Supplementary Materials for Paper: Dual Granular Balanced Deep Forest for Drug Combination Prediction

August 15, 2024

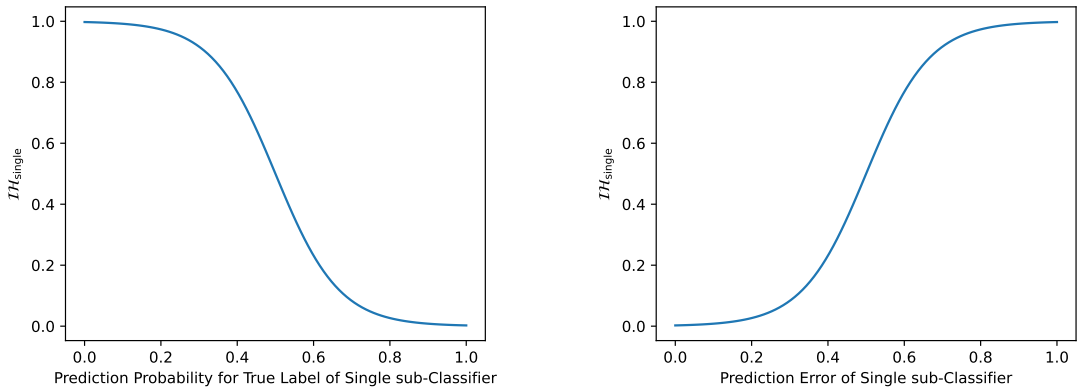


Figure S1. (a) Visualization of the IH function varying with the instance's prediction probability for the correct label. (b) Visualization of the IH function varying with the instance's prediction error.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{S1}$$

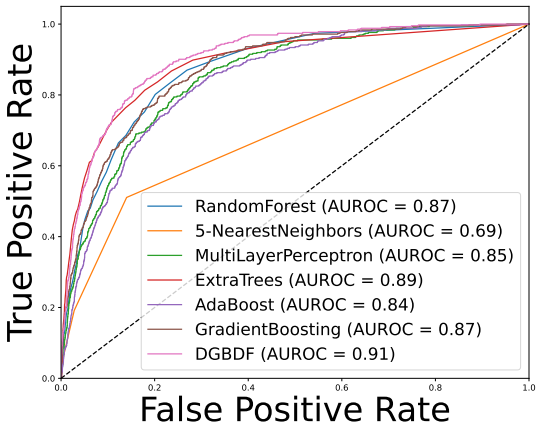
$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{S2}$$

$$\text{f1-macro} = \frac{1}{2}\sum_{i=1}^{2} 2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{S3}$$

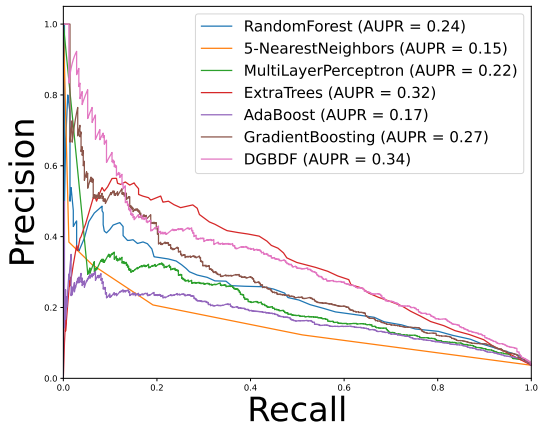
$$\text{Gmean} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \tag{S4}$$

$$\text{AUROC} = \int_0^1 \text{TPR}\, d(\text{FPR}) \tag{S5}$$

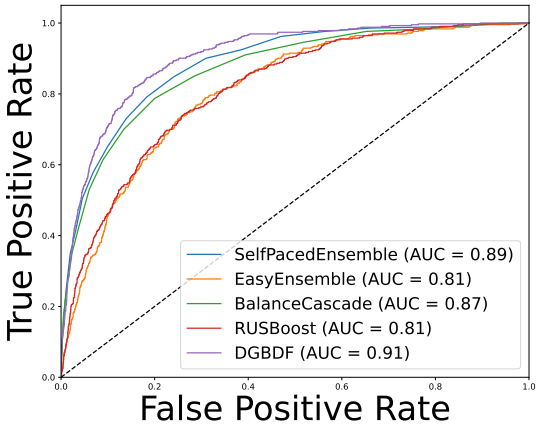
$$\text{AUPR} = \int_0^1 \text{Precision}\, d(\text{Recall}) \tag{S6}$$



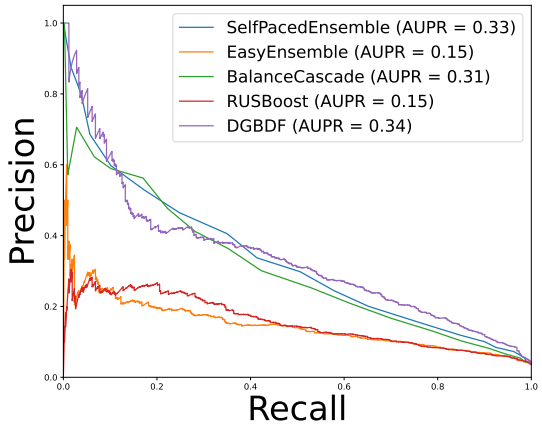
(a) AUROC-1



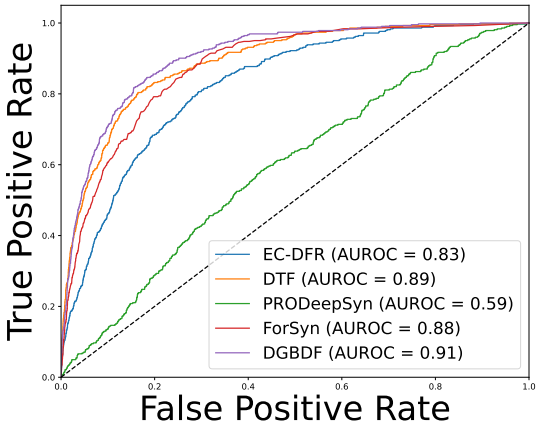
(b) AUPR-1



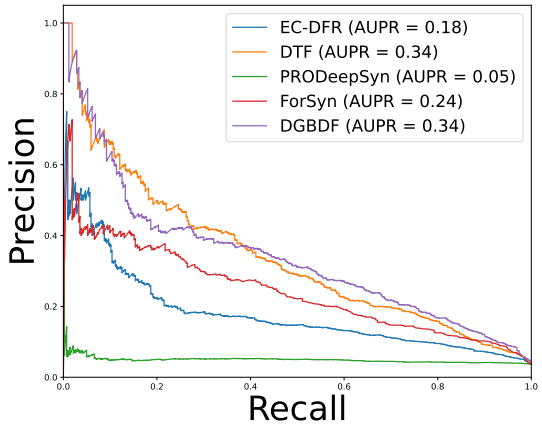
(c) AUROC-2



(d) AUPR-2



(e) AUROC-3



(f) AUPR-3

Figure S2. Comparison of AUROC and AUPR between DGBDF and various Canonical, Imbalanced Learning, and Existing Drug Combination Prediction Algorithms. DGBDF consistently outperforms other models in AUROC and AUPR.

TABLE SI
Summary of Compared Algorithms

Algorithm	**Description**
Self-Paced Ensemble (SPE)	Under-samples by coordinating data hardness through self-paced learning during the process of base classifier training, suitable for highly imbalanced and extremely skewed datasets.
Easy Ensemble (EE)	Under-samples the majority class ($\mathcal{N}$) in the dataset during the training process of each base classifier and integrates the results.
Balance Cascade (BC)	Guides the resampling process each time based on the existing trained base classifiers, based on EE.
RUSBoost (RUSB)	Combines random under-sampling and boosting techniques, randomly under-sampling the majority class instances during each iteration.
EC-DFR	An enhanced cascade deep forest model for drug combination prediction.
DTF	Extracts latent representations of drugs and cell lines from drug combinations, synergizing data through tensor factorization, then predicts the synergistic state of drug combinations through a neural network.
ProDeepSyn	Integrates PPI networks with omics data, constructing low-dimensional embeddings through GCN to predict anticancer synergistic drug combinations (SDCs).
ForSyn	An improved deep forest method that includes two embedded units: the RF-CSU unit for handling data imbalance and the ETF-DR unit for handling high-dimensional features, to predict synergistic drug combinations in different cells.

TABLE SII
Summary of Drug Combinations and Evidence

No	**Drug 1**	**Drug 2**	**Evidence (PMID)**	**Description**
1	PD173074	Erlotinib	**37173994**	Dual inhibition of FGFRs and EGFR by PD173074 and EGFR inhibitor erlotinib was synergistic in CCA
2	Entinostat	Paclitaxel	**27177222**	Entinostat significantly enhanced paclitaxel-mediated anti-proliferative/anti-survival effects on NSCLC cells in vitro and in vivo
3	Entinostat	Irinotecan	**34258881**	Combinations of irinotecan plus entinostat synergistically kill p53-positive CRC cells
4	Entinostat	Sorafenib	**25371323**	A combination of entinostat and sorafenib resulted in an additive to supra-additive growth inhibition of human bile duct adenocarcinoma cell lines
5	Bortezomib	Vinorelbine	**25833390**	Bortezomib increased the efficacy of ifosfamide and vinorelbine in pediatric Hodgkin lymphoma

Algorithm S1 DGBDF

Input:

$\mathcal{D}$: training set

Output:

$\mathcal{F}$: DGBDF

Function:

$\mathcal{F} \leftarrow \emptyset, \mathcal{IH} \leftarrow \emptyset, l \leftarrow 1$
while $l \leq d$ **do** // d is the max depth of the cascade
  if l = 1 **then**
    $\mathcal{L}^l, \mathcal{A}^l, \mathcal{IH}^l \leftarrow$ Algorithm2($\mathcal{D}$,$\mathcal{D}$,4)
  else
    $\mathcal{L}^l, \mathcal{A}^l, \mathcal{IH}^l \leftarrow$ Algorithm2($\mathcal{D}'$,$\mathcal{D}'_s$,4)
  end if
  Put $\mathcal{L}^l$ into $\mathcal{F}$, $\mathcal{IH}^l$ into $\mathcal{IH}$
  Calculate performance M^l for $\mathcal{F}$ by Eq. (13) with equal layer weights
  if $l \neq 1$ **and** $M^l - M^{l-1} < \epsilon$ **then**
    $\mathcal{F} \leftarrow \mathcal{F} - \mathcal{L}^l$
    break
  end if
  $\mathcal{D}' \leftarrow concatenate(X, \mathcal{A}^l), Y$
  $\mathcal{D}'_s \leftarrow$ Algorithm3($\mathcal{D}', \mathcal{IH}, nl, 5$)
  $l \leftarrow l + 1$
end while
Weight per layer by Eq. (11)-(12)
return final cascade model $F(x) = \frac{1}{l} \sum_{j=1}^{l} w^j * \mathcal{L}^j(x)$

Algorithm S2 the generation of DGBDF's l-th layer

Input:

$\mathcal{D}$: training set
$\mathcal{D}_s$: sampled training set
n: number of ensemble classifiers per layer

Output:

the DF's l-th layer $\mathcal{L}^l$
the augmented features $\mathcal{A}^l$
the $\mathcal{IH}^l$ of per ensemble classifier

Function:

$\mathcal{L}^l \leftarrow \emptyset, \mathcal{A}^l \leftarrow \emptyset, \mathcal{IH}^l \leftarrow \emptyset$
for $i = 1$ to n **do**
  Split $\mathcal{D}$ by the 5-fold stratified cross validation to get $\mathcal{T}_j$, $\mathcal{V}_j$ where $1 \leq j \leq 5$
  $\mathcal{T}'_j \leftarrow \mathcal{T}_j \cap \mathcal{D}_s$
  $\mathcal{L}_i \leftarrow \emptyset, \mathcal{A}_i \leftarrow \emptyset, \mathcal{IH}_i \leftarrow \emptyset$
  for $j = 1$ to 5 **do**
    Train $\mathcal{E}_{i,j}$ by Algorithm4($\mathcal{T}'_j$, $\mathcal{T}_j$, 20) and get its prediction $\hat{y}_{i,j}$ for $\mathcal{V}_j$
    Calculate $\mathcal{IH}_{i,j}$ for instances in $\mathcal{V}_j$ by Eq. (3)
    Put $\mathcal{E}_{i,j}$ into $\mathcal{L}_i$, $\hat{y}_{i,j}$ into $\mathcal{A}_i$, $\mathcal{IH}_{i,j}$ into $\mathcal{IH}_i$
  end for
  Put $\mathcal{L}_i$ into $\mathcal{L}^l$, $\mathcal{A}_i$ into $\mathcal{A}^l$, $\mathcal{IH}_i$ into $\mathcal{IH}^l$
end for
return $\mathcal{L}^l$, $\mathcal{A}^l$, $\mathcal{IH}^l$

Algorithm S3 Balancing Module

Input:

$\mathcal{D}$: training set
$\mathcal{IH}$: IH values set for all instances
K: number of all classifiers
b: number of bins

Output:

$\mathcal{D}_s$: sampled training set

Function:

$\mathcal{P} \leftarrow$ positive in $\mathcal{D}$, $\mathcal{N} \leftarrow$ negative in $\mathcal{D}$, $\mathcal{N}_s \leftarrow \emptyset$, $\mathcal{D}_s \leftarrow \emptyset$
Calculate $\overline{\mathcal{IH}}$ of per instance in $\mathcal{N}$ by Eq. (4)
Calculate per bin's capacity c by Eq. (8)
Calculate the percentiles $p_0, p_1, \ldots, p_b$ of $\overline{\mathcal{IH}}$ **w.r.t.** c
Cut majority set $\mathcal{N}$ into b bins by Eq. (9) **w.r.t.** $p_0, p_1, \ldots, p_b$
for $i = 1$ to b **do**
    Calculate uncertainty U of instances in Bin-i by Eq. (7)
    Calculate sampling probability pf per instance in Bin-i by Eq. (10)
    Sample $|\mathcal{P}|/b$ instances B_i' from Bin-i **w.r.t.** sampling probability
    $\mathcal{N}_s \leftarrow \mathcal{N}_s \cup B_i'$
    Put B_i' into $\mathcal{N}_s$
end for
$\mathcal{D}_s \leftarrow \mathcal{N}_s \cup \mathcal{P}$
return $\mathcal{D}_s$

Algorithm S4 Balanced Ensemble Classifier

Input:

$\mathcal{T}$: training set
$\mathcal{T}_s$: initial sampled training set
$n\_estimators$: number of decision trees

Output:

$\mathcal{E}$: balanced ensemble classifier

Function:

$\mathcal{E} \leftarrow \emptyset$, $\mathcal{IH} \leftarrow \emptyset$
for $i = 1$ to $n\_estimators$ **do**
    Train decision tree t_i on $\mathcal{T}_s$
    Calculate $\mathcal{IH}_i$ for instances in $\mathcal{T}$ by Eq. (3)
    Put $\mathcal{IH}_i$ into $\mathcal{IH}$
    $\mathcal{T}_s \leftarrow Algorithm3(\mathcal{T}, \mathcal{IH}, i, 5)$
    Put t_i into $\mathcal{E}$
end for
return $\mathcal{E}$