# Identification of antihypertensive peptides from lupine using a machine learning approach

*Margarita* Terziyska[1,*], *Zhelyazko* Terziyski[2], *Ivelina* Desseva[3], *Alexandra* Garmidolova[3] and *Dasha* Mihaylova[4]

[1] Department of Mathematics, Physics and Information technologies, University of Food Technologies, 26, Maritza blvd, Plovdiv, Bulgaria
[2] Department of Computer Science and Mathematics, Trakia University, Stara Zagora, Bulgaria
[3] Department of Analytical Chemistry and Physical Chemistry, University of Food Technologies, 26, Maritza blvd, Plovdiv, Bulgaria
[4] Department of Biotechnology, University of Food Technologies, 26, Maritza blvd, Plovdiv, Bulgaria

**Abstract.** Bioactive products with antihypertensive biological activity, isolated from natural sources, have been the subject of growing interest in recent years. This is due to their widespread use in medicine for the treatment and prevention of various diseases, as well as dietary supplements for athletes or their inclusion in diets for overweight people. One such source is Lupine. Lupine beans are delicious and useful. They can be used in food as a nutritional source of vegetable proteins. They are also rich in polyphenols, carotenoids, and phytosterols. The approaches to screen antihypertensive peptides, based on information technologies and more concretely on machine learning, doubtlessly have higher throughput and rapid speed than the in vivo and in vitro procedures. Therefore, the scientific literature abounds with articles offering various artificial intelligence algorithms for predicting food-derived antihypertensive peptides. In this study, an Adaptive Boosting (AdaBoost) algorithm was developed for these purposes. The results showed that the AdaBoost model as a novel auxiliary tool is feasible to screen for antihypertensive peptides derived from food, with high throughput and high efficiency.

## 1 Introduction

Lupine is a plant from the legume family, whose seeds are rich in vitamins and minerals vital for the body. It contains proteins that help reduce cholesterol and triglycerides. Lupine seeds support liver function and regulate blood pressure, protect the heart and blood vessels, and protect against atherosclerotic lesions. In addition, many amino acids such as isoleucine, histidine, threonine, leucine, valine, tryptophan, and lysine are also found in lupine. In [1-4], the current knowledge about lupine protein is summarized, as well as the various benefits of bioactive peptides obtained from lupine seeds - anti-inflammatory, antidiabetic, antihypertensive, antioxidant, antibacterial, and hypocholesterolemic actions.

Cardiovascular disease is a prevalent global health issue. The interest in antihypertensive peptides is increasing due to their potential as safer alternatives for commercial medicines. These peptides possess the ability to inhibit angiotensin-converting enzyme (ACE) (EC 3.4.15.1), an enzyme responsible for transforming the decapeptide angiotensin I into the octapeptide angiotensin II. The last type is recognized for causing a rise in blood pressure [5].

Peptides with ACE-inhibitory activity can be isolated from a wide range of plant and animal sources [6]. In contrast to synthetic inhibitors, plant-derived ACE inhibitory peptides are promising alternatives due to their high bioavailability and lack of side effects - taste disturbances, dry cough, skin rashes [7 - 9].

In this paper, the focus is on antihypertensive peptides extracted from lupine. Peptides with this type of biological activity are used in the treatment of people with high blood pressure. Hypertension is a serious problem and affects up to 30 % of the adult population in developed and developing countries. It is often referred to as the "silent killer". There are not a few cases where it leads to the most serious complications - stroke or myocardial infarction.

Various methods for predicting food-derived antihypertensive peptides with artificial intelligence methods have been proposed in the scientific literature. A novel model for the prediction of antihypertensive peptides derived from food, based on the eXtreme Gradient Boosting (XGBoost) algorithm was developed in [10]. The obtained results showed that the XGBoost model achieved outstanding performance, with an accuracy of 86.50% which, according to the authors, is much better than the other dominating machine learning models such as Support Vector Machine (SVM), Random Forest (RF),

* *Corresponding author:* mterziyska@uft-plovdiv.bg

K-Nearest Neighbor (K-NN). A deep learning ensemble method that combines convolutional neural network (CNN) and SVM classifiers is proposed in [11]. The authors declare that this boosted predictor yields 95% accuracy on the benchmark data set and 88.9% accuracy on the independent data set. In [12], a long short-term memory (LSTM) algorithm-based deep learning model is presented, which could predict the $IC_{50}$ value of the peptide in inhibiting ACE activity. The mean accuracy, average sensitivity, and average specificity of the model were 85.20%, 84.92%, and 85.43%, respectively. A deep model is constructed in [13] by combining a convolutional neural network (CNN) and a gated recurrent unit (GRU) for accurate prediction of antihypertensive peptides. As a result, a structure was obtained that reduces the dimensionality of the features and running time and provides a very high prediction accuracy of 96.23%. The authors of this paper have also proposed a deep learning model [13]. The accuracy obtained with it is 85.42%.

In the present study, an Adaptive Boosting (AdaBoost) model was proposed to predict antihypertensive peptides derived from lupine. This model was compared to other widely used models such as SVM and Decision tree (DT). Only lupine peptides that have been shown in the scientific literature to have antihypertensive properties were included in the test data.

## 2 Materials and methods

The process of predicting the biological activity of peptides goes through several stages (Fig. 1) - data preparation; feature selection and coding; building a predictive model, and evaluating the model.

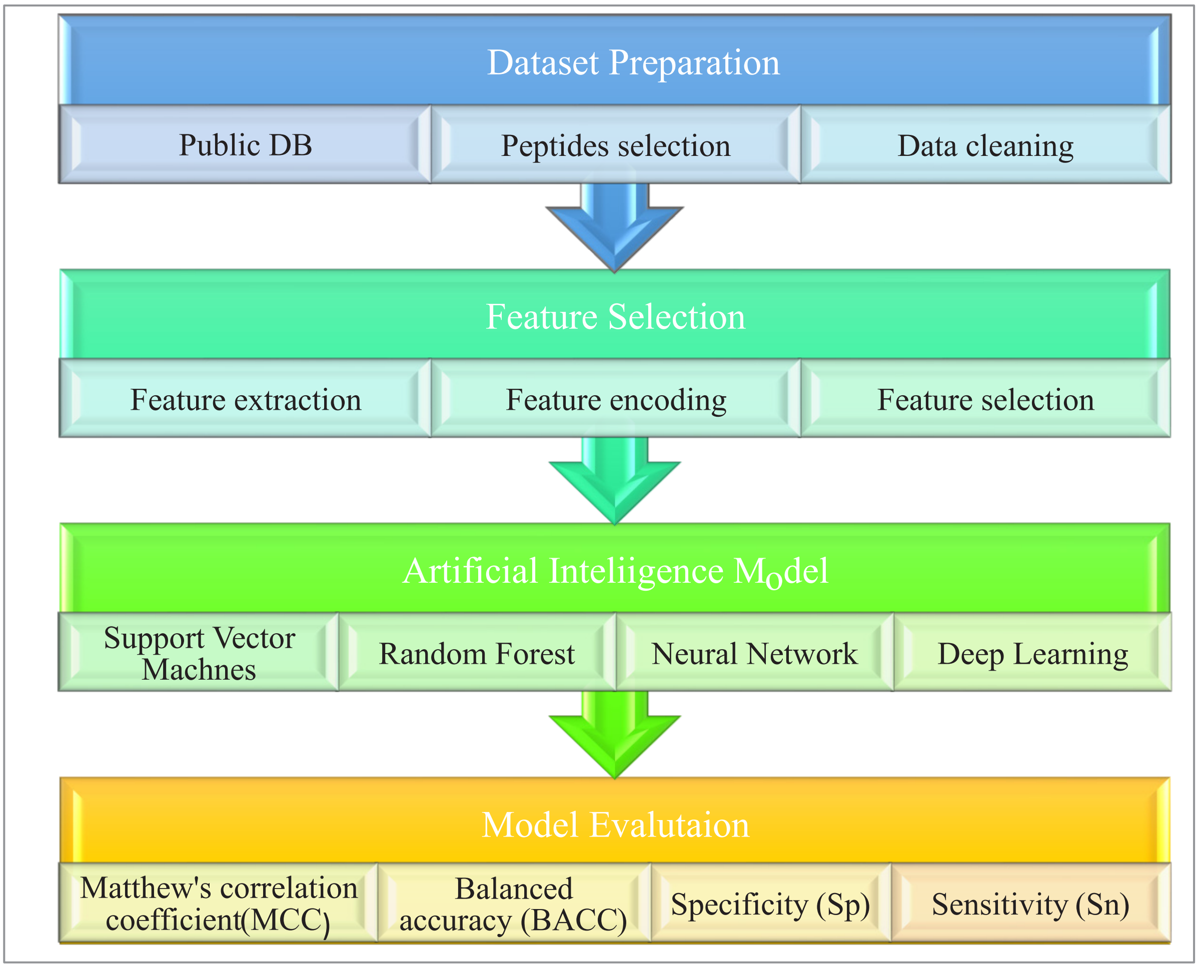


**Fig. 1.** Stages of the process of predicting biological activity of peptides

### 2.1. Data preparation

The prediction accuracy of any machine learning algorithm depends on the quantity and quality of the data. The data must be credible, clean, and without repetitions. The presence of redundancy in the training data sets can affect the accuracy of model performance and often leads to overtraining.

When preparing data for the prediction of biological activity of peptides, two classes of samples should be prepared - positive and negative. Positive are those that have already been experimentally established to possess the corresponding activity. Peptides that do not possess this activity are negative. There are very few experimentally confirmed negative samples in the scientific literature. Therefore, the formation of the negative dataset is done in one of the following ways:

a) random peptides are selected, with a function other than the desired one;

b) toxic peptides are selected;

c) randomly generated peptides.

In this study, the data are from the web-based openaccess PepLab (www.pep-lab.info) platform [15]. The positive dataset included 523 peptide sequences with antihypertensive activity. The negative dataset was formed by extracting from PepLab 523 peptide sequences but with activity other than antihypertensive. In this way, a training dataset of 1046 peptide sequences was formed. The test dataset includes 16 peptides that have been proven in the scientific literature to be extracted from lupine and possess antihypertensive properties [16, 17].

### 2.2. Feature selection

In recent years, the amount of information that artificial intelligence models have to process has increased tremendously. Therefore, it is important to examine the input features and optimize them to reduce their number. For this purpose, from a given input vector with a certain number of features, only a part of them is selected. This subset includes only the features that most influence the prediction process. Those features that do not have a significant impact are dropped. The Filter Feature Selection approach is used to obtain the significant features. Filter feature selection is used to pre-extract and select the most informative features that have a strong relationship with the target variable. It is a model that is based on statistical measures and data properties that are independent of the machine learning model i.e. external criteria are used to evaluate the model.

Statistical measures, correlations, and other metrics are used to assess the importance of features that are computed on the datasets. This makes the Filter feature selection method faster and more efficient in terms of performance compared to other feature selection methods because it does not require training a machine learning model for each of the selected subsets of data.

In addition to feature selection before model training, Filter feature selection can also be used for data pre-processing.

### 2.3. Model design

At this stage of the process, a predictive model is designed by artificial intelligence. In the present study, it is the AdaBoost or Adaptive Boosting model.

The AdaBoost algorithm belongs to ensemble models, which are a statistical approach that enables the improvement of learning algorithms. AdaBoost has established itself as a well-researched, effective approach from the theory of artificial intelligence and machine learning. With it, each training vector from the sample is associated with a weight, which determines the probability that it will be selected for training at a certain stage of an individual classifier. The goal is to obtain a classifier that minimizes the expected error for different distributions of the training sample. At the beginning, all weights have the same value – 1/N. In the training process, if the training vector is classified correctly, then using a deterministic correction rule, its weight is reduced and, accordingly, the probability of being selected again during training. Conversely, if it is misclassified, the likelihood of it being used again increases. The Adaboost algorithm works in the following steps:

1. The Adaboost algorithm selects a training subset randomly.

2. It iteratively trains the AdaBoost machine learning model by selecting the training set based on the accurate prediction of the last training.

3. It assigns a higher weight to wrong-classified observations so that in the next iteration these observations will get a high probability for classification.

4. It assigns the weight to the trained classifier in each iteration according to the accuracy of the classifier. The more accurate classifier will get a higher weight.

5. This process iterates until the complete training data fits without any error or until reaches to the specified maximum number of estimators.

6. To classify, perform a "vote" across all of the learning algorithms you built.

### 2.4. Model evaluation

The following evaluation indicators are most often used to evaluate the prediction performance of artificial intelligence models - classification accuracy (CA), specificity (Sp), sensitivity (Sn), precision (Pr), harmonic mean of precision and sensitivity (F1) and Matthew's correlation coefficient (MCC). They are calculated with the following formulas:

$$CA = \frac{TN+TP}{TN+TP+FN+FP} \tag{1}$$

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN+FP}} \tag{2}$$

$$Sn = \frac{TP}{TP+FN} \tag{3}$$

$$Pr = \frac{TP}{TP+FP} \tag{4}$$

$$MCC = \frac{TP.TN+FP.FN}{\sqrt{(TP+FN).(TP+FP).(TN+FP).(TN+FN)}} \tag{5}$$

$$F1 = \frac{2TP}{2TP+FP+FN} \tag{6}$$

where TP and TN are the number of positive and negative samples correctly classified by the predictive model. FP and FN represent the number of positive and negative samples misclassified.

 https://doi.org/10.1051/bioconf/202410202005

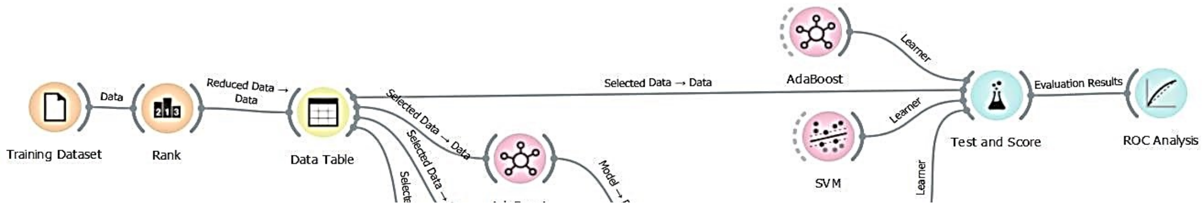


**Fig. 2.** Classification algorithm in orange

### 2.5. Software tools

Two software tools were used to implement the process of predicting the biological activity of lupine peptides. Initially, the open-access web-based platform PepLab was used to generate the datasets. It includes a database of verified peptide sequences extracted from foods. Also, PepLab was then used to transform the peptide sequences into digital form. Finally, Orange Data Mining software was used to realize the AdaBoost model. It is an open source machine learning and data mining toolkit for data analysis through Python scripting and visual programming [18].

## 3 Results and discussion

As already noted, the positive and negative datasets were downloaded from the Peplab platform. The distribution of essential amino acids in the positive dataset is presented in Fig. 3. It can be seen that the most common amino acids in peptides with antihypertensive activity are P at 16.1%, V at 8.7%, and L at 8.5%. Fig. 4 shows the distribution of peptides according to their molecular mass. The majority of them, 64.6%, have a molecular mass between 468.51 and 1008.6 Da, i.e. short sequences predominate. Fig. 5 shows the distribution of peptides with antihypertensive activity included in the positive dataset, according to their Hydrophobicity index (GRAVY). Approximately 85% of the peptides have a hydrophobicity index between -1.34 and 1.23.

Next is the widget Rank, which allows to reduction of the size of the input data using different methods, like information gain, chi-square, etc. For the study, the **information gain ratio**, which is an entropy-based feature evaluation method; **ReliefF**, which is a filter based approach for high-dimensional data reduction; **Fast Correlation-based Feature Selection (FCBF)**, which also is a kind of filter approach and **Chi-square** method, which computes the chi-squared statistical value for all

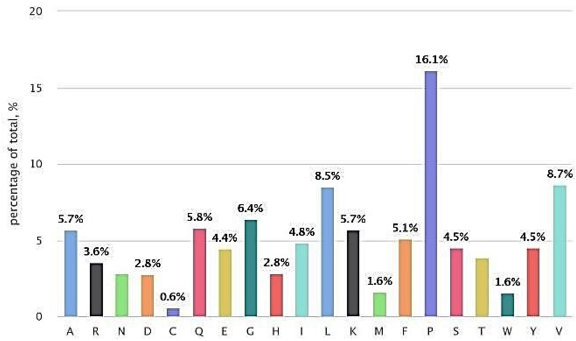


**Fig. 3.** Amino acid distribution in the positive dataset

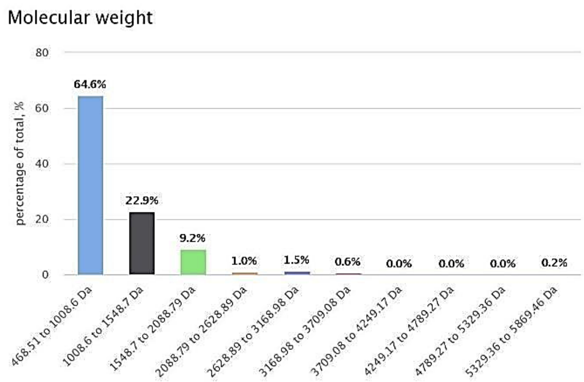


**Fig. 4.** Molecular weight distribution in the positive dataset

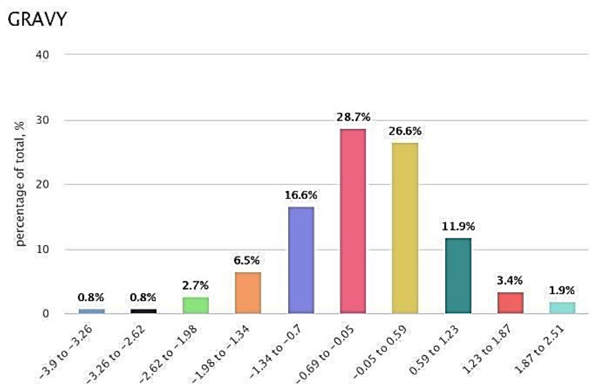


**Fig. 5.** GRAVY distribution in the positive dataset

 https://doi.org/10.1051/bioconf/202410202005

**Table 1.** Decision Tree, SVM and AdaBoost comparison results

| Rank widget | Model | AUC | CA | F1 | Precision | Recall |
|---|---|---|---|---|---|---|
| **$x^2$** | Decision Tree | 0.716 | 0.704 | 0.704 | 0.705 | 0.704 |
| **$x^2$** | SVM | 0.497 | 0.480 | 0.471 | 0.478 | 0.480 |
| **$x^2$** | AdaBoost | 0.779 | 0.715 | 0.715 | 0.715 | 0.715 |
| **InformationGainRatio** | Decision Tree | 0.760 | 0.724 | 0.723 | 0.728 | 0.724 |
| **InformationGainRatio** | SVM | 0.435 | 0.451 | 0.325 | 0.308 | 0.451 |
| **InformationGainRatio** | AdaBoost | 0.768 | 0.722 | 0.721 | 0.726 | 0.722 |
| **ReliefF** | Decision Tree | 0.705 | 0.705 | 0.705 | 0.705 | 0.705 |
| **ReliefF** | SVM | 0.454 | 0.452 | 0.446 | 0.450 | 0.452 |
| **ReliefF** | AdaBoost | 0.800 | 0.738 | 0.738 | 0.738 | 0.738 |
| **FCBF** | Decision Tree | 0.736 | 0.703 | 0.702 | 0.706 | 0.703 |
| **FCBF** | SVM | 0.444 | 0.450 | 0.450 | 0.450 | 0.450 |
| **FCBF** | AdaBoost | 0.758 | 0.700 | 0.700 | 0.700 | 0.700 |

**Table 2.** Decision Tree, SVM, and AdaBoost predictions

| | **SVM** | | | **AdaBoost** | | | **Decision tree** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **decision** | **Class 0 probability** | **Class 1 probability** | **decision** | **Class 0 probability** | **Class 1 probability** | **decision** | **Class 0 probability** | **Class 1 probability** |
| **1** | Class 0 | 0.701685 | 0.298315 | Class 0 | 0.678317 | 0.321683 | Class 1 | 0.447722 | 0.552278 |
| **2** | Class 0 | 0.701685 | 0.298315 | Class 1 | 0.394011 | 0.605989 | Class 1 | 0.458029 | 0.541971 |
| **3** | Class 0 | 0.701685 | 0.298315 | Class 0 | 0.657007 | 0.342993 | Class 1 | 0.463729 | 0.536271 |
| **4** | Class 1 | 0.327273 | 0.672727 | Class 1 | 0.464875 | 0.535125 | Class 1 | 0.449986 | 0.550014 |
| **5** | Class 0 | 0.701685 | 0.298315 | Class 1 | 0.29299 | 0.70701 | Class 0 | 0.544692 | 0.455308 |
| **6** | Class 1 | 0.327273 | 0.672727 | Class 1 | 0.363452 | 0.636548 | Class 1 | 0.414556 | 0.585444 |
| **7** | Class 1 | 0.327273 | 0.672727 | Class 1 | 0.278 | 0.722 | Class 1 | 0.397464 | 0.602536 |
| **8** | Class 1 | 0.327273 | 0.672727 | Class 1 | 0.39122 | 0.60878 | Class 0 | 0.5 | 0.5 |
| **9** | Class 1 | 0.327273 | 0.672727 | Class 1 | 0.385495 | 0.614505 | Class 0 | 0.527105 | 0.472895 |
| **10** | Class 1 | 0.327273 | 0.672727 | Class 1 | 0.432022 | 0.567978 | Class 1 | 0.423518 | 0.576482 |
| **11** | Class 1 | 0.0580913 | 0.9419087 | Class 1 | 0.102049 | 0.897951 | Class 1 | 0.29651 | 0.70349 |
| **12** | Class 1 | 0.327273 | 0.672727 | Class 1 | 0.420521 | 0.579479 | Class 1 | 0.394634 | 0.605366 |
| **13** | Class 0 | 0.701685 | 0.298315 | Class 1 | 0.394011 | 0.605989 | Class 1 | 0.458029 | 0.541971 |
| **14** | Class 1 | 0.327273 | 0.672727 | Class 1 | 0.237124 | 0.762876 | Class 1 | 0.373208 | 0.626792 |
| **15** | Class 1 | 0.327273 | 0.672727 | Class 1 | 0.372513 | 0.627487 | Class 1 | 0.400581 | 0.599419 |
| **16** | Class 1 | 0.0580913 | 0.9419087 | Class 1 | 0.223375 | 0.776625 | Class 1 | 0.316734 | 0.683266 |

features with respect to each class and ranks the features based on the value according to the formula:

$$\chi^2 = \sum_{i=1}^{R}\sum_{j=1}^{C}\frac{(O_{ij}-E_{ij})^2}{E_{ij}} \quad (7)$$

where:

*Oij* = observed frequency of class ‘j’, given attribute value ‘i’
*Eij* = expected frequency of class ‘j’, given attribute value ‘i’
*R* = number of distinct values of attribute ‘a’ C = number of classes.

The next widget from Fig. 2 is a Data Table that allows the to be viewed reduced size dataset. The Test and Score widget displays a table of the main parameters for evaluating models (formulas 1-6). The results obtained in this widget are summarized and presented in Table 1. As can be seen from it, no matter what feature selection method is used, the AdaBoost model makes more accurate predictions compared to SVM and Decision tree models.
From the ROC Analysis widget Fig. 6 and Fig. 7 have been obtained. They show the ROC (receiver operating characteristic) curves showing the performance of classification models.
Finally, the trained models make a prediction to which class completely unknown peptide sequences belong to them. These peptides are taken from the scientific literature and have been shown to possess antihypertensive activity. Thus, all three models should assign these sequences to class 1.

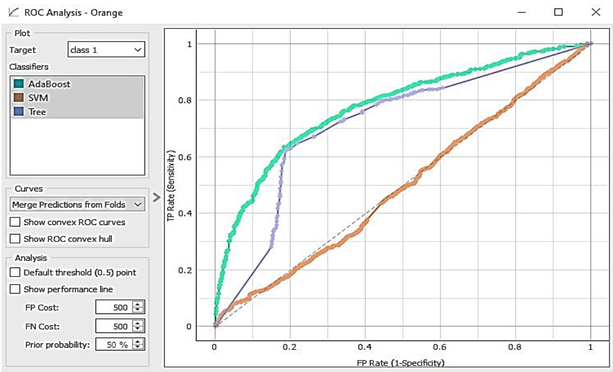


**Fig. 6.** ROC curve of Decision Tree, SVM and AdaBoost for antihypertensive peptides

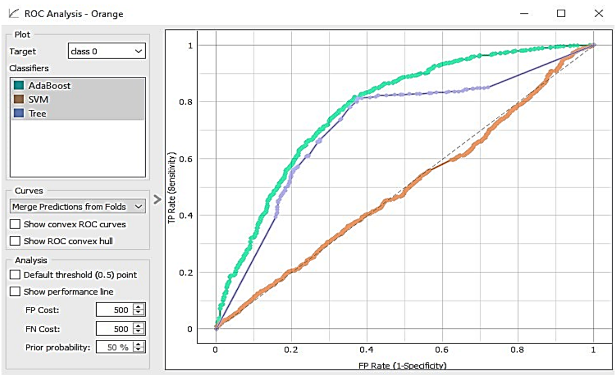


**Fig. 7.** ROC curve of Decision Tree. SVM and AdaBoost for non-antihypertensive peptides

The results of the Predictions widget show that the models still make errors (Table 2). It can be seen that again the AdaBoost model did the best, which "wronged" only two peptides by defining them as negative. The Decision Tree model was wrong for three of the peptide sequences, and the SVM model - as many as 5 peptides. This again confirms the conclusion that among the three models studied the most accurate is AdaBoost and the least accurate is the SVM model.

## Conclusions

In this paper, an AdaBoost model was used to predict the antihypertensive activity of peptides extracted from lupine. The proposed model is compared with two other models from machine learning - SVM and Decision tree.

The main conclusion that can be drawn are that among these three models, the most accurate classification can be made with the AdaBoost model. Furthermore, it should be noted that the choice of feature selection method (at least those methods that have been studied) does not significantly affect the prediction accuracy.

## References

1. A. Garmidolova, I. Desseva, D. Mihaylova. D., A. Lante, App. Sci. **12**, 3766 (2022)
2. S. Sharma, R. Singh, S. Rana, Int. J. Bioautom. **15**, 223 (2011)
3. S. Chakrabarti, F. Jahandideh, J. Wu, Biomed. Res. Int. **2014**, 1 (2014)
4. K. J. Rutherfurd-Markwick, Br. J. Nutr. **108(Suppl.)**, 149 (2012)
5. G. Boschin, G. M. Scigliuolo, D. Resta, A. Arnoldi, Food Chem. **145**, 34 (2014)
6. C. C. Udenigwe, A. Mohan, J. Func. Foods **8**, 45 (2014)
7. S. Rayaprolu, N. Hettiarachchy, R. Horax, J. Am. Oil Chem. Soc. **92**, 1023 (2015)
8. A. Kannan, N. Hettiarachchy, M. Marshall, S. Raghavan, H. Kristinsson, J. Sci. Food Agric. **91**, 1920 (2011)
9. R. Wang, X. Lu, Q. Sun, J. Gao, L. Ma, J. Huang, Int. J. Mol. Sci. **21**, 1059 (2020)
10. L. Wang, D. Niu, X. Wang, J. Khan, Q. Shen, Y. Xue, Foods, **10**, 550 (2021)
11. A. Rauf, A. Kiran, M. T. Hassan, S. Mahmood, G. Mustafa, M. Jeon, Appl. Sci. **11**, 2316 (2021)
12. W. Liao, S. Yan, X. Cao, H. Xia, S. Wang, G. Sun, K. Cai, Molecules. **28**, 4901 (2023)
13. H. Shi, S. Zhang, Interdiscip. Sci.: Comp. Life Sci. **14**, 879 (2022)
14. M. Terziyska, I. Desseva, Z. Terziyski, In The International Symposium on Bioinformatics and Biomedicine (pp. 236-246). (Cham: Springer International Publishing 2020)
15. Z. Terziyski, M. Terziyska, I. Deseva, S. Hadzhikoleva, A. Krastanov, D. Mihaylova, E. Hadzhikolev, App. Sci. **13**, 961 (2023)
16. E. B. Muñoz, D. A. Luna-Vital, M. Fornasini, M. E. Baldeón, E. G. de Mejia, J. Func. Foods. **45**, 339 (2018)
17. G. J. Fadimu, C. Y. Gan, O. A. Olalere, A. Farahnaky, H. Gill, T. Truong, Food Chem. **407**, 135082 (2023)
18. J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, M. Milutinovič, M. Možina et al., J. Machine Learning Res. **14**, 23492353 (2013)