategy that simultaneously maximizes the average margin of labeled data and minimizes the margin variance of unlabeled data, theoretically enhancing generalization.
- Extensive empirical evaluations demonstrate that SSDF significantly surpasses the performance of existing semi-supervised baselines, particularly when labeled data is limited.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 introduces the preliminary background of SSL. Section 4 details the proposed SSDF framework. Section 5 theoretically analyzes the impact of margin distribution on generalization. Section 6 presents experimental results. Section 7 concludes with future directions.

2. Related work

Semi-supervised learning. Semi-supervised learning (SSL) aims to improve learning performance by leveraging a large amount of unlabeled data and a limited amount of labeled data [14]. Traditional SSL approaches can be classified into several main categories based on the role of unlabeled data, namely generative models, pseudo-label-based models, regularization-based models, and their combinations. Classical statistical learning methods model the data generation process and utilize both labeled and unlabeled data, including Gaussian Mixture Models (GMM) [15] and Variational Autoencoders (VAE) [16]. Another line of work generates pseudo-labels for unlabeled data based on predictive models and iteratively trains by expanding the supervisory signal through strategies such as self-training [17] and co-training [11]. Other approaches introduce various regularization terms (such as large-margin regularization [18], Laplacian regularization [19], manifold regularization [20], etc.) to constrain the model using the geometric structure of unlabeled data, ensuring that the prediction function maintains smooth consistency across the data distribution. In recent years, the combination of semi-supervised learning and deep neural networks has become increasingly close, but their training processes all rely on gradient computation [21]. How to integrate with a non-differentiable deep learning framework, such as deep forest, remains an open problem [4]. Broader surveys on SSL methods are available in [9].

Deep forest variants. Emerging as a viable alternative to neural networks, deep forests have demonstrated remarkable versatility in structured data learning across various domains. Seminal works by Utkin and Ryabinin [22] pioneered metric learning adaptations, while their subsequent development of imprecise models replacing exact class probability estimates at decision tree leaves effectively addressed few-shot classification challenges [6]. Practical implementations further validate its adaptability: Zhang et al. [23] achieved state-of-the-art performance in automatic detection for cash-out fraud, Wu et al. [24] advanced synergistic drug combination prediction, and Liu et al. [25] enhanced groundwater recharge modeling accuracy. Hybrid architectures integrating deep forests with neural components [26] have successfully bridged the gap to computer vision tasks, confirming the paradigm's multidimensional utility.

Meanwhile, theoretical analyses have further strengthened deep forest frameworks. Lyu et al. [27] demonstrated that optimizing empirical margin distributions during cascade learning enhances predictive performance. Subsequent studies established that deep

forest-generated feature representations improve consistency convergence rates in random forests [28,29]. Layer-wise sample screening [30] and enhanced feature diversity generation [31] jointly reduce computational and storage overhead, providing lightweight solutions for edge deployment.

While existing deep forest methods achieve promising performance, their reliance on large-scale labeled datasets limits applicability in real-world scenarios where labeled samples are scarce. Recent advancements address this limitation by reformulating DF as differentiable neural network architectures, enabling semi-supervised training frameworks that effectively leverage both labeled and unlabeled data through novel loss functions [32]. However, these approaches exhibit limitations when applied to structured data domains such as financial risk assessment and medical diagnosis, primarily stemming from their architectural design, which reformulates DF into DNN.

3. Preliminary

In semi-supervised learning, we consider an input feature space $\mathcal{X} \subseteq \mathbb{R}^d$ and an output label space $\mathcal{Y} = \{1, \ldots, C\}$ for a C-class classification task. The learning paradigm involves two distinct data components: a limited set of labeled instances $S_l = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$ where each $\boldsymbol{x}_i \in \mathcal{X}$ is paired with its corresponding label $y_i \in \mathcal{Y}$, and a substantially larger collection of unlabeled examples $S_u = \{\boldsymbol{x}_j\}_{j=m+1}^{m+n}$ containing only input observations $\boldsymbol{x}_j \in \mathcal{X}$. In real-world scenarios, due to the high cost of labeling, the amount of unlabeled data far exceeds that of labeled data.

The goal is to learn an L-layer deep forest model $F^{(L)}$ that effectively separates labeled samples from different classes while maintaining distributional compatibility with unlabeled samples. Due to the high model complexity of deep forest, the limited number of labeled samples can easily lead to overfitting during training. This may lead to collapsed feature representations, characterized by insufficient diversity, which consequently results in performance degradation. Given unlabeled data, a common approach is to generate pseudo-labels based on predictions from a supervised model trained on labeled data. We can denote the unlabeled samples as $S_u = \{(\boldsymbol{x}_j, \hat{y}_j)\}_{j=m+1}^{m+n}$, where $\hat{y}_j$ represents the pseudo-label. Although this formulation unifies labeled and unlabeled samples in form, the pseudo-labels are often subject to substantial inaccuracies due to biases in the supervised model and inherent variance in the data distribution. In the cascade structure of deep forest, such errors propagate layer by layer, degrading the quality of feature representations, since feature learning in deep forest relies heavily on label information.

4. The proposed method

In this section, we present SSDF for semi-supervised classification tasks. Rather than using a regularization that maximizes the minimum margin in existing work [33], SSDF employs the cascade forest structure, which transforms new features layer-by-layer, to optimize the entire margin distribution simultaneously. It consists of two steps: margin distribution optimization and cascade forest construction.

4.1. Margin distribution optimization

When Deep Forest leverages both labeled data and unlabeled data, it typically encounters two major challenges: the risk of overfitting due to the scarcity of labeled samples and the propagation of pseudo-label noise across layers when utilizing unlabeled data. The former limits the model's ability to learn stable decision boundaries, while the latter may amplify pseudo-labeling errors as the cascade deepens.

Inspired by Lyu et al. [27], we note that the overall margin distribution plays a more critical role in determining generalization performance than the minimal margin alone. As illustrated in Fig. 1, this perspective provides a unified explanation for the two challenges above: For labeled samples, maximizing the mean margin enhances the model's confidence in correct classifications and alleviates overfitting. For unlabeled samples, minimizing the margin variance allows the model to exploit distributional information from pseudo-labeled data while mitigating the adverse impact of label noise.

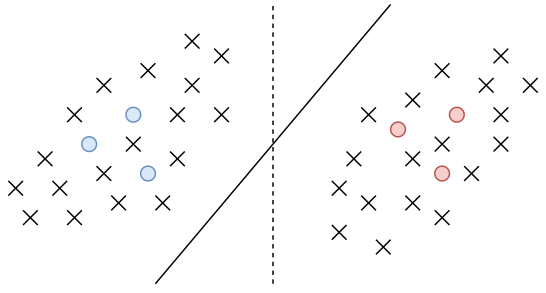

Fig. 1. Illustration of margin distribution and boundaries in semi-supervised learning. Blue circles represent labeled positive samples, red circles represent labeled negative samples, and '×' represents unlabeled samples. The dashed line represents the classification boundary obtained by maximizing the average margin of labeled samples, while the solid line represents the classification boundary obtained by maximizing the average margin of labeled samples while minimizing the margin variance of unlabeled samples.

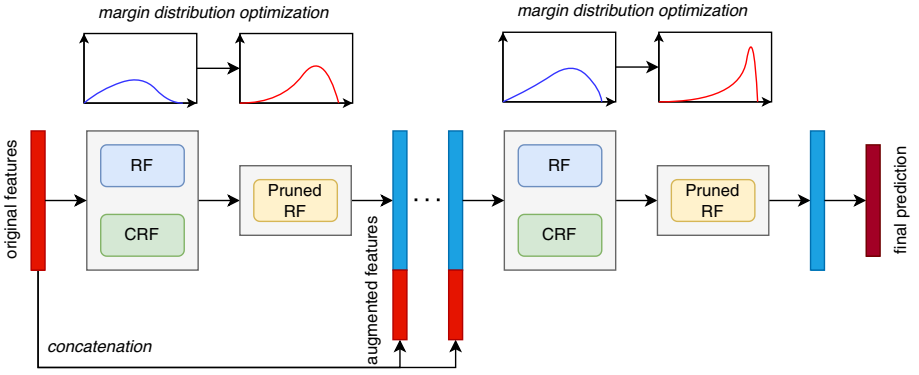

Fig. 2. Illustration of the semi-supervised deep forest structure. The margin distribution is optimized layer by layer through the ensemble pruning module.

Based on this insight, we introduce a Margin Distribution Optimization (MDO) framework as the core objective of each layer, jointly guiding the representation learning of labeled and unlabeled data to achieve a more robust semi-supervised deep forest architecture.

As shown in Fig. 2, each layer of deep forest consists of a forest module composed of two types of random forests: random forest (RF) and completely random forest (CRF). To simplify, we can consider each forest model as an ensemble of a series of randomized decision trees, denoted as $h = \{h_t\}_{t=1}^{N}$, where $h_t : \mathcal{X} \to [0,1]^C$ represents the t-th decision tree in the forest module at each layer, which outputs a C-dimensional class probability vector.

Ensemble pruning [34] can optimize specific objectives (such as validation accuracy, ensemble size, etc.) to improve the predictive performance of the ensemble learner. Therefore, we introduce an ensemble pruning framework to optimize the aforementioned two margin distribution objectives. Since the ensemble model before pruning is not saved, we continue to use h to denote the ensemble pruned by the selection vector $\boldsymbol{s} \in \{0,1\}^N$, where $s_i = 0/s_i = 1$ indicates that the i-th decision tree is unselected/selected. Then, the output of the forest module h for sample $\boldsymbol{x}$ is the average of all selected decision trees, i.e.,

$$h(\boldsymbol{x}) = \frac{1}{|\boldsymbol{s}|} \sum_{t=1}^{N} s_t h_t(\boldsymbol{x}) \,, \tag{1}$$

where $|\boldsymbol{s}| = \sum_{t=1}^{N} s_t$ denotes the number of selected decision trees, i.e., the ensemble size.

As shown in Fig. 1, to achieve better generalization performance, it is necessary to maximize the average margin on labeled samples and minimize the margin variance on unlabeled samples on the validation set. The hypothesis learned by the forest module h aims to predict the class by selecting the highest probability in the probability vector, i.e., $\hat{y} = \arg\max_i h^i(\boldsymbol{x})$, where h^i denotes the probability of the i-th class. Subsequently, we can establish the following definition of the *margin* γ_h of a labeled example $(\boldsymbol{x}, y)$:

$$\gamma_h(\boldsymbol{x}, y) = h^y(\boldsymbol{x}) - \max_{i \neq y} h^i(\boldsymbol{x}). \tag{2}$$

Thus, h misclassifies $(\boldsymbol{x}, y)$ iff $\gamma_h(\boldsymbol{x}, y) < 0$.

Let $S_l' = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m'}$ and $S_u = \{(\boldsymbol{x}_i, \hat{y}_i)\}_{i=1}^{n}$ represent the given labeled validation set and unlabeled set, respectively. For any ensemble model h, its average margin on labeled samples can be expressed as:

$$o^{avg}(h, S_l') = \frac{1}{m'} \sum_{i=1}^{m'} \gamma_h(\boldsymbol{x}_i, y_i) \,, \tag{3}$$

and its margin variance on unlabeled samples can be expressed as:

$$o^{var}(h, S_u) = \frac{1}{2n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\gamma_h(\boldsymbol{x}_i, \hat{y}_i) - \gamma_h(\boldsymbol{x}_j, \hat{y}_j)\right)^2 . \tag{4}$$

The larger the margin mean of labeled samples, the higher the confidence of the ensemble model's predictions, which can better alleviate the risk of overfitting caused by insufficient labeled data. The smaller the margin variance of unlabeled samples, the higher the compatibility between the ensemble model and the overall data distribution, meaning that the probability scores given by the ensemble model for more similar samples are also more similar. Based on the above analysis, we determined $-o^{avg}$ and o^{var} as the two optimization objectives for ensemble pruning. Additionally, considering that the forest module requires a sufficient number of base learners to provide diversity in feature representation, it is also necessary to control the ensemble size $|\boldsymbol{s}| = N/2$ as an additional objective, i.e.,

$$o^{size}(h) = \left| |\boldsymbol{s}| - N/2 \right| . \tag{5}$$

The ensemble size objective o^{size} is introduced to explicitly control the trade-off between bias, variance, and diversity in the pruned ensemble. Retaining too few trees may reduce diversity and increase bias, whereas retaining too many trees may preserve redundant

learners and increase variance. In our implementation, the target size is set to $|s| = N/2$, which provides a balanced trade-off between these factors.

Finally, we formulate ensemble pruning in each layer as a tri-objective minimization problem

$$\arg\min_{s\in\{0,1\}^N} \; (o^{size}(h)), -o^{avg}(h, S_l'), o^{var}(h, S_u) \; . \tag{6}$$

By using NSGA-III to solve the tri-objective problem defined in Eq. (6), we can optimize the margin distribution in each forest module of the deep forest. After the NSGA-III generates a Pareto-optimal set at each cascade layer, a single solution is selected using a validation-based, lexicographic rule. The primary criterion is the model size objective o^{size}, which prioritizes solutions whose tree size is closest to the target to enable fair comparison. Among feasible solutions, the one with the largest average margin o^{avg} on labeled validation samples is chosen to improve accuracy. If needed, the o^{var} on unlabeled samples is used as a tie-breaker to reduce the generalization gap. It is noted that minimizing the first two objectives can leverage labeled and unlabeled data, respectively, to enhance the generalization performance of the forest module, while combining with the third objective can generate more diverse feature representations. To the best of our knowledge, this is the first time the ensemble pruning framework has been applied to optimize deep forest models. Furthermore, we will provide a theoretical analysis in the next section on how the two objectives, o^{avg} and o^{var}, affect the generalization error of the ensemble model.

4.2. Cascade forest construction

The deep forest architecture employs an adaptive cascade structure, as illustrated in Fig. 2. Each level consists of an ensemble of different decision tree forests, i.e., an ensemble of ensembles. Each layer of the cascade sequentially concatenates raw training data with feature representations from preceding layers, generating new feature representations through its forest-generated class probability vectors. Given a test sample, decision trees at each level produce class probability estimates by counting the class distribution of labeled training samples in their corresponding leaf nodes. The final output of each forest is the average prediction across all trees in the same forest. The cascade employs 3-fold cross-validation during the generation of feature representations to alleviate the risk, and automatically terminates if there is no significant performance gain on the validation set. This cascade structure progressively refines feature representations through ensemble pruning modules across cascaded layers.

In the 3-fold cross-validation protocol, only the labeled dataset is partitioned into folds. For each fold, two folds form the labeled training set S_l, and the remaining fold is used as the labeled validation set S_l', which is also employed for pruning and cascade stopping. The unlabeled dataset S_u is not split into folds; instead, for each fold, the model predicts labels (or margins) for all unlabeled samples. The predictions from the three folds are then averaged to obtain stable estimates used in the unlabeled pruning objective. Unlabeled data are not used for supervised training, but only for evaluating margin-based criteria during pruning.

In the proposed cascade framework, pseudo-labels for unlabeled samples are updated at each layer. Specifically, after the pruned ensemble is obtained at layer t, it is used to predict class labels for unlabeled samples, and the resulting pseudo-labels are passed to the next layer. These pseudo-labeled samples are then used when training the unpruned forest at layer $t+1$. No confidence-based filtering is applied, since the pseudo-labels are only used to estimate the margin distribution on unlabeled data (e.g., margin variance), rather than to provide precise supervisory signals. This design avoids additional hyperparameters and maintains a distribution-level regularization effect.

Feature concatenation is the key mechanism in deep forest, distinguishing it from traditional forest algorithms that operate solely in the original feature space. It endows deep forest with the ability to evolve layer by layer. However, in semi-supervised learning, the scarcity of labeled data results in augmented features, such as class probability vectors, derived from labeled information statistics lacking diversity. As is well known, insufficient diversity can impair the predictive performance of ensembles. To enhance the diversity of augmented features generated by each forest module in deep forest, we employ randomized decision trees of varying depths as base learners within the forest. These randomized decision trees of different depths can extract labeled information from the data across multiple scales of spatial partitioning, thereby providing sufficiently diverse feature representations for a large amount of unlabeled data.

The cascade structure of deep forest can be characterized by a pair $(\boldsymbol{h}, \boldsymbol{f})$ as follows:

- $\boldsymbol{h} = (h^{(1)}, \ldots, h^{(L)})$, where $h^{(\ell)}$ denotes the ensemble of different forests at level ℓ and $h^{(\ell)}$ belongs to the hypothesis class $\mathcal{H}_\ell$,
- $\boldsymbol{f} = \{f^{(1)}, \ldots, f^{(L)}\}$, where $f^{(\ell)}$ denotes the cascade of forest ensembles up to level ℓ.

Then, the relationship between $\boldsymbol{h}$ and $\boldsymbol{f}$ can be formulated at each level:

$$f^{(\ell)}(\boldsymbol{x}) = \begin{cases} h^{(1)}(\boldsymbol{x}), & \ell = 1 \; , \\ h^{(\ell)}(\boldsymbol{x} \oplus f^{(\ell-1)}(\boldsymbol{x})), & \ell > 1 \; , \end{cases} \tag{7}$$

where $\boldsymbol{a} \oplus \boldsymbol{b}$ denotes the concatenation of $\boldsymbol{a}$ and $\boldsymbol{b}$ to form a new feature vector. At each layer, the pruned forest outputs a fixed-dimensional augmented feature vector, which is concatenated with the original input features. From the second layer onward, the feature dimensionality remains constant and equals the original feature dimension plus the augmented feature dimension from the previous layer. As a result, feature growth is strictly controlled and does not increase with cascade depth, reducing the risk of overfitting. In summary, a pair $(\boldsymbol{h}, \boldsymbol{f})$ defines a deep forest model F as follows

$$F(\boldsymbol{x}) = \arg\max_{c\in\{1,2,\ldots,C\}} \left[f^{(L)}(\boldsymbol{x})\right]^c \; , \tag{8}$$

Algorithm 1 Semi-Supervised Deep Forest (SSDF).

Input: Labeled set $S_l = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$, unlabeled set $S_u = \{\boldsymbol{x}_j\}_{j=1}^{n}$, number of layers L
Output: Trained cascade model F

Initialize feature representation $\boldsymbol{x}^{(1)} = \boldsymbol{x}$
for $\ell = 1$ to L **do**
  Train each forest $f^{(\ell)}$ on current representation $x^{(\ell)}$
  Use MOEA to prune forests in $\{f^{(\ell)}\}$ by considering three objectives as follows:

- o^{avg}: maximize average margin on labeled data defined by Eq. (3)
- o^{var}: minimize margin variance on unlabeled data defined by Eq. (4)
- o^{size}: control number of trees per layer defined by Eq. (5)

  Obtain Pareto-optimal set $\mathcal{P}_\ell$
  Select $\mathbf{s}_\ell^* \in \mathcal{P}_\ell$ by lexicographic validation-based ranking:
    minimize $o^{\text{size}}(\mathbf{s})$, maximize $o^{\text{avg}}(\mathbf{s})$, and minimize $o^{\text{var}}(\mathbf{s})$
  Construct pruned ensemble $f^{(\ell)}$ using $\mathbf{s}_\ell^*$
  Compute probability vectors $f^{(\ell)}(\boldsymbol{x}^{(\ell)})$ for all samples
  Concatenate features to form next-layer input: $\boldsymbol{x}^{(\ell+1)} = \boldsymbol{x} \oplus f^{(\ell)}(\boldsymbol{x}^{(\ell)})$
  Update pseudo-labels for all samples in S_u using the pruned ensemble
end for
return Final hypothesis $F(\boldsymbol{x}) = \arg\max_{c\in\{1,2,\ldots,C\}} \left[f^{(L)}(\boldsymbol{x})\right]^c$

where $[f^{(L)}(\boldsymbol{x})]^c$ denotes the c-th element of the class probability vector $f^{(L)}(\boldsymbol{x})$. The complete algorithm is shown in Algorithm 1.

5. Theoretical analysis

In this section, we provide theoretical justification for margin distribution optimization in SSDF. Specifically, we prove an upper bound for generalization risk, based on the notion of Rademacher Complexity [35].

We assume that all instances $\boldsymbol{x}$, including labeled and unlabeled instances, are i.i.d. drawn from the feature space $\mathcal{X}$ according to a fixed underlying distribution D. They are labeled by some unknown target concept c^*, i.e., the true label y satisfies $y = c^*(\boldsymbol{x})$. According to the augmented PAC model for SSL [14], a hypothesis space is a set of functions over the instance space $\mathcal{X}$, and we make the assumption (the "realizable case") that the target function belongs to the hypothesis space $\mathscr{H}$ in this work. For any hypothesis $h \in \mathscr{H}$, the *expected risk* is defined over the underlying distribution D,

$$R(h) = \mathbb{E}_{\boldsymbol{x}\sim D}\left[\mathbb{1}(h(\boldsymbol{x}) \neq c^*(\boldsymbol{x}))\right] . \tag{9}$$

Given a labeled training set $S_l = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$, for any hypothesis $h \in \mathscr{H}$, we define the *empirical margin loss* as

$$L_{S_l,\rho}(h) = \frac{1}{m}\sum_{i=1}^{m} \ell(\gamma_h(\boldsymbol{x}_i, y_i)), \tag{10}$$

where $\ell(\gamma) = \min(1, \max(0, 1 - \gamma/\rho))$ is a standard margin loss function [35].

Furthermore, it is essential to establish a compatibility function that assesses the degree to which a hypothesis h aligns with the underlying distribution D. Ideally, there should be a strong alignment between target concepts c^* and underlying distributions D. If the target concept is unrelated to the data distribution, it is possible to generate numerous unlabeled instances from a uniform distribution over the feature space $\mathcal{X}$. However, these unlabeled instances obtained in such a manner hold no significance for learning the target concept. Therefore, we define a notion of *margin distribution compatibility* to be a mapping from a hypothesis h and a distribution D to $[0, 1]$ indicating how "compatible" h is with D from the perspective of margin distribution.

Definition 1 (Margin Loss of Unlabeled Training Set). For any hypothesis $h \in \mathscr{H}$, the *empirical compatibility* is defined over the unlabeled training set S_u,

$$\chi(h, S_u) = \frac{1}{2n(n-1)}\sum_{i=m+1}^{m+n-1}\sum_{j=i+1}^{m+n} (\gamma_h(\boldsymbol{x}_i, \hat{y}_i) - \gamma_h(\boldsymbol{x}_j, \hat{y}_j))^2 . \tag{11}$$

Definition 2 (Unlabeled Risk). For any hypothesis $h \in \mathscr{H}$, denote by $\hat{y} = \arg\max_i[h^i(\boldsymbol{x})]$ the predicted class of $\boldsymbol{x}$. The unlabeled risk of h is defined as the variance of the margin on unlabeled samples drawn from D when the true label is replaced by the model prediction:

$$R_u(h) = \mathrm{Var}_{\boldsymbol{x}\sim D}[\gamma_h(\boldsymbol{x}, \hat{y})] = \mathbb{E}_{\boldsymbol{x}\sim D}\left[\gamma_h(\boldsymbol{x}, \hat{y})^2\right] - \left(\mathbb{E}_{\boldsymbol{x}\sim D}\left[\gamma_h(\boldsymbol{x}, \hat{y})\right]\right)^2 . \tag{12}$$

A smaller $R_u(h)$ indicates that the margin distribution induced by h is more consistent with the geometry of the unlabeled data distribution, i.e., higher compatibility between h and D in terms of margin distribution.

In this paper, we will use Rademacher complexity to measure the size of the hypothesis set, and then use it to establish a uniform convergence upper bound for expected risk.

Definition 3 (Rademacher Complexity [35]). Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ to $\mathbb{R}$, and a fixed sample of size m as $S = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_m\}$, where $\boldsymbol{z}_i = \{\boldsymbol{x}_i, y_i\} \in \mathcal{Z}$. Then, the *empirical Rademacher complexity* of $\mathcal{G}$ with respect to the sample S is defined as:

$$\hat{\mathfrak{R}}_S(\mathcal{G}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i g(\boldsymbol{z}_i)\right] , \tag{13}$$

where $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_m)^\top$, and σ_i is an independent uniform random variable taking values in $\{-1, +1\}$.

Theorem 1. *Let $\mathscr{H}$ be a family of functions represented by the deep forest architecture. Let $\rho > 0$ denote the margin parameter used in the margin loss function defined in Eq. (10). Then, for any $\delta > 0$, with probability at least $1 - \delta$, each of the following holds for all $h \in \mathscr{H}$:*

$$R_u(h) \le \chi(h, S_u) + 16\hat{\mathfrak{R}}_{S_u}(\mathscr{H}) + 9\sqrt{\tfrac{\log(4/\delta)}{2n}} , \tag{14}$$

let $t = 16\hat{\mathfrak{R}}_S(\mathscr{H}) + 9\sqrt{\log(4/\delta))/(2n)}$ and $\tau > 0$ be a given threshold of unlabeled empirical risk, we have

$$R(h) \le L_{S_l,\rho}(h) + \frac{2}{\rho}\hat{\mathfrak{R}}_S\left(\mathscr{H}_{D,\chi}(t+\tau)\right) + 3\sqrt{\frac{\log\frac{1}{\delta}}{2m}} , \tag{15}$$

where $\mathscr{H}_{D,\chi}(t+\tau) = \{h \in \mathscr{H} : R_u(h) \le t + \tau\}$.

Proof. Let S_u be i.i.d. unlabeled instances from D. For $h \in \mathscr{H}$, define

$$\mu_1(h) = \mathbb{E}[\gamma_h(\boldsymbol{x}, \hat{y})], \qquad \mu_2(h) = \mathbb{E}[\gamma_h(\boldsymbol{x}, \hat{y})^2], \tag{16}$$

and their empirical counterparts

$$\hat{\mu}_1(h) = \tfrac{1}{n}\sum_{i=1}^{n} \gamma_h(\boldsymbol{x}_i, \hat{y}_i), \quad \hat{\mu}_2(h) = \tfrac{1}{n}\sum_{i=1}^{n} \gamma_h(\boldsymbol{x}_i, \hat{y}_i)^2, \tag{17}$$

where $\hat{y}_i = \arg\max_c [h(\boldsymbol{x}_i)]_c$. Then

$$R_u(h) = \mu_2(h) - \mu_1(h)^2. \tag{18}$$

Let $\tilde{\mathscr{H}} = \{x \mapsto \gamma_h(\boldsymbol{x}, \hat{y}) : h \in \mathscr{H}\}$ and $\tilde{\mathscr{H}}^{(2)} = \{x \mapsto \gamma_h(\boldsymbol{x}, \hat{y})^2 : h \in \mathscr{H}\}$.

By Rademacher complexity bounds, for any $\delta_2 > 0$, with probability at least $1 - \delta_2$,

$$|\mu_2(h) - \hat{\mu}_2(h)| \le 2\hat{\mathfrak{R}}_{S_u}(\tilde{\mathscr{H}}^{(2)}) + 3\sqrt{\tfrac{\log(2/\delta_2)}{2n}}. \tag{19}$$

Since $\phi(u) = u^2$ is 2-Lipschitz on $[-1, 1]$, Talagrand's contraction lemma gives $\hat{\mathfrak{R}}_{S_u}(\tilde{\mathscr{H}}^{(2)}) \le 2\hat{\mathfrak{R}}_{S_u}(\tilde{\mathscr{H}})$. Further, $\hat{\mathfrak{R}}_{S_u}(\tilde{\mathscr{H}}) \le 2\hat{\mathfrak{R}}_{S_u}(\mathscr{H})$, hence

$$|\mu_2(h) - \hat{\mu}_2(h)| \le 8\hat{\mathfrak{R}}_{S_u}(\mathscr{H}) + 3\sqrt{\tfrac{\log(2/\delta_2)}{2n}}. \tag{20}$$

Similarly, for any $\delta_1 > 0$, with probability at least $1 - \delta_1$,

$$|\mu_1(h) - \hat{\mu}_1(h)| \le 4\hat{\mathfrak{R}}_{S_u}(\mathscr{H}) + 3\sqrt{\tfrac{\log(2/\delta_1)}{2n}}. \tag{21}$$

Then, using $|a^2 - b^2| = |a - b||a + b|$ and $|\mu_1|, |\hat{\mu}_1| \le 1$,

$$|\mu_1(h)^2 - \hat{\mu}_1(h)^2| \le 2|\mu_1(h) - \hat{\mu}_1(h)| \le 8\hat{\mathfrak{R}}_{S_u}(\mathscr{H}) + 6\sqrt{\tfrac{\log(2/\delta_1)}{2n}}. \tag{22}$$

Hence

$$\mu_1(h)^2 \ge \hat{\mu}_1(h)^2 - \left(8\hat{\mathfrak{R}}_{S_u}(\mathscr{H}) + 6\sqrt{\tfrac{\log(2/\delta_1)}{2n}}\right). \tag{23}$$

Taking $\delta_1 = \delta_2 = \delta/2$ and applying the union bound, with probability at least $1 - \delta$,

$$R_u(h) = \mu_2(h) - \mu_1(h)^2 \tag{24}$$

$$\le \hat{\mu}_2(h) - \hat{\mu}_1(h)^2 + 16\hat{\mathfrak{R}}_{S_u}(\mathscr{H}) + 9\sqrt{\tfrac{\log(4/\delta)}{2n}} . \tag{25}$$

Let $a_i = \gamma_h(x_i, \hat{y}_i)$. The identity

$$\frac{1}{n(n-1)}\sum_{i<j}(a_i - a_j)^2 = \frac{n}{n-1}\left(\frac{1}{n}\sum_{i=1}^{n} a_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} a_i\right)^2\right) \tag{26}$$

implies

$$\hat{\mu}_2(h) - \hat{\mu}_1(h)^2 = \frac{n-1}{n}\chi(h, S_u). \tag{27}$$

Therefore,

$$\begin{aligned} R_u(h) &\le \frac{n-1}{n}\chi(h, S_u) + 16\hat{\mathfrak{R}}_{S_u}(\mathscr{H}) + 9\sqrt{\frac{\log(4/\delta)}{2n}} \\ &\le \chi(h, S_u) + 16\hat{\mathfrak{R}}_{S_u}(\mathscr{H}) + 9\sqrt{\frac{\log(4/\delta)}{2n}}\ . \end{aligned} \tag{28}$$

The above analysis establishes that the unlabeled risk $R_u(h)$ can be tightly controlled by its empirical counterpart $\chi(h, S_u)$ together with a complexity penalty term. This result is crucial because it allows us to constrain the hypothesis space to those classifiers whose margin distribution is sufficiently *compatible* with the unlabeled data, i.e., satisfying $R_u(h) \le t + \tau$ for some threshold.

In the next step, we leverage this constraint to derive a generalization bound on the supervised 0-1 risk $R(h)$. Concretely, by restricting the hypothesis class to $\mathscr{H}_{D,\chi}(t+\tau) = \{h \in \mathscr{H} : R_u(h) \le t + \tau\}$, the effective capacity of the learner is reduced, which in turn decreases the Rademacher complexity term appearing in the supervised margin bound. As a result, the combination of the two bounds, one for unlabeled risk and one for supervised risk, shows that minimizing the empirical margin loss on labeled data together with the variance term $\chi(h, S_u)$ on unlabeled data jointly leads to a tighter overall generalization guarantee.

Let $\widetilde{\mathscr{H}} = \{\boldsymbol{x} \mapsto \gamma_h(\boldsymbol{x}, y) : \ h \in \mathscr{H}_{D,\chi}(t+\tau)\}$ be the family of functions represented by the *margin* of labeled data with deep forest model. Consider the family of functions taking values in $[0, 1]$:

$$\widetilde{\mathcal{H}} = \{\Phi_\rho \circ \gamma : \gamma \in \widetilde{\mathscr{H}}\}\ . \tag{29}$$

By theorem 3.3 in [36], with probability at least $1 - \delta$, for all $g \in \widetilde{\mathcal{H}}$, we have

$$\mathbb{E}[g(\boldsymbol{x})] \le \frac{1}{n}\sum_{i=1}^{n} g(\boldsymbol{x}_i) + 2\mathfrak{R}_S(\widetilde{\mathcal{H}}) + 3\sqrt{\frac{\log\frac{1}{\delta}}{2m}}\ , \tag{30}$$

therefore, for all $h \in \mathscr{H}$, we have

$$\mathbb{E}[\Phi_\rho(\gamma_h(\boldsymbol{x}, y))] \le L_{S_l,\rho}(h) + 2\mathfrak{R}_S(\Phi_\rho \circ \widetilde{\mathscr{H}}) + 3\sqrt{\frac{\log\frac{1}{\delta}}{2m}}\ , \tag{31}$$

Since $\mathbb{1}_x \le \Phi_\rho(x)$ for all $x \in \mathbb{R}$, we have $R(h) = \mathbb{E}[\mathbb{1}_{\gamma_h(\boldsymbol{x},y)\le 0}] \le \mathbb{E}[\Phi_\rho(\gamma_h(\boldsymbol{x}, y))]$, thus

$$R(h) \le L_{S_l,\rho}(h) + 2\mathfrak{R}_S(\Phi_\rho \circ \widetilde{\mathscr{H}}) + 3\sqrt{\frac{\log\frac{1}{\delta}}{2m}}\ , \tag{32}$$

Since Φ_ρ is $1/\rho$-Lipschitz, by Talagrand's lemma, we have $\mathfrak{R}_S(\Phi_\rho \circ \widetilde{\mathscr{H}}) \le \frac{1}{\rho}\mathfrak{R}_S(\widetilde{\mathscr{H}})$ and $\mathfrak{R}_S(\widetilde{\mathscr{H}})$ satisfies the equation as follows:

$$\begin{aligned} \mathfrak{R}_S(\widetilde{\mathscr{H}}) &= \frac{1}{m}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\gamma_h \in \widetilde{\mathscr{H}}} \sum_{i=1}^{n} \sigma_i \gamma_h(\boldsymbol{x}_i, y_i)\right] \\ &= \frac{1}{m}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\gamma_h \in \widetilde{\mathscr{H}}} \sum_{i=1}^{n} \sigma_i (f_{y_i}(\boldsymbol{x}_i) - \max_{j \ne y_i} f_j(\boldsymbol{x}_i))\right] \\ &\le \frac{1}{m}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f_1, f_2 \in \mathscr{H}} \sum_{i=1}^{n} \sigma_i |f_1(\boldsymbol{x}_i) - f_2(\boldsymbol{x}_i)|\right]\ . \end{aligned} \tag{33}$$

Combining (31), (32) and (33), we prove the generalization bound of labeled risk (15). □

Remark 1. The above generalization bound provides an intuitive justification for the proposed margin distribution optimization strategy. The bound decreases when the average margin on labeled samples is increased, which encourages confident and well-separated predictions on observed labels. At the same time, controlling the variance of margins on unlabeled samples prevents overly unstable or extreme predictions on unseen data, effectively regularizing the decision function. From a practical perspective, this result explains why jointly maximizing labeled margins and minimizing unlabeled margin variance leads to better generalization, especially in low-label regimes. The proposed pruning objectives are therefore directly aligned with the terms that govern the theoretical generalization behavior of the model.

6. Experiments

In this section, we present experimental results on several real-world datasets. To evaluate the performance level of SSDF, we compared it with a series of state-of-the-art (SOTA) semi-supervised learning methods, including Tri-Training [11], Co-Training [10], Assemble [9], FixMatch [37], and TabNet [38]. Additionally, we demonstrate ablation studies, parameter analyses, and visualization results to confirm the effectiveness of our proposed margin distribution optimization.

6.1. Experiments on real-world datasets

Datasets. We conducted experiments on ten UCI datasets. Table 1 summarizes the detailed information of these ten datasets, with sample sizes ranging from 8124 to 253,680, feature dimensions from 8 to 124, and the number of classes from 2 to 10. Each dataset was randomly split into training and test sets in a 7:3 ratio, with only 5% of the training samples being labeled and the rest being unlabeled. To ensure a fair comparison, in all experiments, each dataset split was used for the proposed SSDF method and other compared methods, with 10 repeated random splits corresponding to 10 independent runs of the training and evaluation process.

Compared methods and hyper-parameters. In the proposed SSDF, we employ one random forest [5] and one completely random forest [4] in each layer's forest module. For each forest type, decision trees with depths $\{2, 4, 6\}$ are used, and 20 trees are trained for each depth, resulting in 60 trees per forest type and 120 trees in total per cascade layer. The pruning stage applies a tri-objective multi-objective evolutionary algorithm to select a subset of trees from this pool. In our experiments, the pruning target is set to $N/2 = 60$, where N denotes the total number of candidate trees. An ablation study on different pruning targets is reported in Table 2, showing that the method is not sensitive to this parameter. The construction of the cascade forest is evaluated using 3-fold cross-validation, and the cascade growth stops if increasing the depth does not improve validation performance. For fairness, we replace the base classifiers in Tri-Training, Co-Training, and Assemble with random forests, setting the depth and number of decision trees the same as in SSDF, while other parameters use default values. FixMatch is implemented using a multilayer perceptron (MLP) backbone with three fully connected layers, each containing 256 neurons and ReLU activation. The batch size is selected from $\{64, 128\}$, the number of training epochs from $\{10, 50, 100\}$, and optimization is performed using SGD with learning rates in $\{2\times10^{-2}, 1\times10^{-2}, 2\times10^{-3}\}$ and momentum values in $\{0.7, 0.8, 0.9\}$. The best configuration on the validation set is reported. In addition, TabNet [38] is included as a tabular-specific deep learning baseline. The learning rate is tuned from $\{2\times10^{-2}, 1\times10^{-2}, 2\times10^{-3}\}$, the dimensions (n_d, n_a) from $\{8, 32, 64\}$, the number of fine-tuning epochs from $\{50, 100\}$, and the pretraining batch size from $\{256, 1024\}$. Other parameters are fixed: pretraining epochs = 50, early stopping patience = 10, virtual batch size = 128, and pretraining ratio = 0.8. During fine-tuning, the batch size is set to 256. TabPFN [39] is not included due to its inability to scale to the dataset sizes considered in this work. We apply Gaussian noise with intensity levels of 0.3/0.15 as strong/weak data augmentation schemes for tabular data. Similar to [37], we adopt the same training protocol, including optimizer, learning rate scheduling, data preprocessing, random seeds, etc. All experiments are conducted on the LAMDA-SSL platform [40] run on a machine with a 3.40 GHz Intel i7-13700KF CPU, and the code will be made available upon acceptance.

Evaluation protocol. In all experiments, model performance was measured by the classification accuracy of test samples. For each dataset, the train-test split and labeled-unlabeled split were randomly generated 10 times, reporting the mean and standard deviation of performance, with statistical significance assessed using the paired Wilcoxon signed-rank test at a confidence level of $\alpha = 0.05$. Furthermore, to handle multiple-method comparisons over multiple datasets, we applied the Friedman-Nemenyi test with a confidence level of $\alpha = 0.05$.

Main results. We report in Table 3 the test accuracy of the proposed SSDF method compared with other baseline methods on 10 real-world datasets. The results show that the proposed SSDF achieves the best overall test accuracy on these 10 datasets, with an average rank of 1.5, which is the highest among all baseline methods. Additionally, the deep learning method FixMatch, which uses neural networks as its backbone, ranks second in terms of comprehensive performance. This indicates that the inherent representation learning capability of deep learning enables it to outperform traditional semi-supervised learning models. Meanwhile, the tree-based

Table 1
Information about the datasets.

Dataset	# Samples	# Features	# Classes
Adult	48,842	124	2
BankMarketing	45,211	37	2
CreditCard	30,000	23	2
Diabetes	253,680	21	2
Gamma	19,020	10	2
HTRU2	17,898	8	2
Mushroom	8124	117	2
RTIoT2022	123,052	91	10
Statlog	57,927	7	4
Websites	11,055	30	2

Table 2
Comparison across different pruning targets. The values in parentheses indicate the corresponding numbers of selected trees (pruning targets).

Dataset	120(40)	120(60)	120(80)
Adult	0.8365 ± 0.0072	0.8319 ± 0.0065	0.8193 ± 0.0088
BankMarketing	0.8904 ± 0.0039	0.8907 ± 0.0049	0.8869 ± 0.0043
CreditCard	0.8130 ± 0.0070	0.8085 ± 0.0086	0.8065 ± 0.0125

Table 3
Comparison of Accuracy, AUROC, and macro-F1 for different SSL methods on ten benchmark datasets (average ± standard deviation of 10 runs). The best result is in **bold**, the second best result is in underline. Considering the paired Wilcoxon signed-rank test with a confidence level of $\alpha = 0.05$, where • (or ∘) indicates that SSDF is significantly better (or worse) than the corresponding method.

Algorithm	Adult	BankMarketing	CreditCard	Diabetes	Gamma	HTRU2	Mushroom	RTIoT2022	Statlog	Websites	Avg. Rank
Accuracy ↑											
SSDF	**0.8319 ± 0.0065**	**0.8907 ± 0.0049**	0.8085 ± 0.0086	0.8612 ± 0.0005	**0.8378 ± 0.0079**	0.9758 ± 0.0028	0.9886 ± 0.0079	**0.9857 ± 0.0023**	**0.9971 ± 0.0005**	**0.9288 ± 0.0056**	1.50
TriTraining	0.8114 ± 0.0061•	0.8878 ± 0.0034	**0.8086 ± 0.0031**	0.8608 ± 0.0001	0.8198 ± 0.0093•	0.9761 ± 0.0025	0.9799 ± 0.0114	0.9833 ± 0.0022•	0.9939 ± 0.0016•	0.9256 ± 0.0070	2.70
CoTraining	0.7717 ± 0.0036•	0.8841 ± 0.0018•	0.7873 ± 0.0053•	0.8607 ± 0.0000•	0.7663 ± 0.0090•	0.9606 ± 0.0075•	0.9664 ± 0.0118•	0.9801 ± 0.0062•	0.9105 ± 0.0079•	0.7946 ± 0.0371•	5.10
Assemble	0.7607 ± 0.0000•	0.8830 ± 0.0000•	0.7828 ± 0.0088•	0.8607 ± 0.0000•	0.7244 ± 0.0068•	0.9685 ± 0.0041•	0.9422 ± 0.0186•	0.9591 ± 0.0097•	0.9857 ± 0.0053•	0.8970 ± 0.0229•	5.50
FixMatch	0.8212 ± 0.0044•	0.8826 ± 0.0034•	0.7931 ± 0.0045•	**0.8616 ± 0.0006**	0.8252 ± 0.0077•	**0.9768 ± 0.0016**	**0.9902 ± 0.0069**	0.9849 ± 0.0032	0.9867 ± 0.0024•	0.9221 ± 0.0041•	2.60
TabNet	0.8140 ± 0.0082•	0.8839 ± 0.0079•	0.8016 ± 0.0065	0.8607 ± 0.0026	0.8057 ± 0.0242•	0.9690 ± 0.0159	0.9809 ± 0.0205	0.9841 ± 0.0040	0.9926 ± 0.0086	0.8863 ± 0.0194•	3.60
AUROC ↑											
SSDF	**0.8993 ± 0.0030**	**0.8907 ± 0.0041**	**0.7668 ± 0.0051**	**0.8165 ± 0.0023**	**0.8936 ± 0.0054**	**0.9665 ± 0.0051**	**0.9994 ± 0.0008**	**0.9994 ± 0.0003**	**0.9598 ± 0.0149**	**0.9798 ± 0.0019**	1.00
TriTraining	0.8848 ± 0.0027•	0.8833 ± 0.0075•	0.7658 ± 0.0048	0.8127 ± 0.0030•	0.8806 ± 0.0058•	0.9643 ± 0.0054•	0.9983 ± 0.0009•	0.9982 ± 0.0009•	0.9586 ± 0.0120	0.9796 ± 0.0015	2.40
CoTraining	0.8816 ± 0.0026•	0.8454 ± 0.0150•	0.7518 ± 0.0102•	0.8143 ± 0.0020•	0.8735 ± 0.0043•	0.9556 ± 0.0060•	0.9967 ± 0.0014•	0.9979 ± 0.0014•	0.9578 ± 0.0109	0.9514 ± 0.0140•	3.40
Assemble	0.8424 ± 0.0096•	0.7881 ± 0.0260•	0.6644 ± 0.0358•	0.7957 ± 0.0074•	0.7321 ± 0.0425•	0.9508 ± 0.0073•	0.9825 ± 0.0129•	0.9930 ± 0.0020•	0.9494 ± 0.0150	0.9703 ± 0.0040•	5.60
FixMatch	0.8626 ± 0.0053•	0.8164 ± 0.0101•	0.7072 ± 0.0114•	0.8098 ± 0.0017•	0.8822 ± 0.0075•	0.9648 ± 0.0094	0.9950 ± 0.0037•	0.9928 ± 0.0036•	0.9436 ± 0.0180•	0.9760 ± 0.0018•	4.20
TabNet	0.8525 ± 0.0126•	0.8289 ± 0.0233•	0.7208 ± 0.0122•	0.8066 ± 0.0078•	0.8725 ± 0.0133•	0.9644 ± 0.0101	0.9951 ± 0.0065•	0.9936 ± 0.0030•	0.9507 ± 0.0158	0.9388 ± 0.0132•	4.40
macro-F1 ↑											
SSDF	0.7062 ± 0.0211	0.5611 ± 0.0624	0.6201 ± 0.0459	0.4707 ± 0.0093	**0.8126 ± 0.0107**	0.9215 ± 0.0117	0.9886 ± 0.0079	0.8417 ± 0.0226	0.7989 ± 0.0416	**0.9275 ± 0.0058**	2.50
TriTraining	0.6413 ± 0.0244•	0.5202 ± 0.0356	0.6255 ± 0.0146	0.4638 ± 0.0016	0.7798 ± 0.0166•	0.9226 ± 0.0099	0.9799 ± 0.0115	0.8042 ± 0.0235•	0.7451 ± 0.0017•	0.9242 ± 0.0074	3.50
CoTraining	0.4790 ± 0.0150•	0.4818 ± 0.0200•	0.4962 ± 0.0353•	0.4626 ± 0.0000•	0.6865 ± 0.0142•	0.8560 ± 0.0364•	0.9663 ± 0.0119•	0.7697 ± 0.0700•	0.6393 ± 0.0130•	0.7706 ± 0.0503•	5.30
Assemble	0.4321 ± 0.0000•	0.4689 ± 0.0000•	0.4658 ± 0.0602•	0.4626 ± 0.0000•	0.6104 ± 0.0219•	0.8893 ± 0.0173•	0.9418 ± 0.0191•	0.5968 ± 0.0909•	0.7371 ± 0.0054•	0.8945 ± 0.0261•	5.60
FixMatch	**0.7530 ± 0.0042**∘	**0.6709 ± 0.0096**∘	**0.6582 ± 0.0089**∘	0.4816 ± 0.0123∘	0.7918 ± 0.0106•	**0.9252 ± 0.0065**	**0.9902 ± 0.0069**	**0.9079 ± 0.0248**∘	0.7461 ± 0.0132•	0.9211 ± 0.0041•	1.60
TabNet	0.7353 ± 0.0146∘	0.6628 ± 0.0258∘	0.6479 ± 0.0228∘	**0.5735 ± 0.0391**∘	0.7879 ± 0.0216•	0.8896 ± 0.0912	0.9809 ± 0.0206	0.8729 ± 0.0332∘	**0.8117 ± 0.0578**	0.8832 ± 0.0206•	2.50

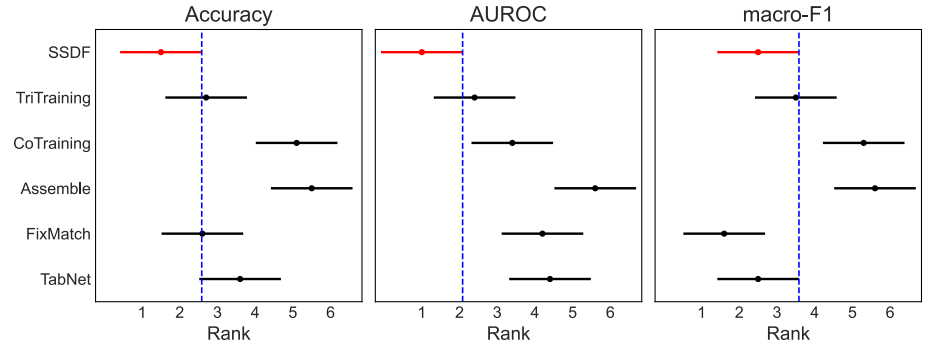

Fig. 3. Friedman-Nemenyi test of the compared methods on ten datasets.

Table 4
Average number of cascade layers determined by the validation-based stopping criterion for each dataset.

Dataset	Adult	BankMarketing	CreditCard	Diabetes	Gamma	HTRU2	Mushroom	RTIoT2022	Statlog	Websites
Avg. Layers	3.40	3.80	2.70	4.40	4.70	4.00	5.00	1.70	7.00	2.50

Table 5
Ablation study on margin distribution optimization and pseudo labels.

Dataset	SSDF	$DF_{\text{raw label}}$	$DF_{\text{pseudo label}}$
Adult	**0.8301 ± 0.0067**	0.7965 ± 0.0062	0.7607 ± 0.0000
BankMarketing	**0.8906 ± 0.0046**	0.8856 ± 0.0035	0.8832 ± 0.0006
CreditCard	**0.8096 ± 0.0060**	0.7999 ± 0.0097	0.7855 ± 0.0084
Diabetes	**0.8610 ± 0.0005**	0.8607 ± 0.0001	0.8607 ± 0.0000
Gamma	**0.8358 ± 0.0054**	0.8355 ± 0.0060	0.8320 ± 0.0083
HTRU2	**0.9753 ± 0.0032**	0.9745 ± 0.0041	0.9714 ± 0.0030
Mushroom	**0.9892 ± 0.0045**	0.9858 ± 0.0062	0.9844 ± 0.0080
RTIoT2022	**0.9867 ± 0.0011**	0.9521 ± 0.0067	0.9290 ± 0.0074
Statlog	**0.9971 ± 0.0004**	0.9961 ± 0.0008	0.9964 ± 0.0009
Websites	**0.9292 ± 0.0057**	0.9276 ± 0.0074	0.9263 ± 0.0076
Avg. Rank	1.00	2.09	2.91

models in SSDF are more suitable for representing structured features in tabular datasets, which is why SSDF outperforms FixMatch on more datasets. In addition to accuracy, we report AUROC and macro-F1 averaged over 10 runs for all datasets, which are particularly important for imbalanced classification tasks. These results demonstrate that SSDF achieves strong and stable performance across different evaluation metrics.

To assess statistical significance, we adopt the paired Wilcoxon signed-rank test at a significance level of $\alpha = 0.05$ in Table 3, which is appropriate for paired comparisons across repeated runs on identical data splits. Furthermore, to handle multiple-method comparisons over multiple datasets, we apply the Friedman-Nemenyi test with a confidence level of $\alpha = 0.05$. The resulting critical difference diagram (Fig. 3) reports average ranks across datasets and highlights statistically significant differences between methods.

The cascade is grown layer by layer using a validation-based stopping criterion. At each layer, performance on the labeled validation set S_l' is evaluated, and cascade growth stops when no further improvement is observed. An upper bound $L_{\max}$ is imposed but is rarely reached in practice. Table 4 reports the average number of cascade layers for each dataset.

6.2. Secondary evaluation

Ablation study. In Table 5, we compare the performance of the proposed SSDF method with DF trained using only labeled samples ($DF_{\text{raw label}}$) and DF trained using labeled samples along with pseudo-labeled unlabeled samples ($DF_{\text{pseudo label}}$) across ten datasets. It can be observed that SSDF consistently outperforms these two baseline methods, demonstrating that optimizing the margin distribution can enhance the performance of DF in semi-supervised learning tasks. Additionally, it is noted that $DF_{\text{pseudo label}}$ does not always surpass $DF_{\text{raw label}}$, which further validates the conclusion from our theoretical analysis: pseudo-labeling is not always beneficial for semi-supervised learning; positive improvement is only achievable when the compatibility between the classifier and the data distribution is good enough.

Different labeled ratios. To evaluate robustness under varying levels of supervision, we further report the performance of SSDF under multiple labeled data ratios, including 1%, 2%, 5%, and 10%. Table 6 summarizes Accuracy, macro-F1, and AUROC on representative datasets, showing consistent performance improvements as the amount of labeled data increases.

Table 6
Performance of SSDF under different labeled data ratios evaluated by Accuracy, macro-F1, and AUROC.

Ratio	Adult			BankMarketing			CreditCard		
	Accuracy	macro-F1	AUROC	Accuracy	macro-F1	AUROC	Accuracy	macro-F1	AUROC
0.01	0.8256 ± 0.0170	0.7022 ± 0.0671	0.8841 ± 0.0068	0.8845 ± 0.0055	0.5666 ± 0.0761	0.8534 ± 0.0123	0.8015 ± 0.0102	0.6049 ± 0.0505	0.7328 ± 0.0185
0.02	0.8314 ± 0.0089	0.7123 ± 0.0341	0.8925 ± 0.0041	0.8894 ± 0.0046	0.5568 ± 0.0629	0.8779 ± 0.0088	0.8076 ± 0.0066	0.6321 ± 0.0480	0.7523 ± 0.0081
0.05	0.8319 ± 0.0065	0.7062 ± 0.0211	0.8993 ± 0.0030	0.8907 ± 0.0049	0.5611 ± 0.0624	0.8907 ± 0.0041	0.8085 ± 0.0086	0.6201 ± 0.0459	0.7668 ± 0.0051
0.1	0.8298 ± 0.0092	0.6954 ± 0.0310	0.9025 ± 0.0021	0.8895 ± 0.0034	0.5367 ± 0.0334	0.8955 ± 0.0051	0.8110 ± 0.0065	0.6271 ± 0.0355	0.7725 ± 0.0050

Table 7
The test accuracy (average ± standard deviation of 10 times of running) of SSDF using different MOEAs on ten classification datasets. The best results on each dataset are highlighted in bold.

Dataset	NSGA-III	NSGA-II	MOEA/D	awGA
Adult	0.8301 ± 0.0067	0.8197 ± 0.0122	**0.8391 ± 0.0085**	0.8273 ± 0.0097
BankMarketing	**0.8906 ± 0.0046**	0.8866 ± 0.0035	0.8901 ± 0.0045	0.8872 ± 0.0045
CreditCard	**0.8096 ± 0.0060**	0.8083 ± 0.0098	0.8092 ± 0.0075	0.8072 ± 0.0101
Diabetes	0.8610 ± 0.0005	**0.8612 ± 0.0005**	0.8609 ± 0.0004	0.8607 ± 0.0002
Gamma	0.8358 ± 0.0054	**0.8383 ± 0.0074**	0.8373 ± 0.0074	0.8380 ± 0.0065
HTRU2	0.9753 ± 0.0032	0.9754 ± 0.0028	0.9753 ± 0.0030	**0.9757 ± 0.0024**
Mushroom	0.9892 ± 0.0045	0.9874 ± 0.0073	**0.9899 ± 0.0070**	0.9887 ± 0.0058
RTIoT2022	0.9867 ± 0.0011	**0.9877 ± 0.0015**	0.9872 ± 0.0016	0.9835 ± 0.0050
Statlog	0.9971 ± 0.0004	**0.9972 ± 0.0004**	0.9970 ± 0.0004	0.9972 ± 0.0004
Websites	0.9292 ± 0.0057	0.9286 ± 0.0062	0.9280 ± 0.0066	**0.9299 ± 0.0061**
Avg. Rank	2.36	2.36	2.45	2.82

Different evolutionary algorithms. Since SSDF can employ various multi-objective evolutionary algorithms (MOEAs) to solve the three-objective problem in Eq. (6), we first compared the performance of SSDF equipped with NSGA-III [41], NSGA-II [42], MOEA/D [43], and awGA [44]. For a fair comparison, we used the same hyperparameter settings for each MOEA: a population size of 100 and 500 generations. The crossover probability P_c and mutation probability P_m were arbitrarily set to 0.7 and 1, respectively. A more careful configuration might yield better results.

Table 7 presents the average and standard deviation of the test accuracy and ensemble size for each method on each dataset. Table 7 uses the name of the MOEA to represent the SSDF method equipped with that algorithm. For example, NSGA-III actually refers to SSDF equipped with NSGA-III. Table 7 shows that using different evolutionary algorithms has little effect on the performance of SSDF. From the average ranking of performance, it can be found that the four evolutionary algorithms have no significant difference in predictive performance on the ten classification datasets.

To examine the influence of NSGA-III hyperparameters, we conduct a sensitivity analysis on the crossover probability P_c and mutation probability P_m. Table 8 reports results under different combinations of $P_c \in \{0.7, 0.9, 1.0\}$ and $P_m \in \{0.05, 0.1, 0.2\}$. The performance differences across these settings are minor, indicating that the proposed method is robust to the choice of NSGA-III parameters. Therefore, the default settings are adopted throughout the experiments without additional tuning.

Margin distribution analysis. Since SSDF explicitly optimizes the margin distribution, we also visualize the margin distributions of layer 0 and the final layer by plotting the probability density of the margin distribution. Fig. 4 shows the results of SSDF on the Mushroom and Websites datasets. It can be observed that the ensemble model selected after margin distribution optimization has a larger mean margin and a smaller margin variance. Moreover, this advantage becomes more pronounced as the depth increases in DF, which also indicates that the layer-by-layer feature transformation and ensemble pruning in deep forest help progressively improve the compatibility between the model and the data. Therefore, SSDF achieves an overall better margin distribution, demonstrating its superior generalization performance, which is consistent with previous observations.

Running time analysis. Although o^{var} is initially defined via pairwise margin differences on unlabeled samples, we adopt its equivalent variance-form expression in practice. Specifically, o^{var} is computed as the empirical variance of margins on the unlabeled set, which can be evaluated in linear time with respect to the number of unlabeled samples. As a result, the computational cost of the proposed method is dominated by the multi-objective optimization process rather than by forest training, as shown in Tables 9 and 10.

Table 8
Sensitivity analysis of SSDF w.r.t. NSGA-III crossover (P_c) and mutation (P_m) probabilities.

P_m	P_c	Adult			BankMarketing			CreditCard		
		Accuracy	macro-F1	AUROC	Accuracy	macro-F1	AUROC	Accuracy	macro-F1	AUROC
default(0.05)	default(1)	0.8319 ± 0.0065	0.7062 ± 0.0211	0.8993 ± 0.0030	0.8907 ± 0.0049	0.5611 ± 0.0624	0.8907 ± 0.0041	0.8085 ± 0.0086	0.6201 ± 0.0459	0.7668 ± 0.0051
0.05	0.7	0.8343 ± 0.0051	0.7137 ± 0.0177	0.8995 ± 0.0029	0.8907 ± 0.0044	0.5576 ± 0.0518	0.8910 ± 0.0060	0.8109 ± 0.0065	0.6424 ± 0.0387	0.7653 ± 0.0045
0.05	0.9	0.8347 ± 0.0078	0.7156 ± 0.0262	0.8994 ± 0.0032	0.8920 ± 0.0050	0.5727 ± 0.0588	0.8914 ± 0.0054	0.8084 ± 0.0066	0.6244 ± 0.0412	0.7669 ± 0.0042
0.1	0.7	0.8325 ± 0.0061	0.7080 ± 0.0208	0.8995 ± 0.0024	0.8885 ± 0.0041	0.5340 ± 0.0495	0.8909 ± 0.0042	0.8091 ± 0.0082	0.6279 ± 0.0471	0.7657 ± 0.0039
0.1	0.9	0.8277 ± 0.0061	0.6918 ± 0.0219	0.8985 ± 0.0034	0.8904 ± 0.0040	0.5550 ± 0.0464	0.8905 ± 0.0052	0.8109 ± 0.0069	0.6388 ± 0.0355	0.7658 ± 0.0046
0.2	0.7	0.8270 ± 0.0092	0.6888 ± 0.0328	0.8991 ± 0.0028	0.8884 ± 0.0045	0.5298 ± 0.0520	0.8906 ± 0.0051	0.8066 ± 0.0082	0.6106 ± 0.0438	0.7661 ± 0.0046
0.2	0.9	0.8255 ± 0.0109	0.6835 ± 0.0385	0.8984 ± 0.0025	0.8866 ± 0.0045	0.5125 ± 0.0525	0.8898 ± 0.0059	0.8097 ± 0.0078	0.6325 ± 0.0435	0.7658 ± 0.0045

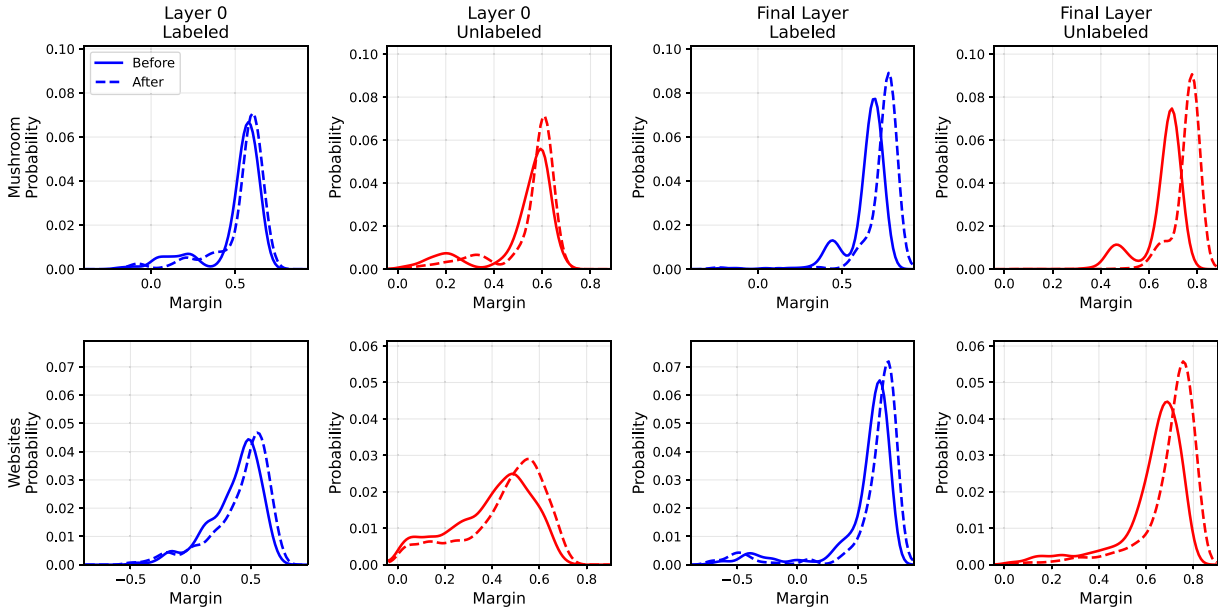

Fig. 4. The probability density plots of the margin distribution obtained by SSDF on the Mushroom and Websites datasets. Blue curves indicate the results on labeled samples, while red curves indicate the results on unlabeled samples. Solid lines represent the results before margin distribution optimization, and dashed lines represent the results after optimization.

Table 9
Comparison of training time among different SSL methods.

Dataset	SSDF	TriTraining	CoTraining	Assemble	FixMatch	TabNet
Adult	799.25s	0.65s	4.52s	520.41s	32.06s	67.58s
BankMarketing	811.30s	0.55s	4.54s	421.21s	29.24s	56.29s
CreditCard	466.47s	0.47s	5.56s	292.83s	28.94s	47.52s

Table 10
Breakdown of SSDF training time into optimization and forest training stages.

Dataset	Optimization Time	Forest Training Time	Total Time
Adult	771.68s (96.5%)	1.79s (0.2%)	799.25s
BankMarketing	798.65s (98.4%)	2.15s (0.3%)	811.30s
CreditCard	459.00s (98.4%)	2.07s (0.4%)	466.47s

7. Conclusion

In this work, we presented a novel semi-supervised learning framework for DF that effectively addresses the challenge of limited labeled data. By jointly maximizing the average margin of labeled samples and minimizing the margin variance of unlabeled samples, our method enhances the generalization ability of DF from both theoretical and empirical perspectives. Experimental results across multiple benchmark datasets confirm the superiority of SSDF over existing semi-supervised approaches. Beyond its competitive performance, the proposed method retains the structural advantages of DF, making it a practical and reliable choice for real-world classification tasks. Future work may explore the integration of SSDF with domain adaptation techniques and its extension to regression or multi-label learning problems.

CRediT authorship contribution statement

Shen-Huan Lyu: Writing – original draft, Resources, Methodology, Funding acquisition, Conceptualization. **Jia-Le Xu:** Validation, Software. **Yi-Xiao He:** Writing – review & editing, Supervision, Software. **Yanyan Wang:** Validation, Investigation, Funding acquisition. **Baoliu Ye:** Writing – review & editing, Supervision. **Qingfu Zhang:** Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 62306104 and 62441225), Basic Research Program of Jiangsu (No. BK20253011), Hong Kong Scholars Program (No. XJ2024010), Research Grants Council of the Hong Kong Special Administrative Region, China (GRF Project No. CityU11212524), Natural Science Foundation of Jiangsu Province (No. BK20230949), Jiangsu Association for Science and Technology (No. JSTJ2024285), and China Postdoctoral Science Foundation (No. 2023TQ0104).

Data availability

Data will be made available on request.

References

[1] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems 30, 2017, pp. 2843–2851.
[3] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[4] Z.-H. Zhou, J. Feng, Deep forest: towards an alternative to deep neural networks, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 3553–3559.
[5] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[6] L.V. Utkin, An imprecise deep forest for classification, Expert Syst. Appl. 141 (2020) 112978.
[7] K. Bibas, O. Sar Shalom, D. Jannach, Semi-supervised adversarial learning for complementary item recommendation, in: Proceedings of the 23rd ACM Web Conference, 2023, pp. 1804–1812.
[8] Y. Ma, J. Wang, J. Yang, L. Wang, Model-heterogeneous semi-supervised federated learning for medical image segmentation, IEEE Trans. Med. Imaging 43 (5) (2024) 1804–1815.
[9] J.E. Van Engelen, H.H. Hoos, A survey on semi-supervised learning, Mach. Learn. 109 (2) (2020) 373–440.
[10] S. Goldman, Y. Zhou, Enhancing supervised learning with unlabeled data, in: Proceedings of the 16th International Conference on Machine Learning, 2000, pp. 327–334.
[11] Z.-H. Zhou, M. Li, Tri-training: exploiting unlabeled data using three classifiers, IEEE Trans. Knowl. Data Eng. 17 (11) (2005) 1529–1541.
[12] F. Breve, L. Zhao, M. Quiles, W. Pedrycz, J. Liu, Particle competition and cooperation in networks for semi-supervised learning, IEEE Trans. Knowl. Data Eng. 24 (9) (2011) 1686–1698.
[13] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Mach. Learn. 63 (1) (2006) 3–42.
[14] O. Chapelle, B. Schölkopf, A. Zien (editors), Semi-Supervised Learning, MIT Press, 2006.
[15] B. Zhao, X. Wen, K. Han, Learning semi-supervised Gaussian mixture models for generalized category discovery, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 16623–16633.
[16] Y. Li, Q. Pan, S. Wang, H. Peng, T. Yang, E. Cambria, Disentangled variational auto-encoder for semi-supervised learning, Inf. Sci. 482 (2019) 73–85.
[17] Q. Xie, M.-T. Luong, E. Hovy, Q.V. Le, Self-training with noisy student improves imagenet classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10687–10698.
[18] X. Peng, A ν-twin support vector machine (ν-TSVM) classifier and its geometric algorithms, Inf. Sci. 180 (20) (2010) 3863–3875.
[19] S. Melacci, M. Belkin, Laplacian support vector machines trained in the primal, J. Mach. Learn. Res. 12 (3) (2011) 262–282.
[20] A. Iscen, G. Tolias, Y. Avrithis, O. Chum, Label propagation for deep semi-supervised learning, in: Proceedings of the 25th IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5070–5079.
[21] Q. Xie, Z. Dai, E. Hovy, T. Luong, Q. Le, Unsupervised data augmentation for consistency training, Adv. Neural Inf. Process. Syst. 33 (2020) 6256–6268.
[22] L.V. Utkin, M.A. Ryabinin, A siamese deep forest, Knowl.-based Syst. 139 (2018) 13–22.
[23] Y.-L. Zhang, J. Zhou, W. Zheng, J. Feng, L. Li, Z. Liu, M. Li, Z. Zhang, C. Chen, X. Li, et al., Distributed deep forest and its application to automatic detection of cash-out fraud, ACM Trans. Intell. Syst. Technol. 10 (5) (2019) 1–19.
[24] L. Wu, J. Gao, Y. Zhang, B. Sui, Y. Wen, Q. Wu, K. Liu, S. He, X. Bo, A hybrid deep forest-based method for predicting synergistic drug combinations, Cell Rep. Methods 3 (2) (2023) 100411.
[25] B. Liu, Y. Sun, L. Gao, Enhancing groundwater recharge prediction: a feature selection-based deep forest model with Bayesian optimisation, Hydrol. Process. 38 (10) (2024) e15309.
[26] L.V. Utkin, A.V. Konstantinov, Attention-based random forest and contamination model, Neural Netw. 154 (2022) 346–359.
[27] S.-H. Lyu, L. Yang, Z.-H. Zhou, A refined margin distribution analysis for forest representation learning, in: Advances in Neural Information Processing Systems 32, 2019, pp. 5531–5541.
[28] L. Arnould, C. Boyer, E. Scornet, Analyzing the tree-layer structure of deep forests, in: Proceedings of the 37th International Conference on Machine Learning, 2021, pp. 342–350.
[29] S.-H. Lyu, Y.-X. He, Z.-H. Zhou, Depth is more powerful than width with prediction concatenation in deep forests, in: Advances in Neural Information Processing Systems 35, 2022, pp. 29719–29732.
[30] M. Pang, K.-M. Ting, P. Zhao, Z.-H. Zhou, Improving deep forest by screening, IEEE Trans. Knowl. Data Eng. 34 (9) (2022) 4298–4312.
[31] Y.-H. Chen, S.-H. Lyu, Y. Jiang, Improving deep forest by exploiting high-order interactions, in: Proceedings of the 21st IEEE International Conference on Data Mining, 2021, pp. 1030–1035.
[32] W. Cui, L. Zhang, B. Li, Z. Chen, M. Wu, X. Li, J. Kang, Semi-Supervised deep adversarial forest for Cross-Environment localization, IEEE Trans. Veh. Technol. 71 (9) (2022) 10215–10219.
[33] Y. Zhao, Y. Zhao, Z. Zhu, TSVM-HMM: transductive SVM based hidden markov model for automatic image annotation, Expert Syst. Appl. 36 (6) (2009) 9813–9818.
[34] C. Qian, Y. Yu, Z.-H. Zhou, Pareto ensemble pruning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 29, 2015, pp. 134–141.
[35] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning: from Theory to Algorithms, Cambridge University Press, 2014.
[36] M. Mohri, A. Rostamizadeh, A. Talwalkar, Foundations of Machine Learning, MIT Press, 2018.
[37] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C.A. Raffel, E.D. Cubuk, A. Kurakin, C.-L. Li, FixMatch: simplifying semi-supervised learning with consistency and confidence, in: Advances in Neural Information Processing Systems, 2020, pp. 596–608.
[38] S.Ö. Arik, T. Pfister, TabNet: attentive interpretable tabular learning, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 35, 2021, pp. 6679–6687.
[39] A. Sethi, S. Gupta, A. Agarwal, N. Agrawal, S. Asthana, Auto-TabTransformer: hierarchical transformers for self and semi supervised learning in tabular data, in: International Joint Conference on Neural Networks, IEEE, 2023, pp. 1–8.
[40] L.-H. Jia, L.-Z. Guo, Z. Zhou, Y.-F. Li, LAMDA-SSL: Semi-supervised learning in python, arXiv preprint arXiv:2208.04610, 2022.

[41] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving Problems with Box Constraints, IEEE Trans. Evol. Comput. 18 (4) (2013) 577–601.
[42] K. Deb, A. Pratap, S. Agarwal, T.A.M.T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[43] Q. Zhang, H. Li, MOEA/d: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[44] Z. Bingul, Adaptive genetic algorithms applied to dynamic multiobjective problems, Appl. Soft Comput. 7 (3) (2007) 791–799.