

Intellectual Systems and Technologies Интеллектуальные системы и технологии

Research article

DOI: <https://doi.org/10.18721/JCSTCS.16201>

UDC 004.85



FLEXIBLE DEEP FOREST CLASSIFIER WITH MULTI-HEAD ATTENTION

*A.V. Konstantinov*¹ , *L.V. Utkin*¹  ,
*S.R. Kirpichenko*¹ 

¹ Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

 lev.utkin@gmail.com

Abstract. A new modification of the deep forest (DF), called the attention-based deep forest (ABDF), for solving classification problems is proposed in the paper. The main idea behind the modification is to use the attention mechanism to aggregate predictions of the random forests at each level of the DF to enhance the classification performance of the DF. The attention mechanism is implemented by assigning the attention weights with trainable parameters to class probability vectors. The trainable parameters are determined by solving an optimization problem minimizing the loss function of predictions at each level of the DF. In order to reduce the number of random forests, the multi-head attention is incorporated into the DF. Numerical experiments with real data illustrate the ABDF and compare it with the original DF.

Keywords: machine learning, classification, random forest, decision tree, deep learning, attention mechanism

Acknowledgement: This work is supported by the Russian Science Foundation under grant 21-11-00116.

Citation: Konstantinov A.V., Utkin L.V., Kirpichenko S.R. Flexible deep forest classifier with multi-head attention. *Computing, Telecommunications and Control*, 2023, Vol. 16, No. 2, Pp. 7–16. DOI: [10.18721/JCSTCS.16201](https://doi.org/10.18721/JCSTCS.16201)





Научная статья

DOI: <https://doi.org/10.18721/JCSTCS.16201>

УДК 004.85



ГИБКИЙ КЛАССИФИКАТОР НА ОСНОВЕ ГЛУБОКОГО ЛЕСА С ИСПОЛЬЗОВАНИЕМ МНОГОМЕРНОГО ВНИМАНИЯ

А.В. Константинов¹ , Л.В. Уткин¹  ,
С.Р. Кирпиченко¹ 

¹ Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

 lev.utkin@gmail.com

Аннотация. В статье предлагается новая модификация глубокого леса, называемая глубоким лесом на основе механизма внимания, для решения задач классификации при ограниченной выборке. Основная идея модификации заключается в использовании механизма внимания для агрегирования предсказаний случайных лесов в виде векторов вероятностей классов на каждом уровне или слое глубокого леса для повышения эффективности классификации все модели. Механизм внимания реализуется путем присвоения веса внимания конкатенированным векторам примеров и векторов вероятностей классов так, что модель внимания имеет обучаемые параметры. Обучаемые параметры определяются путем решения задачи оптимизации, минимизирующей функцию потерь ошибки предсказаний на каждом уровне глубокого леса в процессе обучения глубокого леса на каждом уровне. Чтобы уменьшить количество случайных лесов, в глубокий лес включено так называемое многомерное внимание. Численные эксперименты на реальных данных иллюстрируют предлагаемую модификацию с точки зрения точности классификации и сравнивают ее с оригинальным глубоким лесом.

Ключевые слова: машинное обучение, классификация, случайный лес, дерево решений, глубокое обучение, механизм внимания

Финансирование: Работа выполнена при поддержке гранта РНФ № 21-11-00116.

Для цитирования: Konstantinov A.V., Utkin L.V., Kirpichenko S.R. Flexible deep forest classifier with multi-head attention // Computing, Telecommunications and Control. 2023. Т. 16, № 2. С. 7–16. DOI: 10.18721/JCSTCS.16201

Introduction

A lot of ensemble-based machine learning methods have been proposed [1, 2] due to their efficiency. These methods use a combination of the so-called base models to obtain more accurate predictions. Three types of the ensemble-based methods can be pointed out: bagging [3], stacking [4], and boosting [5]. Each type of methods has cons and pros. One of the important bagging methods is the random forest (RF) [6], which combines predictions of many randomly built decision trees. RFs are popular because they are simply trained and provide outstanding results for many datasets.

RFs can be regarded as powerful machine learning models. However, they cannot compete with deep neural networks. In order to partially overcome this disadvantage Zhou and Feng [7] proposed the so-called Deep Forest (DF) or gcForest, which copies the structure of multi-layer neural networks and consists of several layers or forest cascades. Each layer of the DF consists of several RFs, which produced predictions combined to use them at the next layer. The DF does not require gradient-based algorithms for training. This peculiarity makes the DF simple. Moreover, they have less hyperparameters in comparison with neural networks. Due to efficiency of the DF, many modifications have been proposed [8–16]. The DFs were used in various applications [17–21].

In order to improve RFs, the attention-based RFs were proposed in [22], where the trainable attention weights are assigned to each tree and each example. The weights depend on how far an instance, which falls into a leaf of a decision tree, is from the instances, which fall into the same leaf. The attention weights in the RF are used to compute the weighted average of the decision tree predictions.

It is important to note that the attention mechanism is successfully applied to neural networks to enhance their prediction abilities. It is based on the human perception property to concentrate on an important part of information and to ignore other information [23]. Therefore, the attention mechanism opened a door for implementing many neural network architectures, including transformers, the natural language processing models, etc., which are considered in detail in [23–26].

The attention-based RFs (ABRF) opened another door to the attention models different from the neural networks or their components. Therefore, we proposed a new attention-based model incorporated into the DF to enhance the DF prediction accuracy. The main idea behind the attention in the DF is to assign the attention weights to every RF at each layer to optimally combine the RF predictions and to produce new attended training feature vectors at each layer of the DF for training trees and RFs at the next layer. The attention-based DF is abbreviated as the ABDF.

The paper is organized as follows. A short description of the DF proposed by Zhou and Feng [7] and the attention mechanism are given in Section 2 and 3, respectively. Section 4 shows a general architecture of the attention-based DF. Numerical experiments with real data illustrate the attention-based DF and compare it with the original DF in Section 4. Concluding remarks are provided in Section 5.

A short introduction to the DF

Before considering the weighted DF, we briefly introduce gcForest proposed by Zhou and Feng [7]. The DF can be divided into two parts. The main part of gcForest is a cascade forest structure where each level of a cascade receives feature information processed by its preceding level, and outputs its processing result to the next level [7].

The main part of the DF proposed in [7] is a cascade forest structure shown in Fig. 1. One can see from Fig. 1 that each layer (level) of the cascade consists of several RFs whose number is a tuning parameter. Every RF produces a class probability distribution vector. The probability distributions of classes are determined in the standard way by counting the percentage of different classes of instances at the leaf node where the considered instance falls into. The RF class probability vectors are computed by averaging the class distribution vectors across all trees in the RF. The vectors produced by all RFs at each level are concatenated to each other. Moreover, the obtained concatenated class probability distribution vectors are concatenated with the input feature vector producing the training or testing vector for the next level. The feature vectors of the last level are combined into a single class probability vector by means of averaging. The final prediction corresponds to the largest probability from the class probability vector. The greedy algorithm is used to train the DF so that the next level of the forest cascade is trained on the feature vectors obtained from the previous level.

We suppose that there are Q levels (layers) of the DF, every level contains F forests, every RF consists of T decision trees. It is assumed for simplicity that F and T are identical at all levels.

Suppose that there are n training instances $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$, $\mathbf{x}_i = (x_{i1}, \dots, x_{im}) \in \mathbb{R}^m$, is a feature vector from m features, $y_i \in \{1, \dots, C\}$ is the target output. The class probability vector $\mathbf{p}_l = (p_{l,1}, \dots, p_{l,C})$ as the prediction of the l^{th} tree is defined as follows. Let the vector \mathbf{x} fall into a leaf of the l^{th} tree. Then there holds

$$p_{l,c} = \Pr\{c|\mathbf{x}\} = \frac{n_{l,c}}{\sum_{i=1}^C n_{l,i}} = \frac{n_{l,c}}{n_l},$$

where c is the class index $c \in \{1, \dots, C\}$, $n_{l,c}$ is the number of instances from the class c which fall into the same leaf as the vector \mathbf{x} in the l^{th} tree.

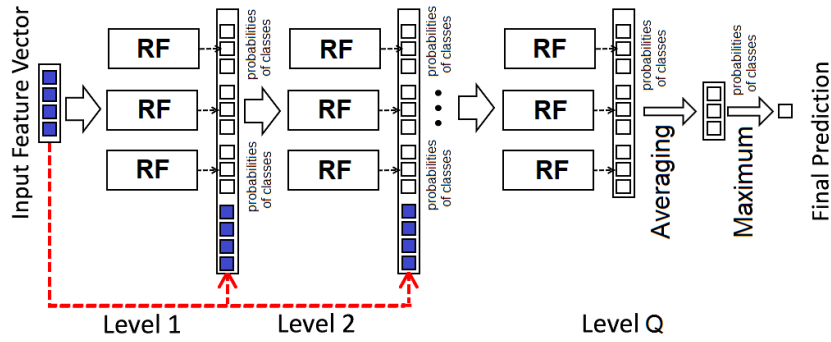


Fig. 1. Architecture of cascade forest

In other words, $p_{l,c}$ is the percentage of instances from class c , which fall into the leaf where the instance \mathbf{x} falls into. The following condition is fulfilled for all trees:

$$\sum_{c=1}^C p_{l,c} = 1.$$

The class probability vector $\mathbf{v}_j(i) = (v_{j,1}(i), \dots, v_{j,C}(i))$ as the prediction produced by the i^{th} RF for \mathbf{x}_j is defined as

$$v_{j,c}(i) = \frac{1}{T} \sum_{t=1}^T p_{j,c}^{(t)}, \quad c = 1, \dots, C.$$

According to [7], the concatenated vector $\mathbf{x}_j^{(q)}$ after the q^{th} level of the DF cascade is

$$\mathbf{x}_j^{(q)} = (\mathbf{x}_j, \mathbf{v}_j(1), \dots, \mathbf{v}_j(F)).$$

It consists of the original vector \mathbf{x}_j and F class probability vectors obtained from F RFs.

The attention mechanism and the attention-based RF

According to [24], the attention mechanism can be considered in terms of the Nadaraya–Watson kernel regression model [27, 28]. Given the training set S , the machine learning task is to find a function $f: \mathbb{R}^m \rightarrow \mathbb{R}$ predicting the target value \tilde{y} of a new instance \mathbf{x} based on the dataset S . Then the Nadaraya–Watson regression model can be written as follows:

$$\tilde{y} = \sum_{i=1}^n \alpha(\mathbf{x}, \mathbf{x}_i) y_i,$$

where $\alpha(\mathbf{x}, \mathbf{x}_i)$ are the attention weights depending on how the vector \mathbf{x}_i from the training set is close to the input vector \mathbf{x} , i.e. the closer \mathbf{x}_i to \mathbf{x} , the greater $\alpha(\mathbf{x}, \mathbf{x}_i)$.

The weights are expressed through the kernel K as:

$$\alpha(\mathbf{x}, \mathbf{x}_i) = \frac{K(\mathbf{x}, \mathbf{x}_i)}{\sum_{j=1}^n K(\mathbf{x}, \mathbf{x}_j)}.$$

Vector \mathbf{x} , vectors \mathbf{x}_i and outputs y_i are called *query*, *keys* and *values*, respectively, [29]. Generally, weight $\alpha(\mathbf{x}, \mathbf{x}_i)$ depends on the trainable parameters \mathbf{w} . If the Gaussian kernel is used to represent the attention weight, then we can write the following:

$$\alpha(\mathbf{x}, \mathbf{x}_i) = \text{softmax}(\mathbf{x}, \mathbf{x}_i, \mathbf{w}) = \frac{\exp\left(-\|\mathbf{w}(\mathbf{x} - \mathbf{x}_i)\|^2\right)}{\sum_{j=1}^n \exp\left(-\|\mathbf{w}(\mathbf{x} - \mathbf{x}_j)\|^2\right)}.$$

Here \mathbf{w} is the vector of trainable attention parameters, $\alpha(\mathbf{x}, \mathbf{x}_i, \mathbf{w})$ is an attention scoring function that maps two vectors to a scalar. It should be noted that there are various forms of incorporating trainable parameters. As a result, different expressions for the attention weights or for the scoring function have been studied and proposed. One of the popular scoring functions is defined as

$$s(\mathbf{x}, \mathbf{x}_i) = \mathbf{w}_v^T \tanh(\mathbf{W}_q \mathbf{x} + \mathbf{W}_k \mathbf{x}_i),$$

where \mathbf{w}_v or \mathbf{W}_v , \mathbf{W}_q , and \mathbf{W}_k are the vector and matrices of trainable parameters.

The corresponding attention is the well-known additive attention [29]. Another popular attention is the dot-product attention [30, 31]. The attention-based RF proposed in [22] is based on the Huber's ϵ -contamination model [32] with a specific trainable parameter, which is the contamination probability distribution.

Generally, the attention function (pooling) can be represented as an attention function f :

$$\mathbf{e} = f(\mathbf{W}_q \mathbf{x}, \mathbf{W}_k \mathbf{x}_i, \mathbf{W}_v y_i),$$

where \mathbf{e} is the output of the attention module (embedding).

Another approach for improving and extending the attention mechanism is to use the multi-head attention which is based on joint use of the different representation of queries, keys, and values in order to take into account multiple different aspects of data. The multi-head attention is implemented by means of different trainable parameters (heads) $\mathbf{w}_v^{(h)}$, $\mathbf{W}_v^{(h)}$, and $\mathbf{W}_k^{(h)}$. In this case, each attention head $\mathbf{e}^{(h)}$ is written as

$$\mathbf{e}^{(h)} = f(\mathbf{W}_q^{(h)} \mathbf{x}, \mathbf{W}_k^{(h)} \mathbf{x}_i, \mathbf{W}_v^{(h)} y_i).$$

When the attention is implemented by neural networks, the heads are determined by different initialization of the neural network parameters. After computing vectors $\mathbf{e}^{(h)}$, $h = 1, \dots, H$, the heads are concatenated.

The attention-based DF

Let us return to the DF. Suppose that we have the trained RFs consisting of T decision trees at the first level of the forest cascade and the instance \mathbf{x} is fed to the i^{th} RF. Let us compute the reconstruction of input feature vector $\hat{\mathbf{x}}(i)$ produced by the i^{th} RF as follows:

$$\hat{\mathbf{x}}(i) = \sum_{k=1}^T \alpha(\mathbf{x}, \hat{\mathbf{x}}^{(k)}(i)) \hat{\mathbf{x}}^{(k)}(i),$$

where the reconstruction produced by k^{th} tree is:

$$\hat{\mathbf{x}}^{(k)}(i) = \frac{1}{\#\mathfrak{S}_i^{(k)}(\mathbf{x})} \sum_{j \in \mathfrak{S}_i^{(k)}(\mathbf{x})} \mathbf{x}_j.$$

Here $\mathfrak{S}_i^{(k)}(\mathbf{x})$ is the set of instances from the training set S which fall into the same leaf from the k^{th} trees in the i^{th} RF as the vector \mathbf{x} falls; $\#\mathfrak{S}_i^{(k)}(\mathbf{x})$ is the number of elements in the set $\mathfrak{S}_i^{(k)}(\mathbf{x})$. It can be seen from

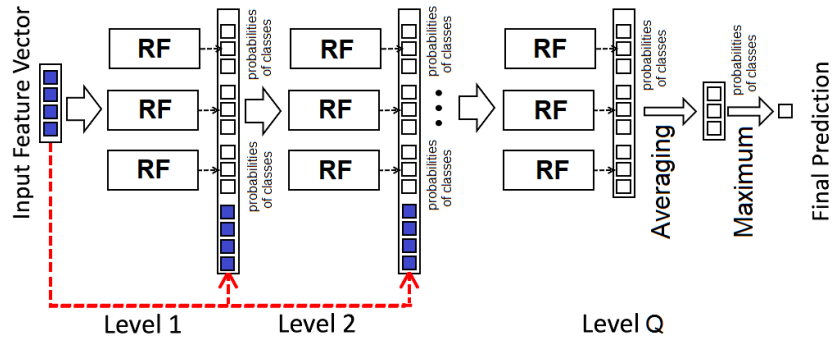


Fig. 2. The modified architecture of a level incorporating the multi-head attention

the above expression that $\hat{\mathbf{x}}(i)$ can be viewed as the weighted average over the vectors from S which are close to \mathbf{x} . It is important to point out that attention mechanism parameters for obtaining $\hat{\mathbf{x}}(i)$ can be optimized. However, these approaches complicate the training procedure, and we use the simplest averaging based on Gaussian kernel.

In order to indicate that the multi-head attention with H heads is used, we will denote the mean vectors $\hat{\mathbf{x}}(i)$ and the vector $\mathbf{v}(i)$ of the class probabilities as $\hat{\mathbf{x}}(i, h)$ and $\mathbf{v}(i, h)$, where i is the number of the RF, $i = 1, \dots, F$, h is the number of the head in the multi-head attention, $h = 1, \dots, H$. So, the prediction of the i^{th} RF at the first cascade, which is used in the h^{th} head of the attention, is the vector of probabilities $\mathbf{v}(i, h)$. We propose to concatenate the vectors $\hat{\mathbf{x}}(i, h)$ and $\mathbf{v}(i, h)$ in order to use the extended RF output $(\hat{\mathbf{x}}(i, h) \parallel \mathbf{v}(i, h))$. If there are F RFs at the level, then their outputs $(\hat{\mathbf{x}}(i, h) \parallel \mathbf{v}(i, h))$, $i = 1, \dots, F$, can be combined by applying the multi-head attention with H heads. In this case, we obtain H embedding vectors $\mathbf{e}^{(h)}(i)$, which can be concatenated for training the next level of the DF. The concatenated vector denoted as \mathbf{E} is transformed to a vector \mathbf{x}_{new} of the smaller size to use it at the next level of the DF cascade. This scheme is repeated for each level.

The proposed attention-based architecture of the DF level is shown in Fig. 2. One can see from Fig. 2 that the input vector \mathbf{x} is fed F to RFs (RF- i), which provide mean vectors $\hat{\mathbf{x}}(i, h)$ and the probabilities of classes $\mathbf{v}(i, h)$. Then concatenated vectors $\hat{\mathbf{x}}(i, h) \parallel \mathbf{v}(i, h)$ are attended with the vector \mathbf{x} (Attent h), and we obtain H vectors $\mathbf{e}^{(h)}(i)$, which are concatenated to each other into the vector \mathbf{E} . After that, the vector \mathbf{x}_{new} is calculated as $\mathbf{x}_{new} = \mathbf{W}_c \mathbf{E}$, where the matrix \mathbf{W}_c is trained jointly with the attention modules. Predictions of each head in the multi-head attention depend on subset of samples that correspond to the head: only samples from the subset are used to reconstruct the input vector and to estimate the class probabilities. The subsets for heads are generated using H -fold division of the training set S . The attention parameters are trained by using the same folds.

The proposed architecture has several advantages. First of all, it is flexible. We can change the number of RFs, the number of heads in the multi-head attention. We can change sizes of embeddings $\mathbf{e}^{(h)}(i)$, the size of the vector \mathbf{x}_{new} . All attention modules as well as the procedure of reducing the concatenated vector $(\mathbf{e}^{(h)}(1) \parallel \dots \parallel \mathbf{e}^{(h)}(H))$ to the vector \mathbf{x}_{new} have trainable parameters which allow us to obtain the best results. Secondly, we can reduce the number of RFs, which are hardly trained, by increasing the number of heads in the multi-head attention. This is a very important feature of the attention-based architecture. Thirdly, changing parameters of each level, we can obtain the heterogeneous structure of the DF, which leads to improved predictions of the whole model.

The simplest implementation of the attention-based DF is when the non-parametric attention mechanisms are used and the output feature vector \mathbf{x}_{new} is obtained by averaging the vectors $\mathbf{e}^{(h)}(1), \dots, \mathbf{e}^{(h)}(H)$. In this case, we train only the RFs. Other components are performed by computing their outputs under condition of certain inputs.

Numerical Experiments

In order to illustrate the attention-based DF, we investigate the model for datasets from UCI Machine Learning Repository [33]. Table 1 is a brief introduction about these datasets, while more detailed information can be found from the data resources. Table 1 contains the number of features m for the corresponding data set, the number of instances n and the number of classes C .

The ABDF implementation is based on the Bosk framework which is available at <https://github.com/NTAILab/bosk>.

Each level of the cascade structure consists of two RFs, each RF consists of 100 decision trees for almost all datasets except for the datasets WDBC, TTTE and Biodeg where numbers of trees in the corresponding RFs are 1000, 500, 500. The number of cascade levels is taken 3. The number of heads in the multi-head attention is 4.

Accuracy measure A used in numerical experiments is the proportion of correctly classified cases on a sample of data. To evaluate the average accuracy, we perform a cross-validation with 100 repetitions, where in each run, we randomly select $n_{tr} = 3n/4$ training data and $n_{test} = n/4$ testing data. Different values for the hyperparameters were tested, choosing those leading to the best results.

Numerical results of comparison of the original DF and the ABDF are shown in Table 2, where the first column contains abbreviations of the tested data sets, the second column contains the accuracy (the mean and standard deviation) of the ABDF, the third column contains accuracy values of the original DF. It can be seen from Table 2 that the proposed attention-based DF outperforms the original DF for most considered datasets.

Another interesting question is how the number of heads in the multi-head attention impacts the prediction accuracy. To study this question, datasets WDBC and TTTE are used, and the accuracy measures are obtained for numbers of heads 2, 4, and 6. The corresponding values of the accuracy for the dataset WDBC are 95.34, 96.64, and 97.20. Values of the accuracy for the dataset TTTE are 96.87, 97.08, and 97.36. It can be seen from the results that the number of heads increases the classification accuracy. On the other hand, the large number of heads in the multi-head attention significantly increase the computation time for training the ABDF. An optimal number of heads can be selected only in the testing phase.

Table 1

Brief introduction to datasets

Data set	Abbreviation	m	n	C
Haberman's Breast Cancer Survival	Haberman	3	306	2
Ionosphere	Ion	34	351	2
Seeds	Seeds	7	210	3
Teaching Assistant Evaluation	TAE	5	151	3
Tic-Tac-Toe Endgame	TTTE	9	958	2
QSAR Biodegradation	Biodeg	41	1055	2
Parkinsons	Parkinsons	22	195	2
Connectionist Bench	Sonar	60	208	2
SPECT Heart	SPECT	22	267	2
SPECTF Heart	SPECTF	44	267	2
Breast Cancer Wisconsin	WDBC	30	569	2

Table 2

Accuracy values (the mean and standard deviation) for comparison of the ABDF with the original DF

Dataset	ABDF	DF
Haberman	71.69±3.38	67.4±4.25
Ion	93.98±1.76	91.7±2.74
Parkinsons	92.65±2.08	91.84±3.65
Seed	95.28±3.50	93.21±2.56
SPECTF	80.15±4.63	81.04±4.43
SPECT	82.94±4.33	82.18±6.46
WDBC	96.41±2.14	95.31±1.90
Sonar	85.77±5.52	83.08±4.11
TAE	61.05±8.47	59.74±7.81
TTTE	97.92±1.05	97.63±0.93
Biodeg	86.63±1.38	87.27±1.65

Conclusion

The paper presented a new efficient modification of the DF. The main idea behind the proposed model is to incorporate the multi-head attention into each level of the DF. Numerical experiments showed that this idea leads to the model that outperforms the original DF.

The proposed model has several advantages. First, it allows us to reduce the number of RFs by increasing the number of heads in the multi-head attention mechanism at each level of the DF cascade. We can even use a single RF because the multi-head attention plays role of the base models like RFs. Secondly, it provides outperforming results due to usage of the attention-mechanism. Thirdly, it is flexible due to the data representation at the levels of the DF. Indeed, the output vector \mathbf{x}_{new} can have a structure different from the input vector produced by the previous level. As a results, RFs at the next level do not depend on RFs from the previous level, and we can expect better results due to some kind of the diversity of the base models. Fourthly, the ABDF opens the door for developing new modifications of the DF based on various forms of the attention mechanism. One of the direct modifications is to change the procedure for computing the average feature vector $\hat{\mathbf{x}}(i)$ producing by the i^{th} RF. We used the simplest procedure of weighted averaging of all vectors that fall into leaves jointly with the vector $\mathbf{x}(i)$. However, the self-attention can be applied to take into account the context of data as it is performed in Transformers. The self-attention can be incorporated into the multi-head attention. The above modifications as well as many other ones can be regarded as directions for further research.

REFERENCES

1. Rokach L. Ensemble-based classifiers. *Artificial Intelligence Review*, 2010, Vol. 33 (1-2), pp. 1–39.
2. Zhou Z.-F. Ensemble Methods: Foundations and Algorithms. CRC Press, Boca Raton, 2012.
3. Breiman L. Bagging predictors. *Machine Learning*, 1996, Vol. 24 (2), pp. 123–140.
4. El El-Dakhly D. Stacked generalization. *Neural networks*, 1992, Vol. 5 (2), pp. 241–259.
5. Freund F.I., Schapire R.E. A decision theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1997, Vol. 55 (1), pp. 119–139.
6. Breiman L. Random forests. *Machine learning*, 2001, Vol. 45 (1), pp. 5–32.

7. **Zhou Z.-FI., Feng J.** Deep forest: Towards an alternative to deep neural networks. Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJ-CAI17), strony 3553–3559, Melbourne, Australia, AAAI Press, 2017.
8. **Shen-Huan Lyu, Yi-Xiao He, Zhi-Hua Zhou.** Depth is more powerful than width with prediction concatenation in deep forest. *Advances in Neural Information Processing Systems*, 2022, no. 35, pp. 29719–29732.
9. **Miller K., Hettinger C., Humpherys J., Jarvis T., Kartchner D.** Forward thinking: Building deep random forests. arXiv:1705.07366, 20 May 2017.
10. **Pang M., Ting K.M., Zhao P., Zhou Z.-FI.** Improving deep forest by confidence screening. Proceedings of the 18th IEEE International Conference on Data Mining (ICDM18), strony 1–6, Singapore, 2018.
11. **Utkin L.V.** An imprecise deep forest for classification. *Expert Systems with Applications*, 2020, Vol. 141 (112978), pp. 1–11.
12. **Utkin L.V., Konstantinov A.V., Chukanov V.S., Meldo A.A.** A new adaptive weighted deep forest and its modifications. *International Journal of Information Technology & Decision Making*, 2020, Vol. 19 (4), pp. 963–986.
13. **Utkin L.V., Ryabinin M.A.** A Siamese deep forest. *Knowledge-Based Systems*, 2018, no. 139, pp. 13–22.
14. **Wen FI., Zhang J., Lin Q., Yang K., Jin T., Lv F., Pan X., Huang P., Zha Z.-J.** Multi-level deep cascade trees for conversion rate prediction. arXiv:1805.09484, May 2018.
15. **Heng Xia, Jian Tang, Junfei Qiao, Jian Zhang, Wen Yu.** DF classification algorithm for constructing a small sample size of data-oriented DF regression model. *Neural Computing and Applications*, 2022, Vol. 34 (4), pp. 2785–2810.
16. **Zhang X., Wang M.** Weighted random forest algorithm based on bayesian algorithm. *Journal of Physics: Conference Series*, wolumen 1924, strona 012006. IOP Publishing, 2021.
17. **Soheila Molaee, Amirhossein Havvaei, Hadi Zare, Mahdi Jalili.** Collaborative deep forest learning for recommender systems. *IEEE Access*, 2021, no. 9, pp. 22053–22061.
18. **Bishnupriya Panda, Shrabanee Swagatika, Sipra Sahoo, Debabrata Singh.** A novel approach for breast cancer data classification using deep forest network. *Intelligent and Cloud Computing: Proceedings of ICICC 2019*, Springer, 2021, no. 2, pp. 309–316.
19. **Liang Sun, Zhanhao Mo, Fuhua Yan, Liming Xia, Fei Shan, Zhongxiang Ding, Bin Song, Wanchun Gao, Wei Shao, Feng Shi, i in.** Adaptive feature selection guided deep forest for covid-19 classification with chest ct. *IEEE Journal of Biomedical and Health Informatics*, 2020, Vol. 24 (10), pp. 2798–2805.
20. **Ran Su, Xinyi Liu, Leyi Wei, Quan Zou.** Deep-resp-forest: a deep forest model to predict anti-cancer drug response. *Methods*, 2019, no. 166, pp. 91–102.
21. **Tianchi Zhou, Xiaobing Sun, Xin Xia, Bin Li, Xiang Chen.** Improving defect prediction with deep forest. *Information and Software Technology*, 2019, no. 114, pp. 204–216.
22. **Utkin L.V., Konstantinov A.V.** Attention-based random forest and contamination model. *Neural Networks*, 2022, no. 154, pp. 346–359.
23. **Niu Z., Zhong G., Yu FI.** A review on the attention mechanism of deep learning. *Neurocomputing*, 2021, no. 452, pp. 48–62.
24. **Chaudhari S., Mithal V., Polatkan G., Ramanath R.** An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology*, 2021, Vol. 12 (5), pp. 1–32. Article 53.
25. **Correia A.S., Colombini E.L.** Attention, please! A survey of neural attention models in deep learning. *Artificial Intelligence Review*, 2022, Vol. 55 (8), pp. 6037–6124.
26. **Lin T., Wang FI., Liu X., Qiu X.** A survey of transformers. arXiv:2106.04554, Jul 2021.
27. **Nadaraya E.A.** On estimating regression. *Theory of Probability & Its Applications*, 1964, Vol. 9(1), pp. 141–142.
28. **Watson G.S.** Smooth regression analysis. *Sankhya: The Indian Journal of Statistics, Series A*, 1964, pp. 359–372.
29. **Bahdanau D., Cho K., Bengio FI.** Neural machine translation by jointly learning to align and translate. arXiv: 1409.0473, Sep 2014.

30. **Luong T., Pham FI., Manning C.D.** Effective approaches to attention-based neural machine translation. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, The Association for Computational Linguistics, 2015, pp. 1412–1421.

31. **Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L., Polosukhin I.** Attention is all you need. Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

32. **Huber P.J.** Robust Statistics. Wiley, New York, 1981.

33. **Lichman M.** UCI machine learning repository, 2013. <https://archive.ics.uci.edu/ml/index.php>

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Andrei V. Konstantinov

Константинов Андрей Владимирович

E-mail: andrue.konst@gmail.com

<https://orcid.org/0000-0003-2275-1473>

Lev V. Utkin

Уткин Лев Владимирович

E-mail: lev.utkin@gmail.com

<https://orcid.org/0000-0002-5637-1420>

Stanislav R. Kirpichenko

Кирпиченко Станислав Романович

E-mail: kirpichenko.sr@gmail.com

<https://orcid.org/0000-0003-2275-1473>

Submitted: 28.05.2023; Approved: 25.06.2023; Accepted: 06.07.2023.

Поступила: 28.05.2023; Одобрена: 25.06.2023; Принята: 06.07.2023.