Chinese Journal of Electronics
Vol.XX, No.X, Sep. 2022

A Region-based Analysis for the Feature Concatenation in Deep Forests*

LYU Shen-Huan, CHEN Yi-He and ZHOU Zhi-Hua*

(*National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210023, China.*)

Abstract — Deep forest [1] is a tree-based deep model made up of non-differentiable modules that are trained without backpropagation. Despite the fact that deep forests have achieved considerable success in a variety of tasks, feature concatenation, the ingredient for forest representation learning still lacks theoretical understanding. In this paper, we aim to understand the influence of feature concatenation on predictive performance. To enable such theoretical studies, we present the first mathematical formula of feature concatenation based on the two-stage structure, which regards the splits along new features and raw features as a region selector and a region classifier respectively. Furthermore, we prove a region-based generalization bound for feature concatenation, which reveals the trade-off between Rademacher complexities of the two-stage structure and the fraction of instances that are correctly classified in the selected region. As a consequence, we show that compared with the prediction-based feature concatenation (PFC), the advantage of interaction-based feature concatenation (IFC) is that it obtains more abundant regions through distributed representation and alleviates the overfitting risk in local regions. Experiments confirm the correctness of our theoretical results.

Key words — Deep forest, Overfitting, Generalization bound, Representation learning.

I. Introduction

Decision tree is a popular supervised machine learning model that has been shown to be effective in a variety of predictive tasks [2–4]. With the development of ensemble learning [5], lots of tree-based algorithms [6–10] have met considerable success in the machine learning community for their predictive performance. In addition, a series of unsupervised learning methods [11–14] based on the isolation degree of completely random trees have achieved significant progress.

Deep neural networks (DNNs) have recently demonstrated their superiority in machine learning and have been successfully applied to a variety of tasks, including computer vision [15,16] (CV), automatic speech recognition [17] (ASR), and natural language processing [18] (NLP). They are composed of parameterized differentiable non-linear modules trained by the backpropagation procedure. However, training DNNs relies on complex hyper-parameter tuning [19] and lots of training data [20]. For increasingly complex application requirements, the tree-based paradigm remains one of the most popular options. As a result, a line of research [21,22] shows that decision trees may be used to build a deep model with excellent accuracy without overfitting the training data.

Considering that traditional tree-based models cannot achieve in-model feature transformation, Zhou and Feng [23] propose the first deep forest model to investigate the possibility of tree-based representation learning. Later on, Feng and Zhou [24] show that random forests can do auto-encoder, implying that the informative rules of decision trees may accomplish representation learning. Deep forest is extended to numerous tasks and is successfully applied in metric learning [25], multi-label learning [26], semi-supervised learning [27], financial fraud detection [28,29], etc. Deep forests, on the other hand, require a significant amount of memory and time due to the storing of multi-layer forest modules to do layer-by-layer prediction on the test set. By proposing a screening strategy, Pang et al. [30] can improve overall efficiency. Chen et al. [31] design an interaction-based deep forest to improve testing efficiency and enrich new features.

Although deep forest has achieved great success in recent years, most of the improvements are heuristic. Lyu et al. [32], Arnould et al. [33] try to analyze the generalization performance from the perspective of variance reducing in some simplified cases. However, *feature concatenation*, a critical component of forest representation learning, still lacks a theoretical explanation. From the existing empirical results, the prediction-based feature concatenation (PFC) is too simple to obtain an abundant feature transformation. After that, the interaction-based feature concatenation (IFC) proposed by Ref. [31] extracts more complex interactions layer by layer to generate abundant new features. As a result, it is critical to investigate and

*Manuscript Received Jun 27, 2022; Accepted Aug 31, 2022. This work is supported by the National Natural Science Foundation of China (No.61921006).

⋆Corresponding author.

 DOI:10.1049/cje.20XX.0X.0XX

comprehend the impact of feature concatenation on deep forests, as this knowledge can help us create a more effective forest representation and train deep forests in open environments [34–36].

In this paper, we find that trees in the second layer of deep forest and beyond tend to primarily choose the new features to split. Based on this phenomenon, we decompose the forest module into two stages: a region selector and a region classifier. Under this two-stage structure, we establish a region-based generalization bound for deep forest, and compare the two main feature concatenation methods: prediction-based feature concatenation (PFC) and interaction-based feature concatenation (IFC). Our contributions are threefold as follows:

- We mathematically formulate the feature concatenation in deep forests and propose the first theoretical analysis for forest representation learning under the two-stage structure.
- Theoretical results show that IFC not only enriches the new features but also controls the overfitting risk in local regions to avoid the deterioration of global generalization performance.
- Experimental results verify that IFC is not as easy as PFC to overfitting with the increasing number of parameters of deep forest, where the number of parameters is related to the depth of cascade structure and the number of new features.

Organization. The rest of the paper is organized as follows. Section II introduces deep forest and existing improvements on its two components. In Section III, we focus on the numerical analysis of the concatenated features in deep forests. Section IV introduces the definition of the two-stage structure and compares PFC and IFC under this structure. Section V reports our region-based theoretical results based on the two-stage structure of feature concatenation. The theoretical results are confirmed by experiments in Section VI. Finally, Section VII concludes with future work.

II. Deep Forest

In Section II-1, we briefly introduce two key components of the cascade structure in deep forests (i.e., feature refinement and feature concatenation) and review the recent improvements. In Section II-2, we briefly introduce two commonly used feature concatenations: PFC and IFC.

1. Existing improvements

Zhou and Feng [23] first propose a deep forest model named gcForest. The cascade structure in each layer is illustrated in Fig. 1. It consists of two components: feature concatenation and feature refinement. Each layer learns new features through a forest module, which consists of an ensemble of decision trees. The forest module outputs the predictions as new features. Then, the new features and the raw features are concatenated together. Finally, they implement feature refinement by iterative replacement of new features in each layer.

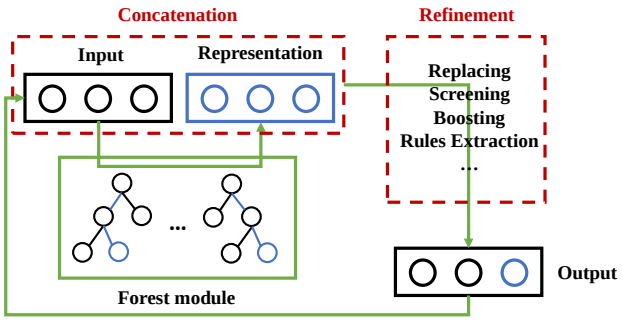


Fig. 1: Illustration of feature concatenation and feature refinement of cascade structure in each layer.

Since Zhou and Feng [1] propose the original deep forest that significantly improves the performance of tree-based models, several improvements are proposed by designing novel feature concatenations and feature refinements. Pang et al. [37] utilize confidence screening and feature screening to refine the concatenated data matrix. The screening method substantially reduces the number of instances that need to be processed and screens out many non-informative features. Therefore, it reduces time cost and memory requirement by one to two orders of magnitude. Lyu et al. [32] reformulate deep forest as an additive model boosting new features by optimizing the margin distribution layer by layer. They first give a theoretical explanation of the success of cascade structure from the perspective of margin theory. However, they only consider the influence of the feature refinement (boosting by margin distribution) and ignore the mechanism of feature concatenation. Chen et al. [31] utilize interactions in the decision rules to enrich concatenated features. The interactions are selected from the rules of different decision tree forests by evaluating their stability. We summarize improvements on feature concatenations and refinements in Table 1.

Lots of work has recently improved the two components of deep forests [29,31,32,37] and expanded the tree-based deep models to some specific settings, such as multi-label learning [26], multi-instance learning [38], multi-modal learning [39], semi-supervised learning [27] and crowdsourcing aggregation [40]. It would be interesting to explore the possibility of exploiting deep forests for rehearsal [41]. But deep forest still lacks theoretical analysis. Recently, Lyu et al. [32] give a theoretical explanation of the success of a

boosting-type feature refinement from the perspective of margin theory. Pang et al.[37] prove that the screening-type feature refinement can vary the model complexity from low to high as the number of layers increases in deep forests. However, the role of feature concatenation remains a mystery. Arnould et al. [33] try to explain it as a variance reducer under the assumption of a shallow tree network. However, their analysis only applies to PFC and relies on the centered randomized tree assumption which is data-independent and ignores the impact of specific forest structures.

Table 1: Summary of existing improvements on feature concatenations and feature refinements.

Existing work	Feature Concatenation	Feature Refinement
gcForest[1]	Predictions	Iterative replacement
gcForest$_{ss}$ [37]	Predictions	Confidence screening & Feature screening
mdDF [32]	Additive predictions	Boosting by margin distribution
hiDF [31]	Interactions	Rules extraction by stability

2. PFC and IFC

Since this paper mainly studies the feature concatenation mechanism in deep forest, we will briefly introduce two main new feature generation methods and formulate their definitions. Consider the supervised learning problem of learning a mapping from the feature space $\mathcal{X}$ to the label space $\mathcal{Y}$, where $\mathcal{Y} = \{1, 2, \ldots, C\}$. Let the training set $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be drawn independently and identically from the underlying distribution D. We denote by $f_k(\cdot)$ the k-th layer random forest module and $\{r_{i,k}(\cdot)\}_{i \in I}$ the set of decision rules in it. A K-layer cascade forest $F_K(x)$ can be recursively defined by

$$F_K(x) = \begin{cases} f_1(x) & K = 1; \\ f_K(\mathrm{Conc}(x, r_{K-1}(x))) & K > 1; \end{cases} \quad (1)$$

where $\mathrm{Conc}(x, r_{K-1}(x))$ denotes the concatenation of the raw features x and the new features $r_{K-1}(x)$. Next, we can define the mechanisms of PFC and IFC respectively.

In PFC, Zhou and Feng[1] directly take the probability vector of random forest as new features:

$$r_k(x) = f_k(\mathrm{Conc}(x, r_{k-1}(x))). \quad (2)$$

In IFC, Chen et al. [31] try to find the feature interaction information from the set of decision rules through an interaction selection algorithm $\mathcal{A}$ to generate new features

$$r_k(x) = \mathcal{A}(\{r_{i,k}(\mathrm{Conc}(x, r_{k-1}(x)))\}_{i \in I}), \quad (3)$$

where Algorithm 1 show the interaction selection.

Algorithm 1 Interaction selection algorithm

Input: Rule set $\{r_i(x)\}_{i \in I}$. Stability parameter τ. Size of sampled rule set N. Number of sampling rounds L.
Output: Interaction-based new features $r(x)$.

1: Disassemble the split of the decision rule into two parts: dimension $j \in \{1, 2, \ldots, d\}$ and threshold $z \in \mathbb{R}$: $r_i(x) = 1_{[x_{j_i} < z_i]}$.
2: **for** $\ell \in \{1, 2, \ldots, L\}$ **do**
3: Randomly sample a set of decision rules of size N.
4: Select a set $\{j_i\}_{i \in I_\ell}$ whose elements have more than τ repetitions in the sampled rule set, where ℓ denotes the randomness of the sampling algorithm.
5: Find the corresponding $\{z_i\}_{i \in I_\ell}$ according to the selected $\{j_i\}_{i \in I_\ell}$, thus we get a stable rule set $\{r_i(x)\}_{i \in I_\ell}$.
6: For any data point x, take the intersection of all the selected rules as the feature interaction of x: $r_\ell(x) = \prod_{i \in I_\ell} r_i(x)$.
7: **end for**
8: **return** $r(x) = (r_1(x), \ldots, r_\ell(x), \ldots, r_L(x))$.

III. Observations of feature concatenation

In this section, we compare the impact of feature concatenation of two types of deep forests (PFC[1] and IFC [31]) empirically. We find that the trees in the shallow layers tend to primarily choose the new features to split, which makes them more important for the predictions.

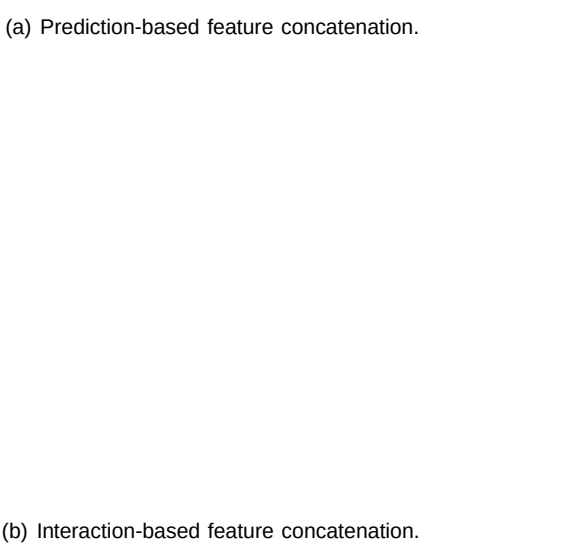
1. The dominance of new features in CART

Following the simplification of deep forests in Ref.[33], we first study a simplified version: a shallow tree-based network, composed of two layers, with one random forest in the first layer and one CART in the second layer. Fig. 2 is an example using the Adult data set, showing that the second-layer tree is observed to always make its first cut over the new features, using whether PFC or IFC.

Fig. 2: Illustration of the first two layers of splits on the Adult data set. Raw features are X [0]-X [13], the rests are the new features generated by the first-layer tree. Left : Splits using PFC. Right : Splits using IFC.

2. The dominance of new features in deep forest

Considering that there is various randomness in the forest model of each layer, we can not get all the exact tree structures. Therefore, we calculate the fraction of new features in the di erent levels of decision trees w.r.t. the layers of deep forests in the heatmap. Fig. 3 shows that the new features mainly appear in the shallow levels of the decision trees and will dominate the shallow splits (there is almost no raw feature in the shallow levels). Moreover, by comparing Fig. 3(a) and Fig. 3(b), we can nd that the dominance of the new features obtained by IFC increases layer by layer, which also implies that the new features of IFC can further evolve.



(a) Prediction-based feature concatenation.

(b) Interaction-based feature concatenation.

Fig. 3: The heatmap calculating the fraction of new features in the di erent depths of decision trees, with respect to the layers of deep forests on the Adult data set.

3. Understanding feature concatenations

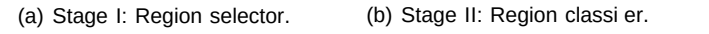
Fig. 1 shows that each forest module except the rst layer deals with both the raw features and the new features that are concatenated together. However, recent empirical works[31,33] show that the importance of these two types of features is totally di erent. Figs. 2 and 3 show that the new features often appear in the shallow levels of decision trees in the forest module, which plays a more important role in prediction.

IV. Two-stage structure

We regard the feature concatenation in each layer as a two-stage model:

- In stage I, the forest module learns a region selector $s(\cdot; j)$ according to the new features, where j represents the index of the selected region and $s(x; j) > 0$ represents that x is in this region.
- In stage II, the forest module learns a classi er $h_j(\cdot)$ on each region through raw features.

The nal learned function in each module is the union of these stages. As shown in Fig. 4, we assume that both concatenated features are independently used in di erent stages, which essentially simpli es the deep forest model to a speci c function structure. It is easier to analyze the impact of di erent new features on generalization performance theoretically. Next, we provide a region-based analysis of the di erence between PFC and IFC under the two-stage structure.



(a) Stage I: Region selector. (b) Stage II: Region classi er.

Fig. 4: Illustration of the two-stage structure on feature concatenation. (a) Stage I: Region selector selects some active regions $s(\cdot; j)$ w.r.t. the new features $\{r_i(\cdot)\}$. (b) Stage II: Region classi er learns a base classi er h_j for each active regions $s(\cdot; j)$. The nal classi er is the ensemble of all the base classi ers trained on active regions.

Prediction-based concatenation. In the original version of deep forest[1], new features are the prediction vectors of forests. We consider the binary classi cation case, then the generated two-dimensional new features $(r_0(x); r_1(x))$ represent the probability that each sample is classi ed as $y \in \{0; 1\}$ by the forest. Due to the strong correlation between probability vector and labels, the data will be highly separable on new features. Therefore, in this case, the regions activated by di erent new features

do not coincide with each other, e.g., $s(x;1) = 1_{r_0(x)\ 1=2}$ and $s(x;2) = 1_{r_1(x)\ 1=2}$, and the boundary of selected region is very complex (equivalent to the decision function learned by the forest model, see Fig. 5(a)). Each instance x can only fall in a certain region, which only activates the classifier of this region in the stage II. This makes the new features difficult to transform layer by layer.

(a) Prediction-based regions. (b) Interaction-based regions.

Fig. 5: Illustration of the region selectors of two different feature concatenations. (a) The boundary of the prediction-based region selector is complex, the raw feature space is highly separable and the new features are almost invariable. (b) The boundary of interaction-based region selector is simple, the separability of raw feature space is weak and the new features can evolve layer by layer. The yellow region shows that the order of interactions can be improved layer by layer.

Interaction-based concatenation. The improved deep forest[31] utilizes stable interactions extracted from decision rules to enrich the new features. Specifically, this algorithm identifies prevalent feature interactions from numerous decision rules in random forests. As shown in Fig. 5(a), each interaction represents a specific region, e.g., $r_1(x) = 1_{0:2<x_{\ 1}<0:8\ \&\ 0:3<x_{\ 2}<1}(x); r_2(x) = 1_{0<x_{\ 1}<0:5\ \&\ 0<x_{\ 2}<0:5}(x)$, and the region selector learns lots of activated regions through the ensemble of interactions in stage I (see Fig. 5(b)), e.g., $s(x;3) = \prod_{i=1}^{2} r_i(x) = 1_{0:2<x_{\ 1}<0:5\ \&\ 0:3<x_{\ 2}<0:5}(x)$ is one of them. Therefore, each instance x can be activated by multiple different regions achieving distributed representation, and the final classifier will be an ensemble of the classifiers in these regions. Furthermore, these activated regions can be used to learn higher-order interactions (the yellow region in Fig. 5(b)) in the next layer of interaction extraction.

V. Region-based analysis

In this section, we provide an analysis of the feature concatenation. In particular, we are interested in the effect of dividing all instances into different sub-regions by splitting along the new features, and we can understand the different impacts of PFC and IFC.

Let $S = \{(x_i;y_i)\}_{i=1}^{m}$ be the training set of size m drawn according to the underlying distribution D, where $x_i \in \mathcal{X} = \mathbb{R}^d$ and $y_i \in \mathcal{Y} = \{0;1\}$ is the associated class label. Let the loss function be $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. For a given forest f, its generalization error $R(f)$ and empirical error $\hat{R}_S(f)$ are:

$$R(f) = \mathbb{E}_{(x;y)\sim D}\left[\ell(f(x);y)\right]; \tag{4}$$

and

$$\hat{R}_S(f) = \frac{1}{m}\sum_{i=1}^{m}\ell(f(x_i);y_i). \tag{5}$$

Definition 1 (Rademacher Complexity). Let $\mathcal{F}$ be a family of functions and a fixed sample of size m as $S = \{(x_1;y_1);\ldots;(x_m;y_m)\}$. Then, the empirical Rademacher complexity of $\mathcal{F}$ with respect to the sample S is defined as:

$$\hat{R}_S(\mathcal{F}) = \mathbb{E}_{\sigma}\left[\sup_{f\in\mathcal{F}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i f(x_i;y_i)\right]; \tag{6}$$

where $\sigma = (\sigma_1;\ldots;\sigma_m)^{\top}$, with σ_is independent uniform random variables taking values in $\{-1;+1\}$. Besides, the Rademacher complexity of $\mathcal{F}$ is the expectation of $\hat{R}_S(\mathcal{F})$ over all samples of size m drawn according to D:

$$R_m(\mathcal{F}) = \mathbb{E}_{S\sim D^m}\left[\hat{R}_S(\mathcal{F})\right]. \tag{7}$$

Suppose the deep forest has K layers, and we define the sequence of forest modules as $\mathcal{F} = \{f_1;\ldots;f_K\}$. For any k-th layer, the forest module can be decomposed into two stages:

- Region selector: Let $r_i \in \mathcal{R}_i$ denotes the i-th new feature learned in the previous layers, the region selector is defined as $s(x;j) = \prod_{i\in C_j} r_i(x)$, where C_j is the learned decision rules of new features in region j and the family of all decision rules in k-th layer is denoted by C_k.
- Region classifier: For each selected region j in $\mathcal{X}$, the classifier $h_j \in \mathcal{H}_j$ is learned by the samples satisfying $s(x;j) > 0$.

In other words, we denote by

$$\mathcal{H}_j = \left\{x \mapsto s(x;j)h_j(x) \middle| s(x;j) \in \prod_{i\in C_j}\mathcal{R}_i; h_j \in \mathcal{H}_j\right\}, \tag{8}$$

the family composed of products of a region j selector and a region j classifier. The training sample that are in the sub-region j and that are correctly classified m_j^+ is defined by $|\{i : y_i h_j(x_i) > 0; s(x_i;j) > 0\}|$. The forest function in k-th layer $f_k : \mathcal{X} \mapsto [-1;+1]$ is defined as follows:

$$\forall x \in \mathcal{X}; f_k(x) = \sum_{j\,:\,C_j\in C_k} s(x;j)h_j(x). \tag{9}$$

We denote by F_k the family of forest functions f_k in k-th layer thereby de ned.

Theorem 1. Assume that for all decision rules of new features $C_j \in C_k$, the two-stage functions in H_j take values in $[-1;1]$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample S of size m, the following holds for all $C_j \in C_k$ and all $f_k \in F_k$:

$$R(f_k) \le \hat{R}_S(f_k) + \sum_{j:C_j \in C_k} \min\left(4\hat{R}_S(H_j), \frac{m_j^+}{m}\right) + C(m,\delta) + \sqrt{\frac{\log\frac{4}{\delta}}{2m}}, \tag{10}$$

where

$$C(m,\delta) = 2\sqrt{\frac{\log K}{m}} + \sqrt{\frac{\log K}{m}\log\frac{m^2}{\log K}}. \tag{11}$$

Proof. First, we introduce the convex ensembles with multiple hypothesis set g. For any $\alpha \in \Delta^{|C_k|}$, denote g_α as follows,

$$g_\alpha(x) = \sum_{j:C_j \in C_k} \alpha_j s_k(x;j) h_j(x), \tag{12}$$

where $\Delta^{|C_k|}$ is the simplex in $R^{|C_k|}$. Fix $\rho > 0$, since g_α is a convex combination of the mappings $x \mapsto s(x;j)h_j(x)$, and note that $f_k(x) = \sum_{j:C_j \in C_k} s(x;j)h_j(x)$. According to Theorem 1 in Ref.[21], we can obtain

$$R(f_k) \le \inf_{\alpha \in \Delta^{|C_k|}} \left[\hat{R}_{S,\rho}(g_\alpha) + \frac{4}{\rho}\sum_{j:C_j \in C_k} \hat{R}_S(H_j)\right] + C(m,\delta) + \sqrt{\frac{\log(4/\delta)}{2m}}, \tag{13}$$

where

$$C(m,\delta) = 2\sqrt{\frac{\log K}{m}} + \sqrt{\frac{\log K}{m}\log\frac{m^2}{\log K}}; \tag{14}$$

and

$$\hat{R}_{S,\rho}(g_\alpha) = \frac{1}{m}\sum_{j:C_j \in C_k}\sum_{s(x;j)>0} \mathbb{1}[y_i \alpha_j h_j(x_i) < \rho], \tag{15}$$

where $\mathbb{1}[e(\cdot)]$ is the indicator function taking 1 if $e(\cdot)$ is true; and 0 otherwise. The second step is to provide the upper bound for the first term in r.h.s. of Eq. (13). Following the analysis in Refs[22,42],

$$\inf_{\alpha \in \Delta^{|C_k|}} \left[\hat{R}_{S,\rho}(g_\alpha) + \frac{4}{\rho}\sum_{j:C_j \in C_k} \hat{R}_S(H_j)\right] \le \hat{R}_S(f_k) + \sum_{j:C_j \in C_k} \min\left(4\hat{R}_S(H_j), \frac{m_j^+}{m}\right) + \min_{1 \le j \le K}\left[\sum_{j:C_j \in C_k} \frac{m_j^+}{m} - 4\hat{R}_S(H_j)\right], \tag{16}$$

where

$$K = \left\{j : m_j^+/m > 4\hat{R}_S(H_j)\right\}. \tag{17}$$

Hence, we combine Eq. (13) and Eq. (16),

$$R(f_k) \le \hat{R}_S(f_k) + \sum_{j:C_j \in C_k} \min\left(4\hat{R}_S(H_j), \frac{m_j^+}{m}\right) + \min_{1 \le j \le K}\left[\sum_{j:C_j \in C_k} \frac{m_j^+}{m} - 4\hat{R}_S(H_j)\right] + C(m,\delta) + \sqrt{\frac{\log(4/\delta)}{2m}}, \tag{18}$$

where

$$C(m,\delta) = 2\sqrt{\frac{\log K}{m}} + \sqrt{\frac{\log K}{m}\log\frac{m^2}{\log K}}. \tag{19}$$

To simplify the presentation, we ignore the non-leading terms and only keep the terms regarding the number of sub-regions $|C_k|$, instance number m and Rademacher complexity terms:

$$R(h) \le \hat{R}_S(h) + \sum_{j:C_j \in C_k} \min\left(4\hat{R}_S(H_j), \frac{m_j^+}{m}\right) + C(m,\delta) + \sqrt{\frac{\log(4/\delta)}{2m}}. \tag{20}$$

□

Remark 1. Theorem 1 provides a data-dependent generalization bound for the learning model of new features. It shows that the over tting risk of each region j activated by the decision rules C_j is bounded by the minimum of Rademacher complexity of two-stage function $\hat{R}_S(H_j)$ and the fraction of instances reaching each region that is correctly classi ed m_j^+/m. It is possible to choose a region classi er from H_j with a relatively large complexity, as the fraction of training sample points reaching that region is small compared to the complexity of H_j.

Theorem 2. Assume that for all $i \in C_j$, the representation functions in R_i take values in $[0;1]$, the classi er

functions in H_j take values in $[-1,+1]$. Then, the empirical Rademacher complexities of H_j for any sample S of size m are bounded as follows:

$$\hat{R}_S(H_j) \leq 2\left[\sum_{i\in C_j}\hat{R}_S(R_i) + \hat{R}_S(H_j)\right]. \quad (21)$$

Proof. First, we present an important lemma as follows:

Lemma 3. (Lemma 3 in Ref. [22]) Let H_1 and H_2 be two families of functions mapping X to $[0,1]$ and let F_1 and F_2 be two families of functions mapping X to $[-1,1]$. Let $H = \{h_1 h_2 : h_1 \in H_1, h_2 \in H_2\}$ and let $F = \{f_1 f_2 : f_1 \in F_1, f_2 \in F_2\}$. Then, the empirical Rademacher complexities of H and F for any sample S of size m are bounded as follows:

$$\hat{R}_S(H) \leq \frac{3}{2}\left[\hat{R}_S(H_1) + \hat{R}_S(H_2)\right]; \quad (22)$$

$$\hat{R}_S(F) \leq 2\left[\hat{R}_S(F_1) + \hat{R}_S(F_2)\right]. \quad (23)$$

The forest module in this paper consists of two stages as follows:

$$H_j = \left\{x \mapsto s(x,j)h_j(x) | s(x,j) \in \prod_{i\in C_j} R_i, h_j \in H_j\right\}, \quad (24)$$

where C_j is the learned decision rules of new features in region j. We can get the following results according to Lemma 3:

$$\hat{R}_S(H_j) \leq 2\left[\sum_{i\in C_j}\hat{R}_S(R_i) + \hat{R}_S(H_j)\right]. \quad (25)$$

Speci cally, when the same decision rule set is used for all new features, that is $R_i = R$ for all i for some R, then the bound admits the following simpler form:

$$\hat{R}_S(H_j) \leq 2|C_j|\hat{R}_S(R) + 2\hat{R}_S(H_j). \quad (26)$$

□

Remark 2. Theorem 2 further shows that the complexity of a two-stage structure consists of two components: the sum of complexities of the new feature set R_i and the complexity of the region classi er set H_j. If we x the set of new features $R_i = R$, then the sum of representation complexity is proportional to the number of new features $|C_j|$.

Overview of theoretical results. Recall that the goal of analyzing feature concatenation is to guide the design of new features, so we compare the two existing types of feature concatenations through our theoretical results.

ˆ Theorem 1 shows that IFC can alleviate over tting risk even by choosing a complex classi er locally, as the fraction of training sample points reaching that region is small. In contrast, PFC is easier to over t, because the region corresponding to the prediction-based feature is highly correlated with the label, and many data points fall into each region. This is one of the reasons explaining why it is infeasible to construct a deep model by simply exploiting stacking, which over ts seriously with more than two layers, as mentioned by Ref. [1].

ˆ Theorem 2 shows that using stability as an evaluation to lter the set of interactions can restrict the complexity of new features, and alleviate the over tting risk in IFC. In contrast, PFC generates complex but almost invariant new features leading to over tting when we increase the number of new features.

VI. Experiments

We discuss our experimental con gurations in Section VI-1, and show that IFC outperforms PFC as the increasing depth of cascade structure and the increasing number of new features in Section VI-2.

1. Con gurations

Data sets. We select four widely used benchmark data sets of binary classi cation tasks from the UCI Machine Learning Repository[43]. Table 2 presents the basic statistics of these data sets.

Table 2: Statistics of the data sets.

Data sets	# of examples	# of features	# of classes
Adult	48,842	14	2
Diabetes	100,000	109	2
Congestive Heart Failure	9809	130	2

Table 3: Details of the con gurations of di erent models.

Data sets	Hyperparam setting	Optimal number of new features	Optimal depth of cascade structure
hiDF			
Adult	2 RF & 2 ERF, 500 trees	24	6
Diabetes		22	5
Congestive Heart Failure		40	3
CasForest$_{w/oCV}$			
Adult	2 RF & 2 ERF, 100 trees, 5-fold cross-validation	4	4
Diabetes		16	6
Congestive Heart Failure		4	4
CasForest$_{w/oCV}$			
Adult	2 RF & 2 ERF, 500 trees	4	2
Diabetes		8	1
Congestive Heart Failure		4	2

(a) Adult. (b) Diabetes. (c) Congestive Heart Failure.

Fig. 6: Performance with increasing depth of cascade structure on three data sets.

(a) Adult. (b) Diabetes. (c) Congestive Heart Failure.

Fig. 7: Performance with increasing number of new features on three data sets.

Compared models. We evaluate the performance of the following forest models on multiple benchmark data sets. These include base classi ers in deep forest, Random Forest and Extremely Random Forest, and their performance will be used as the baselines of non-deep models.

- Random Forest (RF)[7] is an ensemble learning method for classi cation that operates by constructing a multitude of decision trees at training time.
- Extremely Random Forest (ERF)[9] is an ensemble learning method where randomness goes one step further in the way splits are computed. Thresholds are drawn at random for each candidate feature and the best is picked as the splitting rule.

To analyze the key factors a ecting the generalization performance in deep forest, we conduct a series of ablation experiments on the classical CasForest and compared the CasForest under di erent conditions

- CasForest [1] is a deep forest method for classi cation that builds cascade structure through PFC.
  - { $\text{CasForest}_{\text{w/oCV}}$ is a CasForest without k-fold cross-validation.
  - { $\text{CasForest}_{\text{wCV}}$ is a CasForest with k-fold cross-validation.
- hiDF[31] is a deep forest method for classi cation that builds cascade structure through IFC.

Parameter setting. In all experiments, all CasForest and hiDF use the same cascade structure. For fairness, we set the total number of trees used in each layer of di erent forests to be 2000, including the trees in k-fold cross-validation. $\text{CasForest}_{\text{wCV}}$ generates new features by k-fold cross-validation (k=5). See Table 3 for details. We do not select the number of layers of deep forests, but observe the generalization performance layer by layer, and obtain the optimal layer of the deep forest under di erent algorithms in Table 3.

2. PFC vs IFC

In uence of the depth of cascade structure. We plot the curves of test accuracy of various methods with the depth of cascade structure on four di erent data sets, including hiDF, CasForest, $\text{CasForest}_{\text{w/oCV}}$, Random Forest and Extremely Random Forest. Fig. 6 shows that IFC outperforms PFC. hiDF is not easy to over t with the increase of layers. If the CasForest algorithm is implemented only through PFC named $\text{CasForest}_{\text{w/oCV}}$, as shown by the green line, the performance of the algorithm is seriously degraded. Even with the improvement of generalization ability brought by cross-validation, CasForest is still not as robust as hiDF against the over tting risk.

In uence of the number of new features. We plot the curves of test accuracy of hiDF and CasForest with the number of new features on four di erent data sets. The number of new features in hiDF can be increased by lowering the threshold of stability screening. The new features

in `CasForest` can be increased by splitting a forest into multiple forests. In particular, we also add random feature selection to each forest to obtain different new features. Fig. 7 shows that interaction-based representations outperform prediction-based representations and using stability to screen the representations is necessary to prevent the risk of overfitting.

Experimental results confirm Theorem 1 and Theorem 2: the distributed representation generated by IFC in stage I is better than the highly complex and invariant probability representations generated by PFC. The former can use more complex function families in some local regions with a few sample points to improve performance.

VI. Conclusion

In this paper, we offer a region-based analysis for feature concatenation in deep forests. Specifically, we prove a generalization bound, which reveals a trade-off between the Rademacher complexity of the classifier and the fraction of samples in the region selected by new features. This result indicates that interaction-based feature concatenation (IFC) can select regions with a small fraction of samples to alleviate the overfitting risk caused by high complexity, which can explain why IFC usually outperforms traditional prediction-based feature concatenation (PFC). In addition, it is still a long way to fully understand relevant mechanisms in deep forests such as the diversity of new features and the complexity of forest modules, and we leave those to future work.

References

[1] Z.-H. Zhou and J. Feng, "Deep forest," *National Science Review*, vol. 6, no. 1, pp. 74–86, 2019.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Boca Raton, FL: Chapman and Hall/CRC, 1984.

[3] Z.-H. Zhou, *Machine Learning*. Springer, 2021.

[4] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[5] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL: Chapman and Hall/CRC, 2012.

[6] R. E. Schapire and Y. Freund, *Boosting: Foundations and Algorithms*. MIT Press, 2012.

[7] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[8] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[9] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[10] 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.

[11] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proceedings of the 8th IEEE International Conference on Data Mining*, 2008, pp. 413–422.

[12] K. M. Ting, Y. Zhu, and Z.-H. Zhou, "Isolation kernel and its effect on svm," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 2329–2337.

[13] B.-C. Xu, K. M. Ting, and Z.-H. Zhou, "Isolation set-kernel and its application to multi-instance learning," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 941–949.

[14] K. M. Ting, B.-C. Xu, T. Washio, and Z.-H. Zhou, "Isolation distributional kernel: A new tool for kernel based anomaly detection," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020, pp. 198–206.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[16] S.-H. Lyu, L. Wang, and Z.-H. Zhou, "Improving generalization of deep neural networks by leveraging margin distribution," *Neural Networks*, vol. 151, pp. 48–60, 2022.

[17] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, and Y. Zhang, "Quartznet: Deep automatic speech recognition with 1D time-channel separable convolutions," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 6124–6128.

[18] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.

[19] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing Systems 24*, 2011, pp. 2546–2554.

[20] Z.-H. Zhou, "Why over-parameterization of deep neural networks does not overfit?" *Science China Information Sciences*, vol. 64, no. 1, p. 116101, 2021.

[21] C. Cortes, M. Mohri, and U. Syed, "Deep boosting," in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, 2014, pp. 1179–1187.

[22] G. DeSalvo, M. Mohri, and U. Syed, "Learning with deep cascades," in *Algorithmic Learning Theory*, 2015, pp. 254–269.

[23] Z.-H. Zhou and J. Feng, "Deep forest: Towards an alter-