A Margin-Maximizing Fine-Grained Ensemble Method

Jinghui Yuan*
School of Artificial Intelligence, OPtics and ElectroNics (iOPEN)
Northwestern Polytechnical University
Shaanxi, China
yuanjh@mail.nwpu.edu.cn

Hao Chen*
Queen Mary School Hainan
Beijing University of Posts and Telecommunications
Beijing, China
HaoChenn.Eric@gmail.com

Renwei Luo
School of Artificial Intelligence, OPtics and ElectroNics (iOPEN)
Northwestern Polytechnical University
Shaanxi, China
rweiluo@gmail.com

Feiping Nie†
School of Artificial Intelligence, OPtics and ElectroNics (iOPEN)
Northwestern Polytechnical University
Shaanxi, China
feipingnie@gmail.com

***Abstract*—Ensemble learning has achieved remarkable success in machine learning, but its reliance on numerous base learners limits its application in resource-constrained environments. This paper introduces an innovative "Margin-Maximizing Fine-Grained Ensemble Method" that achieves performance surpassing large-scale ensembles by meticulously optimizing a small number of learners and enhancing generalization capability. We propose a novel learnable confidence matrix, quantifying each classifier's confidence for each category, precisely capturing category-specific advantages of individual learners. Furthermore, we design a margin-based loss function, constructing a smooth and partially convex objective using the logsumexp technique. This approach improves optimization, eases convergence, and enables adaptive confidence allocation. Finally, we prove that the loss function is Lipschitz continuous, based on which we develop an efficient gradient optimization algorithm that simultaneously maximizes margins and dynamically adjusts learner weights. Extensive experiments demonstrate that our method outperforms traditional random forests using only one-tenth of the base learners and other state-of-the-art ensemble methods.**

***Index Terms*—Ensemble learning, fine-grained optimization, margin maximization, confidence matrix.**

I. INTRODUCTION

Ensemble learning [1], a core method in machine learning, builds powerful predictive models by integrating predictions from multiple base learners. It has demonstrated exceptional performance across various domains [2], [3], from computer vision [4] to natural language processing [5], bioinformatics [6], [7], and financial forecasting [8]. However, as application scenarios become more complex and computational resources become constrained, traditional ensemble methods face significant challenges in resource-limited environments. Notably, the issue of model efficiency and interpretability remains unresolved, with recent methods failing to maintain or improve model performance while substantially reducing the number of base learners.

* **Equal contributions.**
† **Corresponding author.**

Recent years have seen various approaches to optimize ensemble learning. Classical methods like Bagging [9], [10] and Boosting [11], [12] enhance performance by increasing base learner diversity. Algorithms such as XGBoost [13], [14] introduced gradient boosting, further improving accuracy. Recent research, like improved random forests [15], attempts to reduce decision tree correlations.

While these methods have improved ensemble model performance, they still rely on numerous base learners for optimal results. Traditional ensemble methods often employ simple averaging or global performance-based weighting, overlooking differentiated performance across classification scenarios. Some studies have introduced margin [16]–[20] concepts to enhance generalization, but often lead to difficult-to-optimize problems. These limitations highlight deficiencies in current ensemble learning methods regarding fine-grained integration, efficient use of limited base learners, and generalization capabilities [21].

To address these challenges, we propose an innovative ensemble learning method that surpasses the performance of large-scale ensembles with fewer base learners through fine-grained learner optimization and margin-based generalization enhancement. Our main contributions are:

- Introduction of a learnable confidence matrix Θ, enabling fine-grained optimization of base learners across categories. This approach captures each learner's unique strengths, enhancing the ensemble's accuracy and robustness, particularly for imbalanced classes and difficult samples.
- Design of an innovative margin-based loss function. Using logsumexp techniques, we construct a smooth and partially convex loss function. This enables adaptive allocation of learner confidence under constraints, effectively enhancing model generalization and greatly improving model robustness to unseen data.
- Comprehensive experimental and theoretical analysis.

Our method outperforms traditional random forests and other advanced ensemble methods using only $\frac{1}{10}$ of the base learners across various datasets. We explore its relationship with dropout, reveal its connection to random forests as a special case, and discuss extensions to stacking, advancing heterogeneous ensemble learning research.

II. METHODOLOGY

Assuming we now have k classifiers, denoted as $G_1, ..., G_k$, respectively. We aim to integrate these k base classifiers at a fine-grained level by learning Θ, where Θ_{ij} represents the confidence of the i-th class in assigning the sample to the j-th classifier. Assuming the dataset is $\{x_1, ..., x_n\}$, Y_i is the label of x_i and Y_i is a c-dimensional one-hot encoded column vector. $G_n(x_i)$ is also one-hot encoded column vector like Y_i. Let $g_i = [G_1(x_i), ..., G_k(x_i)]^T$, with $g_i \in \mathbb{R}^{k\times c}$ and $(g_i)_{pq}$ representing the element on p-th row and q-th column. Given Θ, the prediction of the ensemble classifier can be obtained using the following formula.

$$\hat{y}_i = \underset{j=1...c}{\arg\max}\ \mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]_j \tag{1}$$

Here, I is the identity matrix, $\mathcal{S}[\cdot]$ is the softmax operator, which applies the softmax operation to a vector, $\mathbf{1}$ is a c-dimensional column vector filled with 1, and $\odot$ denotes the Hadamard product, representing element-wise multiplication of matrices.

To enhance the generalization ability of the ensemble model, we aim to learn Θ such that the margin is maximized. A reasonable definition of the margin is as follows.

$$Y_i^T(Y_i \odot \mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]) - max_2[\mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]] \tag{2}$$

This margin is reasonable because it seeks to maximize the difference between the probability at the correct position and the probability at the second highest position. Here, $\max_2[\cdot]$ refers to the second highest value in the input vector. However, the problem lies in the fact that the max_2 function is non-smooth and non-convex, which poses significant challenges for optimization. We use a smooth convex function, the logsumexp function, to replace $\max_2$. Under the following assumption

$$max[\mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]] = \|Y_i \odot \mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]\| \tag{3}$$

the max_2 function can be smoothly approximated like the following formula by using the logsumexp function [22]. We will justify this assumption later. Here, α is a constant, and as long as α is relatively large, it does not affect the result [23].

$$max_2[\mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]] = \frac{1}{\alpha}\log\left(\sum_{j=1}^{c} e^{\alpha(\mathcal{S}[(I\odot\Theta g_i)\mathbf{1}]-Y_i\odot\mathcal{S}[(I\odot\Theta g_i)\mathbf{1}])_j}\right) \tag{4}$$

Based on this, the margin can be smoothly represented by the following formula.

$$\mathcal{M} = \sum_{i=1}^{n} Y_i^T(Y_i \odot \mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]) - \sum_{i=1}^{n}\frac{1}{\alpha}\log\left(\sum_{j=1}^{c} e^{\alpha(\mathcal{S}[(I\odot\Theta g_i)\mathbf{1}]-Y_i\odot\mathcal{S}[(I\odot\Theta g_i)\mathbf{1}])_j}\right) \tag{5}$$

To ensure that Eq.(3) is reasonable, we need another component in the loss function to ensure high accuracy. That is, we need to make the ensemble learning as accurate as possible. We can use the cross-entropy loss function to guarantee this, i.e., the second part is following.

$$\mathcal{C} = -\sum_{i=1}^{n}(Y_i^T(Y_i \odot \log(\mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]))) \tag{6}$$

Therefore, the final loss function can be expressed as the weighted sum of the two parts, given by the following formula. Our goal is to minimize the loss function $\mathcal{L}$.

$$\begin{aligned}\mathcal{L} &= \mathcal{C} - \gamma\mathcal{M} \\ &= \sum_{i=1}^{n} -\left(Y_i^T\log(\mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]) + \gamma Y_i^T\mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]\right) \\ &+ \gamma\sum_{i=1}^{n}\frac{1}{\alpha}\log\left(\sum_{j=1}^{c} e^{\alpha(\mathcal{S}[(I\odot\Theta g_i)\mathbf{1}]-Y_i\odot\mathcal{S}[(I\odot\Theta g_i)\mathbf{1}])_j}\right)\end{aligned} \tag{7}$$

It is worth mentioning that even if assumption in Eq.(3) does not hold, it does not affect the validity of our loss function $\mathcal{L}$. Because even if it does not hold, the loss function introduced by $\mathcal{M}$ becomes the following.

$$\mathcal{M}_i = Y_i^T(Y_i \odot \mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]) - max(\mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]) \tag{8}$$

This means that the optimization direction still progresses towards increasing $Y_i^T(Y_i \odot \mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}])$ and decreasing $max(\mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}])$. Once $Y_i^T(Y_i \odot \mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}])$ exceeds $max(\mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}])$, assumption in Eq.(3) holds, and it then transforms into Eq.(5).

III. OPTIMIZATION ALGORITHM

In this section, we design the optimization algorithm. First, we prove that the loss function is convex with respect to $\mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]$. Second, we demonstrate that the loss function is Lipschitz continuous and provide the Lipschitz constant, which makes our model robust and easier to optimize. Finally, we present the optimization method we used.

Theorem 1 The loss function $\mathcal{L}$ is a convex function with respect to $\mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]$

Proof : First, it is evident that $Y_i \odot \mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]$ is a linear transformation of $\mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]$. Leveraging the properties of convex functions, we know that for a convex function $f(x)$, $f(Ax+b)$ is also convex [24]. Combining this with the known convexity of the logsumexp function and the evident convexity of other terms [25], we can conclude the convexity of the loss function $\mathcal{L}$ with respect to $\mathcal{S}[(I \odot \Theta g_i)\,\mathbf{1}]$.

A Lipschitz continuous [26] loss function can make a machine learning model more robust [27]. We will prove that our loss function is Lipschitz continuous, and the Lipschitz constant L is less than $\sqrt{ck} \cdot (1+\gamma+\frac{\gamma}{c}e^{\alpha})$, which facilitates easier optimization.

Theorem 2 The loss function $\mathcal{L}$ is Lipschitz continuous $\|\mathcal{L}(\Theta) - \mathcal{L}(\tilde{\Theta})\| \leq L\|\Theta - \tilde{\Theta}\|_F$. Moreover, the Lipschitz constant L is less than $\sqrt{ck} \cdot (1+\gamma+\frac{\gamma}{c}e^{\alpha})$.

Proof : Similar to [21], the derivative of the loss function with respect to the variable Θ is following. Without loss of generality, we analyse the case with one data belongs to the m-th class, which corresponds to the m-th row of Y being 1 and all other rows being 0. Let g represent the result matrix, specifically the g_i mentioned earlier.

First, we calculate the partial derivative of the softmax function $\mathcal{S}[(I \odot \Theta g)\mathbf{1}]$ with respect to Θ_{kl}, which is the basis for all subsequent calculations:

$$\frac{\partial \mathcal{S}[(I \odot \Theta g)\mathbf{1}]_j}{\partial \Theta_{kl}} = \frac{\left(\delta_{kj} g_{lj} - g_{lk}\mathcal{S}[(I \odot \Theta g)1]_k\right)}{(\mathcal{S}[(I \odot \Theta g)1]_j)^{-1}} \tag{9}$$

where δ_{kj} is the Kronecker delta function [28] and it equals to 1 only if $k = j$, else 0.

Therefore, there is no need for summation, and the loss function can be divided into three parts as followed:

$$\begin{aligned} &\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 \\ &= -Y^T \log(\mathcal{S}[(I \odot \Theta g)\,\mathbf{1}]) - \gamma Y^T \mathcal{S}[(I \odot \Theta g)\,\mathbf{1}] \\ &+\gamma \frac{1}{\alpha} \log\left(\sum_{j=1}^{c} e^{\alpha(\mathcal{S}[(I\odot\Theta g)\mathbf{1}] - Y \odot \mathcal{S}[(I\odot\Theta g)\mathbf{1}])_j}\right) \end{aligned} \tag{10}$$

According to the chain rule and the Eq.(9), the first term $\mathcal{L}_1 = -Y^T \log(\mathcal{S}[(I \odot \Theta g)\,\mathbf{1}])$ with respect to Θ_{kl} is easily known to be

$$\frac{\partial \mathcal{L}_1}{\partial \Theta_{kl}} == -\left(\delta_{km} g_{lm} - g_{lk}\mathcal{S}[(I \odot \Theta g)1]_k\right) \tag{11}$$

Similarly, it is straightforward to compute

$$\frac{\partial \mathcal{L}_2}{\partial \Theta_{kl}} = -\gamma \mathcal{S}[(I \odot \Theta g)1]_m \left(\delta_{km} g_{lm} - g_{lk}\mathcal{S}[(I \odot \Theta g)1]_k\right) \tag{12}$$

It is easy to verify that the derivative of the third term can be expressed in the following form, where $\tilde{\mathcal{S}}_j = \mathcal{S}[(I \odot \Theta g)\,\mathbf{1}]_j$ and $\sum_{j=1}^{c} \mathcal{I}(j \neq m)$ means compute the summary when $j \neq m$.

$$\frac{\partial \mathcal{L}_3}{\partial \Theta_{kl}} = \frac{\sum_{j=1}^{c} \mathcal{I}(j \neq m) e^{\alpha \tilde{\mathcal{S}}_j} \tilde{\mathcal{S}}_j \left(\delta_{kj} g_{lj} - g_{lk} \tilde{\mathcal{S}}_k\right)}{\gamma^{-1}(\sum_{j=1}^{c} e^{\alpha \tilde{\mathcal{S}}_j} - e^{\alpha \tilde{\mathcal{S}}_m} + 1)} \tag{13}$$

After obtaining the complete gradient expression of the loss function with respect to Θ_{kl}, our next objective is to prove that this gradient is bounded. This step is crucial for subsequently establishing Lipschitz continuity, which in turn lays the foundation for applying efficient optimization algorithms.

To prove the boundedness of the gradient, we need to carefully analyze each component of the gradient expression. First, we observe that the output of the softmax function, $\mathcal{S}[(I \odot \Theta g)\,\mathbf{1}]_j$, is always within the interval $[0, 1]$. Similarly, g_{pk} and δ_{km} are both either 0 or 1. Hence, $|\frac{\partial \mathcal{L}_1}{\partial \Theta_{kl}}| \leq 1$ and $|\frac{\partial \mathcal{L}_2}{\partial \Theta_{kl}}| \leq \gamma$.

As for the third term, since we have $\mathcal{S}[(I \odot \Theta g)\,\mathbf{1}]_j \leq 1$ and $|\delta_{kj} g_{lj} - g_{lk}\mathcal{S}[(I \odot \Theta g)1]_k| \leq 1$, and the fact that

$$|\sum_{j=1}^{c} e^{\alpha \mathcal{S}[(I\odot\Theta g)\mathbf{1}]_j} - e^{\alpha \mathcal{S}[(I\odot\Theta g)\mathbf{1}]_m} + 1| \geq c \tag{14}$$

holds since the exponent is positive, we obtain the boundary as followed $|\frac{\partial \mathcal{L}_3}{\partial \Theta_{kl}}| \leq \frac{\gamma}{c} e^{\alpha}$. Combining these results, we can deduce an overall upper bound for the gradient: $|\frac{\partial \mathcal{L}}{\partial \Theta_{kl}}| \leq 1 + \gamma + \frac{\gamma}{c} e^{\alpha}$. By the Mean Value Theorem, for any two points Θ and $\tilde{\Theta}$, there exists an intermediate point ξ such that $\mathcal{L}(\Theta) - \mathcal{L}(\tilde{\Theta}) = \nabla \mathcal{L}(\xi) \cdot (\Theta - \tilde{\Theta})$ and by the Cauchy-Schwarz Inequality we obtain:

$$|\mathcal{L}(\Theta) - \mathcal{L}(\tilde{\Theta})| \leq \|\nabla \mathcal{L}(\xi)\|_F \cdot \|\Theta - \tilde{\Theta}\|_F \tag{15}$$

Using the given bound on partial derivatives $|\frac{\partial \mathcal{L}}{\partial \Theta_{kl}}| \leq 1 + \gamma + \frac{\gamma}{c} e^{\alpha}$, we can calculate the square of the Frobenius norm of the gradient:

$$\|\nabla \mathcal{L}(\xi)\|_F^2 = \sum_{k,l} (\frac{\partial \mathcal{L}}{\partial \Theta_{kl}})^2 \leq ck(1 + \gamma + \frac{\gamma}{c} e^{\alpha})^2. \tag{16}$$

Finally we derive the Lipschitz constant $L = \sqrt{ck} \cdot (1 + \gamma + \frac{\gamma}{c} e^{\alpha})$ and have proved the loss function $\mathcal{L}$ is Lipschitz continuous with a Lipschitz constant $L \leq \sqrt{ck} \cdot (1 + \gamma + \frac{\gamma}{c} e^{\alpha})$. Lipschitz continuity makes our optimization more robust. We employ gradient descent [29] to optimize the loss function.

Algorithm 1 Gradient Descent

Require: Matrix G
Ensure: Θ^*
1: Initialize Θ_0 randomly.
2: **while** not converged **do**
3: Randomly select a batch of columns g_i from G.
4: Compute $\nabla_\Theta \mathcal{L}(\Theta_p)$ by Eq.(11)(12)(13).
5: Update Θ by $\Theta_{p+1} = \Theta_p - \beta \nabla_\Theta \mathcal{L}(\Theta_p)$.
6: **end while**

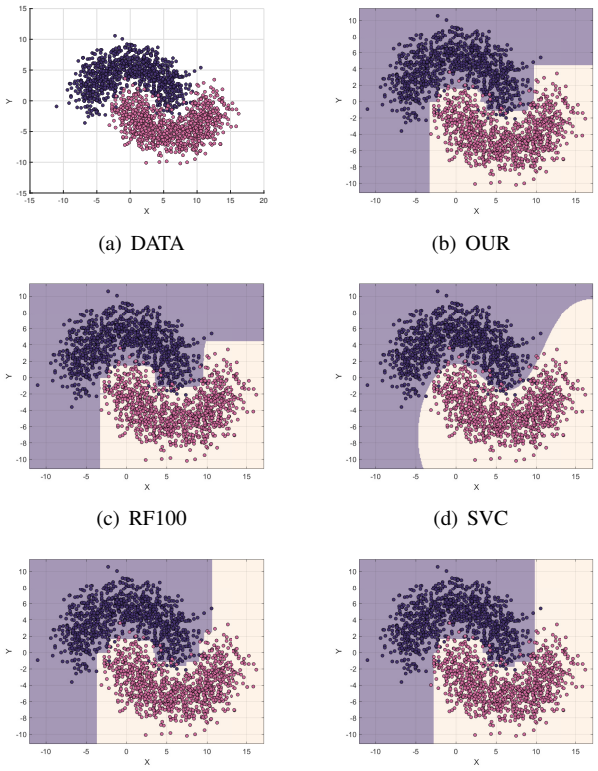
IV. Experiment

A. Experimental Methodology

To rigorously evaluate our proposed algorithm, we designed a comprehensive experimental framework utilizing a diverse dataset. This dataset comprises a moon-shaped toy dataset and 8 real-world datasets (BASEHOCK, breast uni, chess, iris, jaffe, pathbased, RELATHE, and wine) [30]–[34], encompassing a wide range of sample sizes and class numbers. We benchmarked our algorithm against Support Vector Classification (SVC), XGBoost, LightGBM and Random Forests with varying two numbers (50, 100), ensuring identical maximum depths for all tree-based models to maintain fairness. Our algorithm's hyperparameters were carefully tuned, with α

TABLE I: The accuracy metrics of our algorithm on various datasets

Data	Categories	OUR	RF50	RF100	SVM	XGBoost	LightGBM	# Object	# Attribute	# Class
TOY	Train	98.56	98.44	98.38	98.00	**98.69**	98.56	1600	2	2
	Test	**98.75**	98.00	98.50	98.50	98.00	97.75	400		
BASEHOCK	Train	95.80	92.16	92.35	**98.81**	95.92	95.17	1594	4862	2
	Test	93.73	88.72	88.72	**95.24**	90.00	93.23	399		
breast uni	Train	99.82	**100**	**100**	64.94	99.28	97.5	559	10	2
	Test	95.71	95.00	95.00	67.86	95.71	**96.43**	140		
chess	Train	**99.77**	99.57	99.61	94.17	99.50	98.90	2557	36	2
	Test	**99.06**	98.44	98.44	92.5	98.44	97.19	639		
iris	Train	**100**	100	100	95.50	98.33	98.33	120	4	3
	Test	**96.67**	93.33	93.33	96.67	96.67	96.67	30		
jaffe	Train	**100**	100	100	100	100	100	170	1024	10
	Test	97.67	95.35	97.67	**100**	93.02	93.02	43		
pathbased	Train	**100**	100	100	99.17	100	99.58	240	2	3
	Test	**100**	100	100	96.67	95.00	98.33	60		
RELATHE	Train	91.67	88.61	88.52	**95**	94.39	89.66	1142	4322	2
	Test	86.71	82.17	81.47	**90.56**	85.31	85.66	285		
wine	Train	**100**	100	100	71.83	100	100	142	13	3
	Test	**97.22**	97.22	97.22	69.44	91.67	91.67	36		



(a) DATA (b) OUR (c) RF100 (d) SVC (e) XGBOOST (f) LIGHTGBM

Fig. 1: The decision boundaries of various algorithms on the toy dataset. (a) DATA. (b) OUR. (c) RF100. (d) SVC. (e) XGBOOST. (f) LIGHTGBM.

fixed at 10 and γ randomly selected from $\{5, 10, 15, 20, 25\}$ to test stability. Employing an $80 : 20$ train-test split, we evaluated performance using training set accuracy, test set accuracy, and overall accuracy to gain comprehensive insights into the algorithm's capabilities. **Our random seed is fixed at 1, and we are about to open-source the code.**

B. Experimental Results and Analysis

The experimental results reveal significant advantages of our algorithm across multiple dimensions. **Notably, despite utilizing only 10 decision trees, our algorithm consistently outperformed Random Forests with 100 trees on several datasets**, demonstrating its exceptional efficiency in handling large-scale data classification tasks. Fig.1 shows the decision boundary of our algorithm on the toy dataset, demonstrating high accuracy in plane partitioning. This "less is more" performance underscores the effectiveness of our fine-grained approach in optimizing computational resources while maintaining high accuracy.

Our algorithm's generalization capability is particularly noteworthy. Although our algorithm does not achieve the best performance on every dataset, it surpasses the Random Forest algorithm using only $\frac{1}{10}$ of the trees, while maintaining optimal generalization ability.This pattern strongly indicates successful mitigation of overfitting, a crucial factor for model robustness in real-world applications.

V. Conclusion

In this paper, we introduce a novel fine-grained ensemble learning method achieving superior performance with fewer base learners. Our approach uses a learnable confidence matrix and a margin-based loss function, enabling precise optimization across categories and enhancing generalization. We provide theoretical guarantees on the loss function's local convexity and Lipschitz continuity. Experiments show our method outperforms random forests with 100 trees using only 10 base learners. The algorithm's rapid convergence and generalization highlight its potential for large-scale, resource-constrained applications. Future work will explore scalability to more complex datasets and adaptability to other machine learning paradigms.

REFERENCES

[1] C. Zhang and Y. Ma, *Ensemble machine learning*. Springer, 2012, vol. 144.

[2] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain adaptive ensemble learning," *IEEE Transactions on Image Processing*, vol. 30, pp. 8008–8018, 2021.

[3] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 2, pp. 757–774, 2023.

[4] R. Haeusler, R. Nair, and D. Kondermann, "Ensemble learning for confidence measures in stereo vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 305–312.

[5] S. Sangamnerkar, R. Srinivasan, M. Christhuraj, and R. Sukumaran, "An ensemble technique to detect fabricated news article using machine learning and natural language processing techniques," in *2020 International Conference for Emerging Technology (INCET)*. IEEE, 2020, pp. 1–7.

[6] A. Verma and S. Mehta, "A comparative study of ensemble learning methods for classification in bioinformatics," in *2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*. IEEE, 2017, pp. 155–158.

[7] X. Wang, C. Li, F. Li, V. S. Sharma, J. Song, and G. I. Webb, "Simlin: a bioinformatics tool for prediction of s-sulphenylation in the human proteome based on multi-stage ensemble-learning models," *BMC bioinformatics*, vol. 20, pp. 1–12, 2019.

[8] L. T. Bui, T. T. H. Dinh *et al.*, "A novel evolutionary multi-objective ensemble learning approach for forecasting currency exchange rates," *Data & Knowledge Engineering*, vol. 114, pp. 40–66, 2018.

[9] G. Ngo, R. Beard, and R. Chandra, "Evolutionary bagging for ensemble learning," *Neurocomputing*, vol. 510, pp. 1–14, 2022.

[10] G. Tüysüzoğlu and D. Birant, "Enhanced bagging (ebagging): A novel approach for ensemble learning," *International Arab Journal of Information Technology*, vol. 17, no. 4, 2020.

[11] C. Ying, M. Qi-Guang, L. Jia-Chen, and G. Lin, "Advance and prospects of adaboost algorithm," *Acta Automatica Sinica*, vol. 39, no. 6, pp. 745–758, 2013.

[12] G. I. Webb and Z. Zheng, "Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 8, pp. 980–991, 2004.

[13] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[14] O. Sagi and L. Rokach, "Approximating xgboost with an interpretable decision tree," *Information sciences*, vol. 572, pp. 522–542, 2021.

[15] Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, and X. Liang, "An improved random forest based on the classification accuracy and correlation measurement of decision trees," *Expert Systems with Applications*, vol. 237, p. 121549, 2024.

[16] J. Liu, L.-W. Huang, Y.-H. Shao, W.-J. Chen, and C.-N. Li, "A nonlinear kernel svm classifier via l0/1 soft-margin loss with classification performance," *Journal of Computational and Applied Mathematics*, vol. 437, p. 115471, 2024.

[17] B. Li, B. Yang, C. Liu, F. Liu, R. Ji, and Q. Ye, "Beyond max-margin: Class margin equilibrium for few-shot object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7363–7372.

[18] W. Feng, W. Huang, and J. Ren, "Class imbalance ensemble learning based on the margin theory," *Applied Sciences*, vol. 8, no. 5, p. 815, 2018.

[19] Q. Hu, L. Li, X. Wu, G. Schaefer, and D. Yu, "Exploiting diversity for optimizing margin distribution in ensemble learning," *Knowledge-Based Systems*, vol. 67, pp. 90–104, 2014.

[20] L. Li, Q. Hu, X. Wu, and D. Yu, "Exploration of classification confidence in ensemble learning," *Pattern recognition*, vol. 47, no. 9, pp. 3120–3131, 2014.

[21] J. Yuan, W. Jiang, Z. Cao, F. Xie, R. Wang, F. Nie, and Y. Yuan, "Achieving more with less: A tensor-optimization-powered ensemble method," *arXiv preprint arXiv:2408.02936*, 2024. [Online]. Available: https://arxiv.org/abs/2408.02936

[22] P. Blanchard, D. J. Higham, and N. J. Higham, "Accurately computing the log-sum-exp and softmax functions," *IMA Journal of Numerical Analysis*, vol. 41, no. 4, pp. 2311–2330, 2021.

[23] G. C. Calafiore, S. Gaubert, and C. Possieri, "A universal approximation result for difference of log-sum-exp neural networks," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 12, pp. 5603–5612, 2020.

[24] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[25] X. Xi, J. Xu, and Y. Lou, "Log-sum-exp optimization based on continuous piecewise linearization techniques," in *2020 IEEE 16th International Conference on Control & Automation (ICCA)*. IEEE, 2020, pp. 600–605.

[26] M.-M. Zühlke and D. Kudenko, "Adversarial robustness of neural networks from the perspective of lipschitz calculus: A survey," *ACM Computing Surveys*, 2024.

[27] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel, "Evaluating the robustness of neural networks: An extreme value theory approach," *arXiv preprint arXiv:1801.10578*, 2018.

[28] D. Kozen and M. Timme, "Indefinite summation and the kronecker delta," 2007.

[29] S.-i. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4-5, pp. 185–196, 1993.

[30] J. Yuan, C. Zeng, F. Xie, Z. Cao, C. Mulin, R. Wang, F. Nie, and Y. Yuan, "Doubly stochastic adaptive neighbors clustering via the marcus mapping," *arXiv preprint arXiv:2408.02932*, 2024.

[31] G. Yang, J. He, X. Lan, T. Li, and W. Fang, "A fast dual-module hybrid high-dimensional feature selection algorithm," *Information Sciences*, vol. 681, p. 121185, 2024.

[32] W. Shan, D. Li, S. Liu, M. Song, S. Xiao, and H. Zhang, "A random feature mapping method based on the adaboost algorithm and results fusion for enhancing classification performance," *Expert Systems with Applications*, vol. 256, p. 124902, 2024.

[33] H. Chang and D.-Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 191–203, 2008.

[34] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Proceedings Third IEEE international conference on automatic face and gesture recognition*. IEEE, 1998, pp. 200–205.