# A Distributed Classification Method for Real-time Healthcare Data Processing

Vladislav Kaverinskiy[1,†], Oleksandr Palagin[2,†], Kyrylo Malakhov[2,*,†]

[1] *I.M. Frantsevich Institute for Problems of Material Sciences of the National Academy of Sciences of Ukraine, 3 Krzhizhanovsky str., Kyiv, 03142, Ukraine*

[2] *Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine, 40 Glushkov ave., Kyiv, 03187, Ukraine*

**Abstract**

A complex classifier based on the Random Forest approach is developed. The main essence of the classifier is that it consists of several (in the current realization 4) binary classifier modules; each is trained to distinguish its class separation. A truth table has been developed which allows basing on the binary classifiers results to reach more precise and fine classification into a larger number of classes (5 in the current realization). The developed classification service has been trained and tested using a medical data set obtained at the Ternopil Regional Clinical Psychoneurological Hospital and matches the patients' depression level estimation with arterial pulsations oscillograms analysis data. The developed classifier exhibits a high accuracy range (up to 97%) in distinguishing all 5 levels. SHAP testing of the input factors' significance has been performed on the test data sample. Also were studied in classes' confusion and benchmarks for sequential and parallel classification modules performance.

**Keywords**

Classification, Machine Learning, Distributed Classification Method, Data Science, Random Forest

## 1. Introduction

Classification AI systems have become a cornerstone in various fields such as medical diagnosis, finance, and natural language processing. These systems are designed to categorize data into predefined classes, enabling automated decision-making and pattern recognition. The effectiveness of classification systems largely depends on the algorithms used, with several notable approaches including support vector machines (SVMs), neural networks, and decision tree-based methods.

One of the most popular and powerful classification methods is the Random Forest algorithm. Introduced by Leo Breiman in 2001 [1], the Random Forest approach is an ensemble learning technique that builds multiple decision trees during training and outputs the mode of the classes for classification tasks or the mean prediction for regression tasks. This method combines the simplicity of decision trees with the power of ensemble learning, making it robust and versatile.

The Random Forest algorithm creates a 'forest' of decision trees, each trained on a random subset of the data. This random selection process, known as bootstrapping, and the randomness introduced in the selection of features for each split, help in reducing overfitting – a common problem with individual decision trees. By aggregating the predictions of multiple trees, Random Forests improve generalization and accuracy.

*ProfIT AI 2024: 4th International Workshop of IT-professionals on Artificial Intelligence (ProfIT AI 2024), September 25–27, 2024, Cambridge, MA, USA*

* Corresponding author.

† These authors contributed equally.

hisie@ukr.net (V. Kaverinskiy); palagin_a@ukr.net (O. Palagin); k.malakhov@outlook.com (K. Malakhov)

0000-0002-6940-579X (V. Kaverinskiy); 0000-0003-3223-1391 (O. Palagin); 0000-0003-3223-9844 (K. Malakhov)

Numerous studies have demonstrated the efficacy of Random Forests in various domains. For instance, in the medical field, Random Forests have been used to classify and predict disease outcomes based on patient data [2]. In finance, they are employed for credit scoring and fraud detection [3]. Furthermore, environmental scientists use Random Forests for ecological and climate modelling due to their ability to handle complex interactions between variables [4].

Recent advancements have further enhanced the capabilities of Random Forests. Techniques such as the implementation of extremely randomized trees [5] and the integration with deep learning methods are areas of active research. These advancements aim to combine the strengths of Random Forests with other powerful techniques to tackle increasingly complex datasets and tasks.

As mentioned, AI classification systems are widely used for medical purposes. A medical dataset was used to train and test the developed in the current study system. Nevertheless, the approach presented here is not limited to the medical tasks and the investigated subject area but could be applied to various scopes, especially those that have to operate with datasets with a wide number of features and a comparably limited number of data rows. Below we will give a brief description of the subject we focused on and the dataset we have operated with.

Measuring blood pressure is a crucial procedure for doctors at all stages of medical care [6, 7]. The advent of electronic blood pressure meters has enhanced the process by providing more detailed and informative data. Some advanced models of electronic pulse meters can register arterial pulsations, allowing them to calculate blood pressure values and transmit this information for further analysis [8, 9]. These readings contain valuable information that offers deeper insights into the overall state of the body and its systems [6, 10, 11, 12].

In the analysis process, arterial oscillograms are created from the blood pressure curve for further examination. The practical application of arterial oscillography opens up significant opportunities for researchers and clinicians. It enables the use of machine learning methods to develop automated diagnostic AI systems that can assess a patient's condition based on objective measurement results.

This particular research focuses on creating a classifier capable of estimating the level of depression. Unlike patient surveys, which can be biased or outdated, this classifier would use blood pressure oscillograms as indicators of the body's condition. These oscillograms can serve as markers for various health states, providing a more objective and reliable basis for diagnosis.

Thus, **the main objective** of the presented study is to develop and test an AI classification system capable of dealing with datasets with a wide number of features and a comparably limited number of data rows and accurately distinguishing all the given classes. To solve the problem the following tasks are to be fulfilled: to find the most valuable features; to train and tune hyperparameters of the classifiers to increase accuracy; to test the developed classifier and find its peculiarities in classes mismatch and the features' significance according to the SHAP test.

## 2. Initial Dataset Characterization and Numerical Research Procedure

Patients aged 32 – 65 with mental disorders were treated at Ternopil Regional Clinical Psychoneurological Hospital. Their mental and psychotic disorders were assessed using the Hospital Depression Rating Scale (HDRS) and the DASS-21 depression, anxiety, and stress scale. Diagnoses included bipolar affective disorder with a current episode of depression and depressive disorders without psychotic features.

Due to the high number of input parameters (1030 factors) and a relatively small dataset (181 patients), the parameters were analyzed separately to avoid distorted results. The patients were divided into five groups based on their initial depression levels. Groups 1 to 4 were formed based on questionnaire results, with increasing depression levels. Group 5 included patients undergoing treatment in the hospital.

Several studies aimed at telerehabilitation and informational medical rehabilitation support have been carried out in the V. M. Glushkov Institute of Cybernetics of NAS of Ukraine, which have resulted in a cloud telerehabilitation platform creation [13, 14, 15, 16]. The current study is a

continuation of the efforts in the mentioned fields that could provide there a classification service for medical purposes.

To initial analyze the dataset the "Lazy Predict" technique from Scikit Learn was used. This method automatically learns several classifiers and ranks them by accuracy. The dataset had over a thousand parameters but only 181 data rows, many of which were inter-correlated. Thus, selecting key parameters was crucial for building a reliable classifier.

Not all features within the selected group were valuable for classification. Practical experience showed that using raw data did not yield a satisfactory classifier. Hence, feature correlations within each class were considered. Some features showed strong correlations in specific classes, serving as distinguishing factors.

To improve classification accuracy, input parameters were selected based on a combination of features and their value products, focusing on correlated patterns across classes. Additional effective features were identified through a systematic search methodology, aiming to enhance accuracy. This process involved adding new features from correlated pairs, retraining the classifier by the "Lazy Predict" technique, and selecting features that improved accuracy.

In the final stage, various classification options were explored using the selected features, developing a complex classifier capable of distinguishing different class groups and detecting multiple classes within the dataset. In this stage not only the best classifiers by the "Lazy Predict" ranging were tested but also ones that have somewhat lower positions, because due to the hyperparameters tuning, they may exhibit even more accuracy values. All the trained classifier objects were made to distinguish two classes (being binary) each of those may include one or several of the a priori classes. Then a truth table has been formed to define the classifier object results combination as belonging of the patient to one of the a priori classes with the corresponding additional remarks if necessary. The truth table in JSON formalized condition becomes a part of the automatic classification service.

## 3. Results and Discussion

### 3.1. Input Parameters Selection

From the vast number of given in the data set parameters, ULF ones were selected to be the base. ULF is the power of ultra-low-frequency (ultra-slow) oscillations (with a frequency of less than 0.003 Hz.); the functional value of ULF waves consists in the integration and adaptation of the functional state of the organism to the influence of external factors; coordinates the functional activity of all body systems following changes in the external environment. According to the UMAP analysis, ULF factors could quite accurately split the patients into 2 groups: 1 – classes 1 and 2 (lighter levels of depression); 2 – classes 3, 4 and 5 (more serious depression cases). Those two groups then have been chosen for the initial classification fitting through the "Lazy Predict" technique.

A cauterization on reduced dimensions using UMAP also has shown that 11 patients form a completely separated cluster. Among them were ones of each of the a priori classes. Because they do not match the found main classes' separation they have been excluded from the training dataset. It has been found that the patients from that small cluster have some particular feature values, namely: zero values of ULF_100-70 and ULF_per_100-70 along with high values of ULF_70-100 (> 2000) and ULF_70-end (>300). This observation has formed the basis of the filter which does not regard the input data for the further classification process if they match the named conditions.

Direct usage of the ULF factors values (even normalized), however, did not allow to reach high classification accuracy. A particular essence of the initial features selection technique accepted in this study is finding the feature pairs which have a strong correlation for one of the classes but are weak in another or those that are correlated in both classes but with different singe of the correlation coefficient. The products of such correlated features were used along with the direct feature values to be in the set for the subsequent selection of the classifier.

The scheme of the feature selection is presented in Figure 1. Features from the formed initial set (including the mentioned correlated pairs products) are being added to the model. The feature that provides the biggest accuracy is added to the model and removed from the consideration list. Then the cycle repeats until either there are no features left in the list or adding each of the remained features does not give any improvement to the model's accuracy.

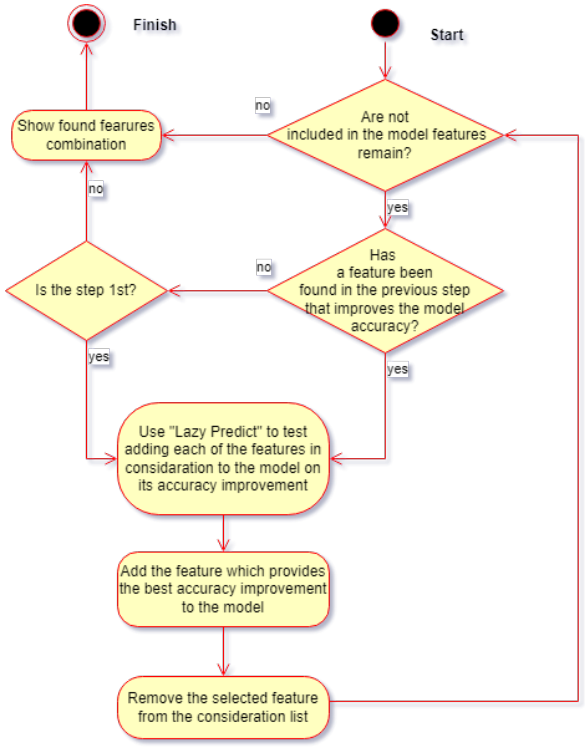



**Figure 1:** Model's features selection scheme.

According to the analysis, 2 separate features from the ULF parameters and 12 their products have been selected for the model.

The given set of features increased the classifier's accuracy on the test data to ~0.78. However, for practical use, higher accuracy is desirable. To achieve this, more parameters were added to the model, specifically products of the ULF factors with other features. Additional experiments were conducted, sequentially adding products of ULF factors with other features to the model, and selecting combinations that most significantly increased the model's accuracy. As a result, 11 feature products have been added to the model.

This supplementation increased the model's accuracy to 0.95 on the test data. The best accuracy using the "Lazy Predict" technique was achieved by KNeighborsClassifier, but RandomForestClassifier also performed well. The obtained accuracy is suitable for developing a production classification service. However, it would be more desirable to make a classifier which can distinguish all 5 classes, but not 2 like this does. Taking into account quite a small dataset it was assumed to develop several binary classifiers each of which could distinguish different classes separation and grounding on their results to make a classification decision.

### 3.2. Development of a Distributed Classifier

Despite KNeighborsClassifier being found to show the best accuracy according to the "Lazy Predict" when classes grouping by the system classes 1 and 2 – group 1, classes 3, 4, and 5 – group 2, tuning of the hyperparameters showed that the Random Forest classifier could exhibit even bigger accuracy in this case – up to 0.97. Furthermore, it can provide rather better accuracy for other classes grouping cases. In Table 1, the values of the classifier hyperparameters provide the best-found accuracy for tasks of certain class groups distinguishing. For the hyperparameters (N estimators, Max depth, and Min samples split) and values optimization, the Powel method was used

and implemented in SciPy. For the classifier that is trained to distinguish Class 5 (clinical cases) from all the others the Decision Tree classifier was used instead of the Random Forest. For the case, it can provide excellent accuracy being a relatively lighter type of model.

Each of the classifiers can return two values (class 1 or class 2) according to the way it was trained. All the possible classifier results combinations have been gathered in a table to analyze the meaning of all the possible options. Such a truth table is presented in Table 2.

**Table 1**
Classifiers hyper parameters value and achieved accuracy

| Classifier No. and Classes groups | Classifier type | Hyper parameters | | | Achieved accuracy |
|---|---|---|---|---|---|
| | | N estimators | Max depth | Min samples split | |
| Classifier 1<br>1 – Level 1<br>2 – Levels 2, 3, 4, 5 | Random Forest | 116 | 40 | 12 | 0.95 |
| Classifier 2<br>1 – Levels 1 and 2<br>2 – Levels 3, 4, 5 | Random Forest | 47 | 7 | 7 | 0.97 |
| Classifier 3<br>1 – Levels 1, 2, 3<br>2 – Levels 4, 5 | Random Forest | 47 | 11 | 5 | 0.93 |
| Classifier 4<br>1 – Levels 1, 2, 3, 4<br>2 – Level 5 | Decision Tree | — | — | 2 | 0.99 |

The provided interpretations include possible controversial cases of the classifier's results. If the resulting combination is (1, 2, 1, 1) it means that the result of Classifier 2 contradicts the one of Classifier 1. Classifier 1 says that this is a light depression level (Class 1) but Classifier 2 says that this may be classes 3, 4 or 5 (harder levels). Classifier 3 does not contradict Classifier 2 – by their results combination this is probably Class 3. Classifier 4 also does not contradict any other, it merely says that this is not Class 5. So this case might be considered as Class 3, but taking into account the result of Classifier 1, which insists that it is a light case, we probably may say of a palliated Class 3.

The next controversial combination is (1, 1, 2, 1). In this case according to the Classifiers 1 and 2 results this may be Class 1 (light depression). However, the Classifiers 3 and 4 results give the Class 4 (severe depression). Because Classifier 4 insists that this is just not Class 5 (a clinical case) but might be also Class 1, as Classifiers 1 and 2 say, this case probably could be considered as Class 1 (but probably somewhat burdened).

The next case is the results combination (2, 1, 2, 1). Here Classifier 2 contradicts Classifier 3. The Classifiers' 1 and 2 results say that this is Class 2 but Classifier 3 insists that this is a much harder case – Class 4 (not 5 due to the Classifier 4). Thus, it may be Class 2 but burdened.

Such a combination as (1, 2, 2, 1) means that the Classifiers 2, 3, and 4 responses result in Class 4 (a severe case). However, Classifier 1 in this case insists on Class 1 (light depression). Therefore, this may be treated as Class 4 but somewhat palliated.

In any case, if Classifier 4 provides value 2 due to its high reliability such situations are to be considered as Class 5 (a clinical case). However, we could additionally provide a range of such cases, but the practical tests have shown that in all the regarded situations other classifiers return the value 2 – combination (2, 2, 2, 2).

**Table 2**
A truth table for the classifiers possible results combinations

| Classifier results combinations | | | | Result interpretation |
|---|---|---|---|---|
| Classifier 1 | Classifier 2 | Classifier 3 | Classifier 4 | |
| Class 1 | Class 1 | Class 1 | Class 1 | Class 1 (light depression) |
| Class 2 | Class 1 | Class 1 | Class 1 | Class 2 (higher depression level) |
| Class 1 | Class 2 | Class 1 | Class 1 | not reliable result, probably Class 3 (middle higher depression level) |
| Class 2 | Class 2 | Class 1 | Class 1 | Class 3 (serious depression level) |
| Class 1 | Class 1 | Class 2 | Class 1 | not reliable result, probably Class 2 (higher depression level) |
| Class 2 | Class 1 | Class 2 | Class 1 | not reliable result, probably Class 2 (higher depression level, may be burdened) |
| Class 1 | Class 2 | Class 2 | Class 1 | not reliable result, probably Class 4 (serious or severe depression level) |
| Class 2 | Class 2 | Class 2 | Class 1 | Class 4 (severe depression level) |
| Class 1 | Class 1 | Class 1 | Class 2 | |
| Class 2 | Class 1 | Class 1 | Class 2 | |
| Class 1 | Class 2 | Class 1 | Class 2 | |
| Class 2 | Class 2 | Class 1 | Class 2 | Class 5 (clinical cases) |
| Class 1 | Class 1 | Class 2 | Class 2 | |
| Class 2 | Class 1 | Class 2 | Class 2 | |
| Class 1 | Class 2 | Class 2 | Class 2 | |
| Class 2 | Class 2 | Class 2 | Class 2 | |

The possible results combinations along with the corresponding interpretations have been organized as a JSON structure used by the developed classification service for the final results provided. For the numerical estimations the controversial cases have been treated as +0.25 to the base obtained level for the burdened cases and as -0.25 for the palliated ones.

### 3.3. Architecture and Working Principles of the Distributed Classifier

The structure of the developed classification service is given in Figure 2. Input messages through the API service come to the initial data filter. It checks if the input massage body has all the necessary fields, are the features values in the allowed range and whether the ULF_100-70, ULF_per_100-70, ULF_70-100, and ULF_70-end are fit the mentioned above restrictions (ULF_100-70 and ULF_per_100-70 must be not zero and in the case if ULF_70-100 > 2000 and ULF_per_100-70 > 300). If the input message has some extra fields they will be merely ignored. If there is a lack of some fields or the values are not sufficient the appropriate message will be returned without any subsequent classification procedures involved. Also, this component of the system orders the input features, received in the form of a dictionary, according to the given input structure which matches the order of the parameters as it was when the classifiers were training. This order is stored in a special file.

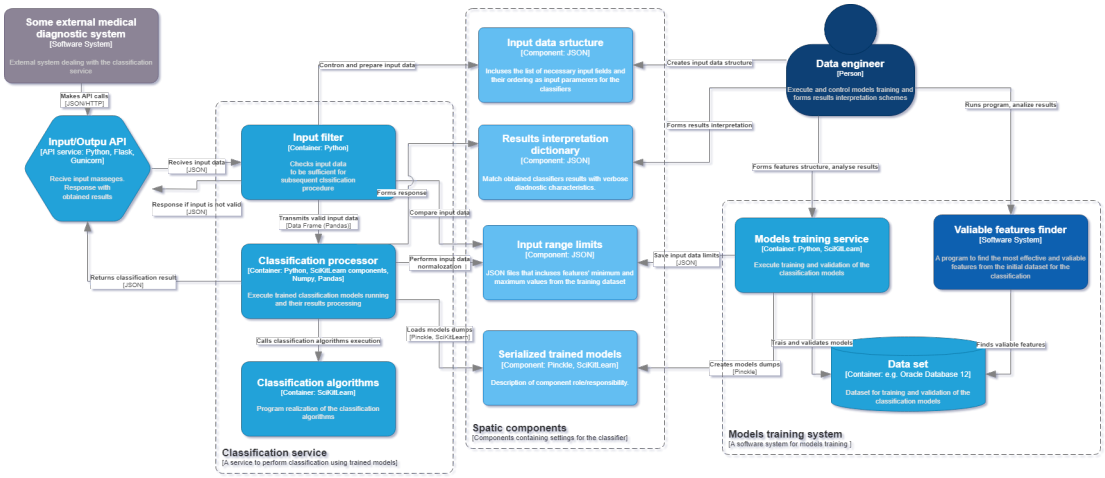



**Figure 2:** The structure of the developed classification service**.**

The structure of the income with a POST request input JSON massage is as follows:

{"passwoed": "passwoed text row",

"values": {

"ULF_total": a float value from 1.25 to 39.7,
"ULF_20": a float value from 0.001 to 460.7,
"ULF_20-70": a float value from 0 to 1381.2,
"ULF_70-100": a float value from 2.2 to 37357.5,
"ULF_100-70": a float value from 0 to 2988.4,
"ULF_70-end": a float value from 0 to 6855.3,
"ULF_per_total": a float value from 2.24 to 16.5,
"ULF_per_20": a float value from 0.001 to 5.2,
"ULF_per_20-70": a float value from 0 to 10.3,
"ULF_per_70-100": a float value from 0.034 to 13.4,
"ULF_per_100-70": a float value from 0 to 16.0,
"ULF_per_70-end": a float value from 0 to 14.2,
"systola_2": a float value from 96 to 217,
"O_IVR_neg": a float value from 11.2 to 254.5,
"O_L2_pos": a float value from 24.9 to 1712.3,
"Hurst_20-70": a float value from 0 to 0.94,
"O_Mo_neg": a float value from 0.5 to 1.3,
"V": a float value from 3.68 to 37.6,
"HFx18x21_20": a float value from 0.32 to 99.5,
"HF_70-100_int_p": a float value from 0 to 29574.7,
"S_Hil_25_60Hz_total": a float value from 0.4 to 21.1,
"S_Hil_HFx13x15_20": a float value from 0.006 to 13.5,
"S_Hil_HFx15x18_20": ": a float value from 0.004 to 13.2 }}

It includes a password field to protect the service from illegal usage. The password is kept in the system in an encrypted form. The main input information is to be given in the “values” section. It is a dictionary of the feature names and values which should be float values in the given ranges. The values’ allowed ranges are determined by the corresponding values ranges in the training dataset and are stored in the form of JSON in a file. These files are generated automatically when the models are training.

If the input data are satisfactory they are transformed into a Pandas data frame and normalized using the stored minimum and maximum values that come to the classification processor. It runs the previously trained classifiers on the given input data. A possibility is provided to run the classifier's execution subsequently or in parallel. The obtained results are compared with the stored

interpretations. Then a resulting message is to be formed and returned through the API. The normal output message has the following JSON structure:

{
"class_code": "number of the estimated depression level class (from 0 to 4)",
"prob": 1 if the result reliable, 0 if the result not reliable,
"result": "verbose description of the result"
}

The value of the "class_code" field is a string. It will be supplemented with a "-" sign (for example "2-") if the case has been interpreted as a palliated one or "+" if it has been regarded as a burdened one. The "prob" will have a 0 value if the classifiers disagree as was denoted in Table 2. In other cases, it will be 1. The "result" field contains a verbose description of the classification result, for example, "Serious depression level (3 from 4). Not clinical.". Here the levels are counted from 1 to 4, and 5 is for the clinical cases.

For the abnormal situations the format of the output messages is different.

- for wrong password:

{
"error": "wrong password",
"message": "access denied"
}

- for software error occurrence:

{
"error": "the error message",
"message": "an error occurred during the message handling"
}

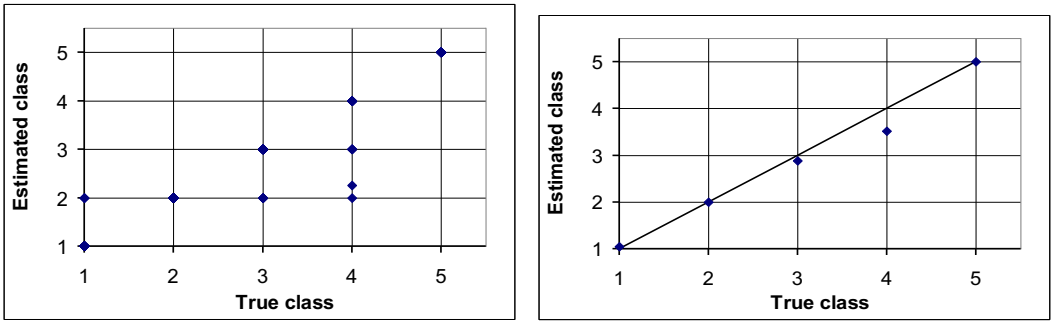
- for wrong input data format:

{
"result": "atypical input data",
"reasons": ["list of explanations regarding the input inconsistence"]
}

Such massages could help to understand easily if something wrong is with the input or the system performance and what exactly is to be corrected.

### 3.4. Testing and the results analysis of the developed classification system

The resulting accuracy of the developed distributed classification system in distinguishing all 5 classes according to the final test on a test sample is obtained at ~0.92. However, the mismatching is different for certain classes. We can't split the results into positive and negative because the classifier is not binary. Nevertheless, we can consider how the classes could be mismatched. Plots in Figure 3 demonstrate such observed mismatch patterns. For the estimated as-burdened values 0.25 was added to the class value and for the palliated 0.25 was subtracted.




a) b)

**Figure 3:** The classes mismatching test results: a) certain values; b) average values.

As we can observe, the mismatching is different for the estimated classes. Except the Class 1 (light depression) the system does not tend to overestimate the depression level. But Class 1 in rare cases may be mismatched with some a harder depression Class 2. However, it also might be addressed to the priory estimation, which could be rather subjective. Nevertheless, it has never been mismatched with the more severe cases in the testing. The Class 2 has always been determined precisely. Class 3 in some cases might be mismatched with the neighboring less severe Class 2. Most of the mismatches have appeared for class 4, which has a smaller number of members. It rather often could be mismatched with less severe classes 2 and 3, but not with 1 or 5. The determination of the Class 5 (clinical cases) seems to be precise.

Actually, in this case, we can't treat results as true negative ones, because it just does not make sense. Nevertheless, as the true positive results we can accept the cases when the estimated class matches the a priori one. If the classifier overestimates the depression level, such a result may be considered a false positive, and if it underestimates the depression level we will treat the result as a false negative one. Therefore, we may say that we have true negative results number TN = 0. Thus, we can calculate the formal criteria. The confusion matrix made on the whole dataset for the considered case is given in Table 3.

According to the obtained results the values of the formal criteria will be as follows:

Accuracy = 0.9172

Recall = 0.9341

Precision = 0.9811

**Table 3**

The confusion matrix

| | Positive | Negative |
|---|---|---|
| True | 156 | 0* |
| False | 3 | 11 |

* - we unable to consider any result as a true negative for the given case

Thereby, we can see that the developed classifier has quite a high precision metric, but a lower recall value. That means that the estimated depression level is probably not lower than it is. However, the obtained evidence of a low depression level is not so reliable. According to the results for Class 4, where 37.5 estimations appear wrong, the model is rather poor in high depression level diagnostic. That is probably caused by a small sample number for this class (10 patients only). Also, we may not discount the possibility of exaggeration of a real depression level in some cases when the dataset was being formed.

The performance benchmark tests of the classification system have been carried out using the following hardware:

HP Probook 445 G7

6 cores CPU – AMD Ryzen 5 4500U with Radeon Graphics

32.0 Gb RAM

The tests have been performed for both options of subsequent and parallel classifiers running. The time gap has been measured from the prepared data sent for the classification and the not-interpreted result obtained. The obtained average numbers are as follows:

- for subsequent executions: 0.1042 seconds

- for parallel executions: 0.1123 seconds

Thus, we can note that in this case running the classifiers in parallel does not make any profit in the performance, moreover, this even slows down the process.

For a better understanding of the selected features' role in the classification results a SHAP test has been carried out on a test 20 % sample. The main test results are shown in a diagram in Figure 4.

The obtained results show that the 5 most valuable input features are:

ULF_70-100
ULF_per_100-70
O_L2_pos
ULF_100-70
ULF_per_total

The role of HF_70-100_int_p and Hurst_20-70 has been found rather insignificant.

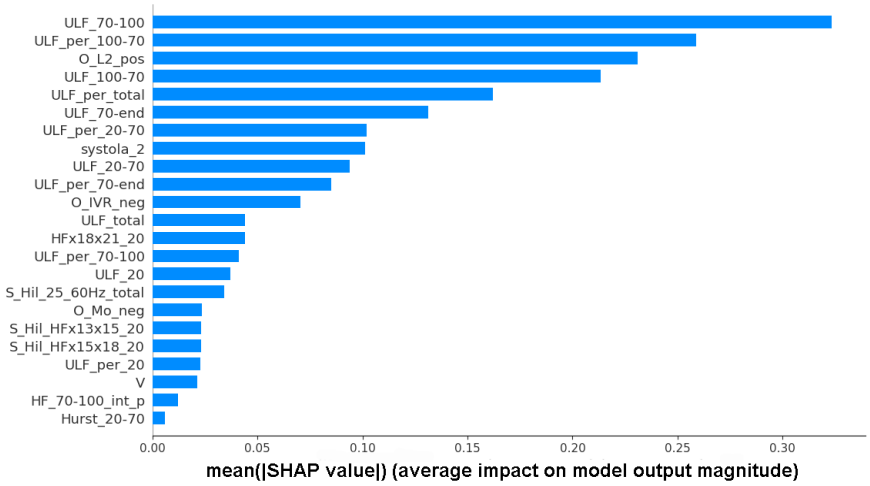



**Figure 4:** SHAP test results on the features impact to the classification

It is worth noting that ULF_70-100, which has been found the most valuable according to the SHAP test, is included in the model as separate parameters, not only as products. That additionally proves their importance for the classification purpose. However, another separate parameter ULF_per_70-100 has rather a mediocre impact.

The proposed classification method demonstrates significant scalability, versatility, and adaptability, making it a valuable tool in modern healthcare. The method's distributed architecture allows for parallel processing and load balancing, enabling it to handle large datasets and complex classification tasks efficiently. The system's modular design further supports scalability by allowing the easy addition of new classifiers and incremental training, which is particularly advantageous in dynamic medical environments where data and conditions evolve over time.

In terms of integration into existing healthcare systems, the classifier can enhance clinical workflows by automating data analysis and providing real-time assessments. This is particularly beneficial in telemedicine platforms, where remote monitoring and diagnosis are critical. The classifier's ability to integrate seamlessly with Electronic Health Record systems also supports the automation of alerts and decision support, reducing the diagnostic workload on healthcare professionals and improving the accuracy of patient assessments. Moreover, the method's generalizability allows it to be adapted to various medical domains beyond the initial application to blood pressure oscillograms. The classifier can be extended to analyze cardiovascular, respiratory, neurological, and other types of medical data, as well as manage chronic conditions like diabetes and cancer. This flexibility makes it a promising tool for personalized medicine, where treatments can be tailored to individual patient needs based on detailed classifications.

Also, it could be noticed, that reusing software development artifacts, such as modules of the proposed here classification system, especially through a domain engineering approach with semantic analysis, is crucial for enhancing the effectiveness of software lifecycle processes. This

approach, as analyzed in case studies, provides valuable insights and recommendations, particularly in the context of information defense systems [17 – 22].

## 4. Conclusions

A classification service was developed, whose main peculiarity is the usage of several binary classification components each of them trained to distinguish its class separation. Grounding on the obtained results combination it is possible to determine several classes using for the model training a relatively small dataset. Another essence of the developed classification approach is usage only separate input feature values, but also products of the feature values whose correlation is significant for a particular data group only, but insignificant or has a different sign for the others. The developed technique has been implemented on the example of the medical dataset on blood pressure oscillograms as indicators of the body's condition, which have been matched with the depression levels.

Despite quite a small dataset the testing results have showed rather high metric values: Accuracy = 0.9172, Recall = 0.9341, and Precision = 0.9811. Thus the developed classification service has a high Precision value, but Recall is relatively lower. That is because it is more likely to tend to underestimate the depression level than to overestimate it. Thus, grounding on the obtained classification results we can assert that the depression level with a rather high probability is not lower than the received value. The most mismatched has appeared the Class 4 which had a very small data sample addressed to it.

According to the SHAP test, the most valuable for the classification are the following features: ULF_70-100, ULF_per_100-70, O_L2_pos, ULF_100-70, ULF_per_total. Most of them relay to the ULF group – the power of ultra-low-frequency (ultra-slow) oscillations (with a frequency of less than 0.003 Hz.)

## Data Availability Statement

The data that support the findings of this study are available from the corresponding author, Kyrylo Malakhov, upon reasonable request.

## Acknowledgements

This study would not have been possible without the financial support of the National Research Foundation of Ukraine (Open Funder Registry: 10.13039/100018227). Our work was funded by Grant contract: Development of the cloud-based platform for patient-centered telerehabilitation of oncology patients with mathematical-related modeling [23 – 25], application ID: 2021.01/0136.

## References

[1] L. Breiman, Random Forests, Machine Learning 45, (2001) 5-32. doi:10.1023/A:1010933404324

[2] T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016 pp. 785-794. doi:10.1145/2939672.2939785

[3] A. Liaw, M. Wiener, Classification and Regression by Random Forest. R News 2, (2002) 18-22. http://CRAN.R-project.org/doc/Rnews/

[4] A. M. Prasad, L. R. Iverson, and A. Liaw, Newer Classification and Regression Tree Techniques: Bagging and Random Forests for Ecological Prediction, Ecosystems 9, (2006) 181-199. doi:10.1890/07-0539.1

[5] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Machine Learning 63, (2006) 3-42. doi:10.1007/s10994-006-6226-1

[6] C. G. Caro, T. J. Pedley, R. C. Schroter, W. A. Seed, K. H. Parker, The Mechanics of the Circulation, 2nd ed. Cambridge University Press, 2011. doi: 10.1017/CBO9781139013406

[7] D. Vakulenko, L. Vakulenko, L. Hryshchuk, L. Sas, Aplication Arterial Oscilography to Study the Adaptive Capacity of Subject with COVID-19 in Primary Care, Primary Health Care, A. Emel Önal, Ed., IntechOpen, 2022. doi: 10.5772/intechopen.98570.

[8] D. Vakulenko, L. Vakulenko, Spectral analysis of arterial pulsations recorded during blood pressure measurement by the Oranta-AO information system, Arterial Oscillography: New Capabilities of the Blood Pressure Monitor with the Oranta-AO Information System, in New Developments in Medical Research. Nova Science Publishers, Inc., 2023. https://novapublishers.com/shop/arterial-oscillography-new-capabilities-of-the-blood-pressure-monitor-with-the-oranta-ao-information-system/

[9] V. P. Martsenyuk, D. V. Vakulenko, L. A. Hryshchuk, L. O. Vakulenko, N. O. Kravets, N. Ya. Klymuk, On the Development of Directed Acyclic Graphs in Differential Diagnostics of Pulmonary Diseases with the Help of Arterial Oscillogram Assessment, Graph-Based Modelling in Science, Technology and Art, S. Zawiślak and J. Rysiński, Eds., in Mechanisms and Machine Science, vol. 107. Cham: Springer International Publishing, 2022, pp. 157–173. doi: 10.1007/978-3-030-76787-7_8.

[10] A. Pokrovsky, Clinical angiology, Medicine, 1979. Available: https://cdn.e-rehab.pp.ua/u/pokrovsky-clinical-angiology-1979.pdf

[11] H. R. Warner, S. H. Swan, D. C. Connolly, Quantitation of beat-to-beat changes in stroke volume from the aortic pulse contour in man, J Appl Physiol. 5, (1953) 495–507. doi: 10.1152/jappl.1953.5.9.495. Available: https://pubmed.ncbi.nlm.nih.gov/13034677/

[12] G. J. Langewouters, K. H. Wesseling, W. J. Goedhard, The static elastic properties of 45 human thoracic and 20 abdominal aortas in vitro and the parameters of a new model, J Biomech 17, (1984) 425–435. doi: 10.1016/0021-9290(84)90034-4

[13] O. Palagin, V. Kaverinskiy, A. Litvin, K. Malakhov, OntoChatGPT information system: Ontologydriven structured prompts for ChatGPT meta-learning, International Journal of Computing 22 (2023) 170–183. doi:10.47839/ijc.22.2.3086.

[14] O. Palagin, V. Kaverinsky, M. Petrenko, K. Malakhov, Digital health systems: Ontology-based universal dialog service for hybrid e-rehabilitation activities support, in: 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), volume 1, 2023, pp. 84–89. doi:10.1109/IDAACS58523.2023.10348639.

[15] V. Kaverinskiy, K. Malakhov, Natural language-driven dialogue systems for support in physical medicine and rehabilitation, South African Computer Journal 35 (2023) 119–126. doi:10.18489/sacj.v35i2.17444.

[16] O. V. Palagin, M. G. Petrenko, Methodological foundations for development, formation and it-support of transdisciplinary research, Journal of Automation and Information Sciences 50 (10) (2018) 1–17. doi:10.1615/JAutomatInfScien.v50.i10.10.

[17] O. Chebanyuk, Investigation of Drawbacks of the Software Development Artifacts Reuse Approaches based on Semantic Analysis, Lecture Notes on Data Engineering and Communications Technologies 181, (2023) 514–523. doi: 10.1007/978-3-031-36118-0_46.

[18] O. Chebanyuk, Formal foundations for software model to model transformation operation, 11th International Scientific and Practical Conference from Programming (UkrPROG'2018), Kyiv, Ukraine, CEUR Workshop Proceedings, 2139, (2018) 124-131. https://ceur-ws.org/Vol-2139/124-131.pdf.

[19] O. Chebanyuk, Software Reuse Approach Based on Review and Analysis of Reuse Risks from Projects Uploaded to GitHub, Computer Science and Education in Computer Science. CSECS 2023, vol 514. Springer, Cham. 144–155. doi: 10.1007/978-3-031-44668-9_11

[20] M. Petrenko, A. Sofiyuk, On One Approach To The Transfer Of An Information Structures Interpreter To Pld-Implementation, Upr. Sist. I Mashiny 188.6 (2003) 48–57.

[21] O. Kurgaev, M. Petrenko, Processor structure design, Cybern. Syst. Anal. 31.4 (1995) 618–625. doi:10.1007/bf02366417.
[22] M. Petrenko, O. Kurgaev, Distinguishing features of design of a modern circuitry type processor, Upr. Sist. I Mashiny 187.5 (2003) 16–19.
[23] K. S. Malakhov, Insight into the digital health system of Ukraine (eHealth): Trends, definitions, standards, and legislative revisions, International Journal of Telerehabilitation 15 (2023) 1–21. doi:10.5195/ijt.2023.6599.
[24] P. I. Stetsyuk, A. Fischer, O. M. Khomiak, Unified Representation of the Classical Ellipsoid Method, Cybern. Syst. Anal. 59.5 (2023) 784–793. doi:10.1007/s10559-023-00614-x.
[25] I. V. Sergienko, V. P. Shylo, V. O. Roshchyn, Algorithm Unions for Solving Discrete Optimization Problems, Cybern. Syst. Anal. 59.5 (2023) 753–762. doi:10.1007/s10559-023-00611-0.