Article

Estimation of vegetation indices with Random Kernel Forests

Dmitry Devyatkin [1,†,‡]*

[1] Federal Research Center "Computer Science and Control" Russian Academy of Sciences; devyatkin@isa.ru
† Current address: Moscow, Russia

Abstract: Vegetation indexes help perform precision farming because they provide useful information regarding moisture, nutrient content, and crop health. Primary sources of those indexes are satellites and unmanned aerial vehicles equipped with expensive multispectral sensors. Reducing the price of obtaining such information would increase the availability of precision farming for small farms. Several studies have proposed deep neural network methods to estimate the indexes from RGB color images. However, those methods report relatively large errors for mature plants, when highly non-linear relationships of images and vegetation indexes arise. One could apply multilayer random forest-based models (Deep Forests) to solve this problem, but the discriminative power of such models is limited: they cannot catch complex dependencies between image features. In this paper, we propose a method that combines ideas of deep forests, random forests of kernel trees, and global pruning of random forests to tackle the problem. As a result, the method considers the properties of objects with a complex structure: the presence of relationships between groups of features, displacement, and scaling of objects. The experimental results show the proposed method outperforms neural network-based solutions on several datasets.

Keywords: vegetation indices; NDVI; RGB images; Deep Forest; Random Kernel Forests

---

1. Introduction

Vegetation indexes help estimate many crucial agricultural indicators such as moisture, nutrient content, and crop health. One of the most frequently used vegetation indexes is NDVI (normalized difference vegetation index). Primary sources of those indexes are satellites and unmanned aerial vehicles (UAV) equipped with expensive multispectral sensors. Reducing the price of obtaining such information would increase the availability of precision farming for small farms. Several studies have proposed deep neural network methods to estimate the indexes from RGB color images. However, as it is pointed out in [1] the framework provides accurate results only for immature plants, and the reason is highly non-linear relationship between RGB colors and vegetation for senescent plants. Consequently, the "high-frequency functions" phenomenon comes into effect [2].

In this paper, we propose to utilize a Deep Forest to deal with that issue because this approach does not use smooth models, but at the same time, it provides competitive results on image processing [3]. Deep forest uses a cascade structure to perform layer-by-layer processing of raw features. However, several studies report that Deep Forest has limited ability to catch dependencies between features, which can lead to poor performance in some cases [4]. Besides, it uses Extremely Random Forests to control overfitting, which in practice can lead to unstable results. We modified Deep Forest to overcome those issues as follows:

1. The use of random forests of decision trees with multivariate non-linear splits as the basic classification algorithm allows considering the relationships between the features of the analyzed objects, reducing the number of data processing layers and, consequently, improving performance [5,6].
2. The use of Extremely Random Forests is abandoned. Instead, we apply pruning from [7], which makes it possible to increase both the accuracy and the stability of the method.

This paper is structured as follows. Section 2 provides an overview of vegetation index detection studies, as well as various modifications of the Deep Forest model. Section 3 presents description of the proposed Deep Kernel Forest. Besides, it describes a modification of Kernel Forests to solve regression problems. Section 3 presents datasets used to perform experiments, and Section 4 provides conclusion and future work.

2. Related Work

Visual analysis of land is the primary tool of precision agriculture. For example, vegetation indices obtained by the analysis of multispectral images help monitor crop health. Paper [8] proposes a remote sensing recognition method based on a convolutional neural network. They combine 4 channels (red, green, blue, and near-infrared) to reveal the changing characteristics of the landslide. Finally, a convolutional neural network was applied to solve the problem. The experiments showed that the method is more accurate than traditional methods. The high cost of multispectral cameras led researchers to focus on the analysis of pure RGB color images. The paper [1] uses a convolutional neural network to reveal the non-linear relationship between a color land image and related vegetation indexes. That network obtains vegetation indexes of various crops. Experiment results show that the obtained values agree with ground-measured indexes. However, they also revealed that the method provides accurate outputs only until appearance of senescence. Paper [9] shows a method to estimate vegetation indexes with a cheap RGBN (RGB + near infrared) camera and machine learning algorithms. Experiment results provide a comparison of the results obtained with a multispectral camera and the predictions of the RGBN camera-based solution to analyze corns under different nitrogen and water treatments. They show that the proposed approach achieved high performance at estimating vegetation indexes with the machine learning model. Study [10] proposes a method to process high-resolution drone images consisting of RGB and near-infrared bands to detect vegetation indexes. The experimental results provide insight into applying drones and neural networks as a solution for precision agriculture.

All the studies above utilize neural networks to estimate vegetation levels. However, as was pointed out above the framework provides accurate results only for immature plants. We believe the reason is that the "high-frequency functions" phenomenon comes into effect [2] and Deep Forest (DF) models could overcome that issue. Deep Forest is a multilayer cascade model based on non-differentiable modules in contrast to deep neural networks. Besides, those models require a small amount of training data due to a small number of parameters. Paper [11] presents a detailed analysis that shows deep forests have sufficient model complexity with enough depth, and the cascaded structure boosts the feature representations layer by layer instead of the predictions. Many experiments show that Deep Forest has comparable performance to deep neural networks; therefore, it has been applied to solve many real-world data and text mining problems. Primary efforts in developing this approach focus on tuning it to solve various machine learning settings. For example, study [12] proposes a deep forest algorithm for multi-instance learning. The experiments show this algorithm achieves competitive results. Yang et al. [13] present a multi-label learning deep forest algorithm, which employed measure-aware feature re-use and layer growth to solve a multi-label learning problem. Paper [14] presents an adaptive weighted Deep forest. The training procedure of this forest assigns weights to each training sample at each level of the model just like the AdaBoost approach.

Although Deep Forests show competitive results on many problems, there is still room for improvements related to considering various feature interactions.

For example, Chen et al. argue that the prediction-based feature representation of Deep Forest is a critical deficiency because the predicted class probabilities deliver very limited information [4]. They present a deep forest model that utilizes high-order interactions of input features to generate more informative and diverse feature representations [4]. They created a generalized version of Random Intersection Trees to reveal stable high-order relationships and apply activated linear combinations to transform them into hierarchical distributed representations. These relationship-based representations obviate the need to store random forests in the front layers, thus greatly improving computational efficiency. The provided experiment results show that the proposed forest achieves competitive classification scores with significantly reduced time and memory costs.

Another way to catch those feature interactions is to use more complex decision splits in ensemble trees. In this study, we replaced standard Random forests in the layers with more complex Kernel forests to do so [6].

3. Kernel Forest-based Methods

3.1. Kernel Forest

Kernel Forest is a Random Forest built from trees with kernel decision splitters [6]. The method to train such trees at the top level follows a general top-down induction procedure; however, it produces quasi-optimal oblique and kernel splits at the decision stump level. At each stump, the algorithm greedily finds a quasi-optimal distribution of classes to subtrees (in terms of impurity minimization) and trains this stump as a binary classifier via optimization of an SVM-like loss function with a margin re-scaling approach [15]. This approach helps optimize the margin between subtree data and arbitrary impurity criteria (Gini impurity, Information gain, etc.).

For each decision stump, parameters of the decision surface are obtained with the following optimization problem:

$$a^* = \arg\max_a -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} a_i a_j K(x_i, x_j) + \sum_{i=1}^{m} a_i, \tag{1}$$

s.t.

$$\sum_{i=1}^{m} \frac{a_i}{L(h_i, -h_i)} \leq \frac{C}{m} \tag{2}$$

where x_i is features of the object with index i, $h_i \in \{-1, +1\}$ defines the target subtree for the sample with index i, a_{ij} is the weight of the training sample i (non-zero for the support vectors), $L(h_i, -h_i)$ reflects the growth of the impurity criterion in case of miss-classification, $K(x_i, x_j) : \mathbb{R}^{\mathfrak{f}} \times \mathbb{R}^{\mathfrak{f}} \to \mathbb{R}$ is a kernel function for objects with the feature-set size f, C is the regularization term, and m is the size of the training dataset.

3.2. Kernel Forest Regressor

Many practical applications in agriculture such as vegetation index detection require solving regression problems. Regressors map object features to some target real values $X \to \mathbb{R}$. In this study, we propose a way how to modify the classification method from [6] (Kernel Forest) to solve those problems. As in the original Kernel Forest, the method utilizes standard top-down induction of a decision tree, and at each step of this induction, it performs training of a kernel-based decision stump.

In a regression tree, the decision stump assigns some real values R_1 and R_2 to the left and right subtrees. The Kernel Forest requires those assignments to be done before actual training of the decision split. Therefore, we need to pick up those values in such a way as to minimize the average distance between all the training samples lying at each subtree. In this study, we utilized the K-means clustering algorithm with $K = 2$ to find those values. K-Means algorithm clusters samples by separating them into groups of equal variance, minimizing within-cluster sum-of-squares, i.e. it fits out goal. Finally, we use the found cluster centroids as values of R_1 and R_2.

Another feature of the method [6] is that it scales the training sample weights accordingly to their effect on the impurity criterion (Gini impurity, information gain, etc.). In case of regression, we use the mean square distance from all the training samples of a particular subtree to the values assigned to this subtree instead of impurity:

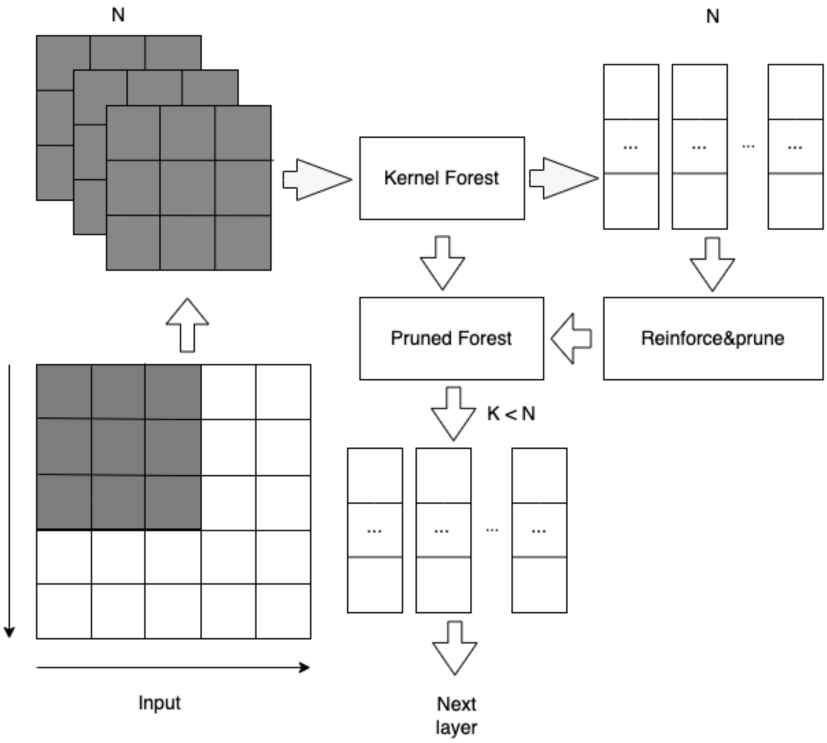

Figure 1. Training of the input layer of the proposed algorithm (Deep Kernel Forest, DKF)

$$l(R, y) = ||R - y||_2^2 \tag{3}$$

3.3. *Deep Kernel Forest*

The method is a modification of the Deep Forest, in which data is processed sequentially on several layers. The input layer has the following structure (Fig. 1). In that layer, the multi-grained scanning [3] generates a set of objects based on each sample from the training set and all those generated objects are labeled with the class of the original sample. These objects are used to train a random kernel forest [6], then the trained forest is strengthened and pruned with the method from [7].

The basic idea of that strengthening procedure is to replace the original class empirical probabilities stored in all tree leaves of a pre-trained forest with the synthetic ones generated by explicitly minimizing a global loss function, according to the averaging rule of random forests. Suppose the forest has T trees with Γ leaves on each tree. Let $\Phi : \mathbb{R}^{\mathfrak{f}} \to \{0, 1\}^{T\Gamma}$ be a function that for any sample x returns the binary vector, whose elements are 1 if x goes to the corresponding decision tree leaf and 0 otherwise.

$$\Phi(x) = (\phi_1(x), \phi_2(x), \dots, \phi_{T\Gamma}(x)) \tag{4}$$

Matrix W contains the corresponding class weight for all the decision tree leaves the ensemble.

$$W = (w_1, w_2, \dots, w_{T\Gamma}) \tag{5}$$

Ren with colleagues define the refined classifier as the following linear function [7]:

$$y = W^* \Phi(x) \tag{6}$$

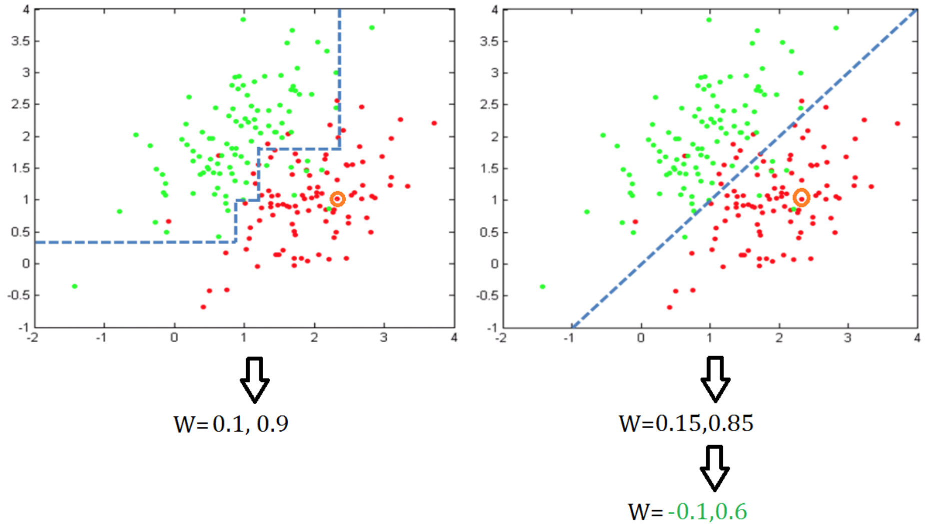

Figure 2. Difference in the feature generation in Deep Forest (left) and Deep Kernel Forest (right)

where W^* can be found with the following SVM-like optimization on a training set with size m.

$$\begin{aligned} W^* &= \arg\min_W \frac{1}{2}||W||_F^2 + \frac{C}{m}\sum_{i=1}^{m} l(y_i, \hat{y}_i), \\ & s.t. y_i = W\Phi(x), \forall i \in [1, m], \end{aligned} \tag{7}$$

where C is a regularization term, and $l(y_i, \hat{y}_i)$ is a loss function.

As a result, the complementary information between trees is exploited, and the fitting power is significantly improved. However, the global optimization in training might cause over-fitting for many tree leaves. To tackle this Ren proposes a global pruning method that alternates between refining all tree leaves and merging the insignificant leaves to reduce the risk of over-fitting and model size. The method is to join two adjacent leaves if the norm of their leaf vectors is close to zero.

After the pruning procedure, the random kernel forest is used to form embeddings of processed samples for the next layer. These embeddings represent the generated synthetic class probabilities from trees of the kernel forest. The embeddings include the original feature features as well as the empirical probability vectors returned by the refined forest trees. In practice, as in the original study [3] we utilize cross-validation to estimate the class probabilities because it reduces the bias of the obtained values. Fig. 2 highlights difference in the feature generation procedure in Deep Forest and Deep Kernel Forest. Deep Kernel Forest allow obtaining less fragmented regions in the original feature-set. The global refinement procedure leads to forming more helpful embeddings for the next layer.

The next layer has the same structure, except it does not perform multi-grained scanning. The following layers can be added to the model until accuracy scores on cross validation keep growing.

4. Datasets

4.1. Standard image recognition datasets

First, we conducted experiments on three standard UCI multi-class datasets, and the CIFAR-10 image dataset. We used USPS, Letter, and MNIST from the UCI [16]. They are devoted to image recognition problems. For example, the MNIST and USPS datasets contain handwritten images of digits, while the Letter dataset contains Latin letters. The CIFAR-10 dataset is also related to image recognition [17]. It contains 32 by 32 colored

doi:10.20944/preprints202210.0070.v1

images of 10 classes (airplane, horse, bird, etc.) with eight gray levels. We apply a simple preprocessing technique to all the image recognition datasets. Namely, we perform feature-level normalization of the data with "MinMaxScaler" and "Normalize" tools from Scikit-Learn [18]. No other complex processing is used.

4.2. RGB-NDVI prediction dataset

We collected a dataset to evaluate RGB to NDVI models as follows. Generally, we have stuck the procedure described in [19]. First, we obtained several multispectral satellite images of rural areas in Europe from April to October 2018. We got the high-resolution multispectral raster data (RGB and Infrared) of the Sentinel-2 satellite from the Copernicus web platform [20]. Then we applied QGIS Desktop software tool to evaluate $NDVI$ based on red and infrared bands. With those bands one can use a simple expression to evaluate that index:

$$NDVI = \frac{I - R}{I + R}, \tag{8}$$

where I is infrared level and R is red level.

Finally, we generated RGB and NDVI images with size 232×232, then we store RGB images as the sample features and NDVI average values as the labels. The size of the obtained dataset is 1000 samples for training and 1000 for validation. We also divided all the dataset into two pieces: the first one covers the data range from April to June (Spring), and the second one covers the range from July to October (Summer/Autumn). Here we utilize a naive presumption that the first subset should contain mostly images of immature plants, while the second one contains images of mature ones. Therefore, accuracy scores would be different for those subsets. As for the classification datasets, we did not perform any complex feature pre-processing because the primary goal of this study is to assess helpfulness of the proposed algorithm modifications rather than achieving best results on particular datasets.

5. Experiment results

In all the experiments we used Deep Forests and Kernel Forests with three layers. We applied a commonly recognized grid search with cross-validation technique to estimate the ensembles hyperparameters: decision stump regularization $\{100, 1000, 3000, 5000\}$, maximum tree depth $\{4, 5, 6, 7, 8\}$, the proportion of features to be considered at each stump $\{0.08, 0.1, 0.2\}$, pruning (up to 0.9) ratio, kernel parameters ($gamma = \{10, 100\}$ for the Gaussian kernel, size of the sliding window $[8 - 128]$, and size of the sliding window step $[2 - 32]$ depending on a dataset. We applied the grid search on sampled subsets of the original datasets because training time of DF and KDF is really long. Finally, we used the obtained hyperparameter values to perform tests on full datasets.

In the first experiment, we assessed the classification quality on commonly recognized datasets. We tested the original Deep Forest (gcForest), Random Kernel Forest (a forest with multivariate decision trees), the Deep Kernel Forest, in which basic classifiers are replaced by Random Kernel Forests, and modifications of the Deep Kernel Forest with pruning and sliding window. We used accuracy to evaluate the classification quality because most studies on UCI and Cifar-10 datasets utilize this score, so we can stay comparable with these results.

Table 1. Classification scores of the tested methods (accuracy).

Dataset	Deep Forest	Kernel Forest	Deep Kernel Forest	Deep Kernel Forest + prune	Slide window + Deep Kernel Forest + prune
MNIST	99.2	99.1	98.0	99.2	**99.4**
USPS	95.9	95.8	93.5	97.8	**97.9**
Letter	96.3	97.4	97.2	98.5	**98.5**
Cifar-10	61.8	58.0	60.3	62.9	**63.2**

Table 1 shows that further complexification of the basic estimators in the Deep Forest without any additional regularization does not lead to any significant improvements in analysis accuracy. We believe that means the complexity of Deep Forest is pretty high, and further increasing that complexity leads to over-fitting. On the other hand, adding a simple tree refinement and pruning [7] leads to notable accuracy growth. The feature and sample re-generation with the sliding window approach [3] leads to significant improvement for Cifar-10 only. We believe this is because images from this dataset have higher resolution and provide more diversity in terms of represented objects, which means the image scaling and transforming can have much effect on classification accuracy.

Table 2. MSE scores of the NDVI prediction.

Interval	Deep Forest	Deep Kernel Forest + prune	AlexNet
Spring	0.006	**0.004**	0.006
Summer/Autumn	0.007	**0.004**	0.006
All	0.007	**0.004**	0.007

In the second experiment we assessed the quality of NDVI prediction with the Deep Forest, Deep Kernel Forest (with pruing and sliding window), and AlexNet neural network [17]. We considered AlexNet in this experiment because it is widely used to predict NDVI in other studies [1]. In the experiments we evaluated mean squared error (MSE) of the predictions. Results from Table 2 show that Deep Kernel Forest can predict the NDVI level more accurately than Deep Forest or AlexNet models. In contrast to [1] we did not detect any dramatic difference for NDVI evaluation in "Spring" and "Summer/Autumn" subsets with AlexNet. We believe this is because first of all, the naive division we used to separate immature plats from senecent ones. Besides, in [1] they use UAV images with larger scale, when "high-frequency functions" effects are more observable. The obtained NDVI prediction error is pretty low, although we did not perform any complex image pre-processing. Therefore, the proposed modified model can be applied to assess NDVI in practical software applications for precise farming.

6. Discussion

The experiments show that the proposed modifications improve quality for both classification and regression tasks. On the one hand, accurate and informative feature representation generation is a cornerstone of cascade models such as Deep Forest. Each layer of Deep Forest for each data sample encodes a feature subspace related to this sample. Kernel Forest detects more homogeneous subspaces than Random Forest and considers complex feature relationships, while Ren's refinement approach helps directly improve those feature representations via optimization of a global loss [7]. On the other hand, algorithms to generate multivariate tree ensembles have significantly lower training speed [6], which remains an open problem.

7. Conclusions

The paper presents a modified Deep Forest that combines the Kernel Forest model and random forest refinement technique. The experiments on commonly recognized image classification datasets show the proposed method significantly outperform the original Deep Forest. Tests on the RGB-NDVI datasets confirm that the proposed method forms accurate predictions for immature and senescent plants. We believe the proposed combination of multi-layer Deep forests and refined Kernel Forests can be considered as a small step towards general-purpose multi-layer models to process non-smooth relationships in data.

The remaining issue of the Deep Kernel Forests is that the multi-grained scanning procedure leads to an exponential growth of the training dataset. In the future, we will try to develop an online modification of the proposed method to tackle that problem.

Funding: The study was financially supported by the Ministry of Education and Science of the Russian Federation (agreement no. 075-15-2020-805 from 2 October 2020).

Data Availability Statement: The datasets generated and analysed during the current study are available in the RGB-NDVI repository at http://keen.isa.ru/ndvi. The Kernel Deep Forest implementation used in the current sudy is available at https://github.com/masterdoors/kernel_trees/tree/master/sources/cascade.

References

1. Khan, Z.; Rahimi-Eichi, V.; Haefele, S.; Garnett, T.; Miklavcic, S. J. Estimation of vegetation indices for high-throughput phenotyping of wheat using aerial imaging. *Plant methods* **2018**, *14(1)*, 1–11.
2. Tancik, M.; Srinivasan, P.; Mildenhall, B.; Fridovich-Keil, S.; Raghavan, N.; Singhal, U. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems* **2020**, *33*, 7537–7547.
3. Zhou, Z. H.; Feng, J. Deep forest *arXiv preprint* **2017** *arXiv:1702.08835*.
4. Chen, Y. H.; Lyu, S. H.; Jiang, Y. Improving deep forest by exploiting high-order interactions. In Proceedings of the 2021 IEEE International Conference on Data Mining (ICDM), IEEE, 2021; 1030-1035.
5. Yang, B.B.; Shen, S.Q.; Gao, W. Weighted oblique decision trees. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 2019; 5621-5627.
6. Devyatkin, D. A.; Grigoriev, O. G. Random Kernel Forests. *IEEE Access* **2022** *10*, 77962–77979.
7. Ren, S.; Cao, X.; Wei, Y.; Sun, J. Global refinement of random forest. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015; 723-730.
8. Wang, Y.; Wang, X.; Jian, J. Remote sensing landslide recognition based on convolutional neural network. *Mathematical Problems in Engineering* **2019**.
9. Wang, L.; Duan, Y.; Zhang, L.; Rehman, T. U. Ma, D., Jin, J. Precise estimation of NDVI with a simple NIR sensitive RGB camera and machine learning methods for corn plants. *Sensors* **2020**, *20(11)*, 3208.
10. El Hoummaidi, L.; Larabi, A.; Alam, K. Using unmanned aerial systems and deep learning for agriculture mapping in Dubai. *Heliyon* **2021** *7(10)*, e08154.
11. Lyu, S. H.; Yang, L.; Zhou, Z.-H. A Refined Margin Distribution Analysis for Forest Representation Learning, *Advances in Neural Information Processing Systems* **2019** *32*, 5530-–5540.
12. Ren, J.; Hou, B.; Jiang, Y. Deep forest for multiple instance learning. *Journal of Computer Research and Development* **2019** *56(8)*, 1670-–1676.
13. Yang, L.; Wu, X.; Jiang, Y.; Zhou, Z. Multi-label learning with deep forest, arXiv preprint, 2019, arXiv: 1911.06557.
14. Utkin, L.V.; Konstantinov, A.V.; Chukanov, V.S.; Kots, M.V.; Meldo, A.A. A new adaptive weighted deep forest and its modifications. *International Journal of Information Technology Decision Making* **2020** *19(04)*, 963-–986.
15. Tsochantaridis, I.; Joachims, T.; Hofmann, T.; Altun, Y.; Singer, Y. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research* **2005** *6(9)*, 15–64.
16. Murphy, P.M.; Aha, D.W. UCI Repository of machine learning databases. Irvine, CA, University of California, Department of Information and Computer Science; 1991.
17. Krizhevsky, A. Learning multiple layers of features from tiny images *Technical report* University of Toronto, 2009.
18. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **2011** *12*, 2825–2830.
19. Plants vs CO2. Available online: https://github.com/GrHalbgott/Plants-vs-CO2 (accessed on 27.08.2022).
20. Copernicus web platform. Available online: https://scihub.copernicus.eu (accessed on 22.08.2022).