Information Technology and Management Science
ISSN 2255-9094 (online)
2021, vol. 24, pp. 45–52
https://doi.org/10.7250/itms-2021-0007
https://itms-journals.rtu.lv

Variability of Classification Results in Data with High Dimensionality and Small Sample Size

Jana Buša[1*], Inese Poļaka[2]
[1,2] *Riga Technical University, Riga, Latvia*
[2]*Institute of Clinical and Preventive Medicine, University of Latvia, Riga, Latvia*

***Abstract* – The study focuses on the analysis of biological data containing information on the number of genome sequences of intestinal microbiome bacteria before and after antibiotic use. The data have high dimensionality (bacterial taxa) and a small number of records, which is typical of bioinformatics data. Classification models induced on data sets like this usually are not stable and the accuracy metrics have high variance. The aim of the study is to create a preprocessing workflow and a classification model that can perform the most accurate classification of the microbiome into groups before and after the use of antibiotics and lessen the variability of accuracy measures of the classifier. To evaluate the accuracy of the model, measures of the area under the ROC curve and the overall accuracy of the classifier were used. In the experiments, the authors examined how classification results were affected by feature selection and increased size of the data set.**

***Keywords* – Classification algorithms, feature selection, high dimensionality, machine learning.**

I. Introduction

With the development of sequencing technologies and data analysis methods over the past decade, there is a dramatic increase in the number of highly complex datasets due to biological studies quantifying molecular variables such as protein, gene, and microbiome composition [1], [2]. Moreover, these technologies increased the ability to characterise the human microbiome, suggesting its potential use in predicting disease states [3]. A microbiome is a collection of all microbes found in or on the human body, and it plays a key role in maintaining human health [4]. Data collected in studies reflect changes in the microbiome when conditions such as disease, lifestyle, and dietary habits change. Thus, a correlation is formed between certain microorganisms and the specific condition; this information can be used to determine the health status as well as to conduct further studies [5]. However, the analysis of these datasets is complicated and challenging due to their unique structure, such as high dimensionality, low sample size, and excessive zeros [4]. Such datasets are called high-dimension low-sample size (HDLSS) datasets and they consist of a large number of features f, exceeding a small number of samples n, resulting in $f > n$ [6].

Machine learning is used in many fields and is becoming increasingly popular in biotechnology because it can process multidimensional data and has the potential to predict disease and support medical diagnoses [7]. Machine learning uses algorithms based on mathematical rules to determine relationships between variables and discover patterns in data, with the goal of developing a predictive model. The developed models are not only able to predict the health status but also to identify potential pathogens [8]. Many of these algorithms are being used for classification, and the most common ones used for microbiome analysis are Support Vector Machine, Random Forest, k-NN [9], Naïve Bayes [10], and Neural Network [3]. Although machine learning classification algorithms play an important role in finding important relationships in complex biological systems, there are still challenges in applying them to data with high dimensionality and small sample size [6].

The large number of features in HDLSS poses a particular challenge for solving classification problems caused by the presence of many noisy features, leading to severe overfitting [11]. In dimensionality reduction, Feature Selection is a powerful technique for analysing such data by selecting the subset of “relevant” features, thereby reducing the size of the feature space and the risk of overfitting [6]. However, even after reducing the feature space (where $f < n$) there is still a risk of overfitting due to the small sample size, leading to model prediction errors (bias and variance) with high variance even in the presence of low bias [12]. Variance measures the extent to which classifier predictions differ from a training sample to a training sample. There is evidence that the variance should decrease as the size of the dataset increases [12].

This study focuses on the analysis of HDLSS data containing information on the number of genome sequences of bacteria of the gut microbiome before and after antibiotic therapy, using six classification algorithms. The aims of this study are the following:

1. to create a classification model that performs the most accurate microbiome classification before or after antibiotic ingestion;
2. to analyse the changes in the variability of the model predictions after applying Feature Selection and adding more data to the dataset;
3. to suggest areas for further research.

[*] Corresponding author. E-mail: janabusa@outlook.com

II. Literature Review

The data set $D = \{(A_1C_1), \ldots, (A_nC_n)\}$ is given in the classification task, in which the microbiome of each patient n is characterised by the bacteria $A = (A_1, \ldots, A_i)$ it contains and its belonging to a class C: before (C_1) or after (C_2) the use of antibiotics. The aim of this study is to develop a classification model $f(\cdot)$ that can perform the most accurate classification of a patient's gut microbiome into one of the classes C (1).

$$C' = f(A_n), \quad (1)$$

where C' is class prediction.

Since it is recommended to apply and evaluate more than one classification algorithm and select the one with the best performance [9], in this study, six different classification algorithms based on different approaches will be compared:

- Support Vector Machine (SVM);
- Decision Tree (DT);
- Random Forest (RF);
- *k*-Nearest Neighbour (*k*-NN);
- Artificial Neural Network (ANN);
- Naïve Bayes (NB).

Before applying classification algorithms, a dimensionality reduction of the dataset should be performed using the methods of Feature Selection. However, there is no "best" Feature Selection method, so two different methods – Filter Feature Selection and Wrapper method – will be compared to find a suitable method for a particular problem. AUROC and Classification Accuracy are used to evaluate the classifiers.

A. Classification Algorithms

SVM is one of the most popular supervised machine learning algorithms widely used for bioinformatics data classification [9]. The SVM algorithm aims at determining an optimal hyperplane in n-dimensional space (where n is the number of features) that separates the two classes with maximal margin using a minimum number of data points, also called support vectors. An infinite number of hyperplanes could be chosen to separate the data points into two classes, so the goal is to find a plane with the maximum margin, i.e., the largest distance between the data points of the two classes. The idea behind SVM is to choose the hyperplane that provides the best generalization ability. Theoretically, it has been shown that the margin maximization principle gives SVM a good generalization capacity. SVM can be applied upon linear and nonlinear problems. The linear SVM is used when a straight line or hyperplane can divide the data into the classes. The nonlinear SVM is used when the data cannot be partitioned linearly [13]. The slack variable was introduced to relax the margin (soft margin optimization) and kernel functions – to convert nonlinear problems into linear problems [14].

The decision tree is one of the most popular and widely used classification model types. A tree consists of nodes (features), edges (possible values for the feature), and leaves (classes). Decision trees are constructed recursively using a training set. The construction process proceeds from top to bottom, starting with the tree's root node, and at each step selects the feature that best splits the set and creates new nodes in a tree [15]. There are many metrics to find the "best" features, and one of them is Information Gain (IG). IG measures entropy reduction (entropy determines the amount of ambiguity and randomness in a data set) by splitting a data set according to a specific value of a random variable (2). A larger IG indicates groups of samples with lower entropy and, therefore, lower uncertainty.

$$IG(F,D)=E(D) - \sum_{t\in T} p(t)E(t), \quad (2)$$

where F is a feature that splits the data set D into subsets t belonging to a set of subsets T, $E(D)$ is the entropy of the original dataset and $E(t)$ is the entropy of subset t and $p(t)$ is the proportion of the number of elements of data set D belonging to the data subset t. The tree construction process continues until the breakpoint is reached, at which point a new node becomes a leaf and is assigned to a class. New inputs are classified by following a path through the tree, taking the edges that correspond to the values of a feature [14].

The RF algorithm is based on ensemble methods, while its techniques combine different classifiers using an aggregation technique, such as majority voting. This method has many properties, and one of them is that it prevents overfitting. In this case, aggregation of classifiers using the voting technique reduces the variation of the final classifier and ensures good classification performance. As the name implies, the Random Forest consists of many individual decision trees (CART), where each tree performs a class prediction for the input, and the class with the most votes (most predicted) is the final class for the input [16]. The advantage of this algorithm is that it adds additional randomness to the model when building decision trees. Instead of finding the "best" feature that splits the tree (in a decision tree), it searches for the "best" feature among the randomly created subsets of trees, resulting in more diversity, which usually leads to better model performance [17].

The *k*-NN is one of the simplest machine learning algorithms; it is nonparametric and used for classification and regression. Objects are classified based on a distance measure that indicates the distance to each object in the training dataset. The most used distance metrics are correlation coefficients and Euclidean distance. The principle of this method is based on the concept that data objects of the same class should be closer to each other in the feature space. For example, for a data point x of an unknown class, the distance between it and all the data points in the training dataset must be calculated and then x should be assigned to a class based on the k data points closest to x. This method has its drawbacks – the classification results depend heavily on the value of k. If the value of k is too small, the classifier may be sensitive to noise, but for high values of k, the object may be misclassified, so the optimal value of k is often determined experimentally, provided that $k \leq \sqrt{n}$, where n is the number of elements in the training set [18].

An ANN is a mathematical model "inspired" by the structure of biological neural networks. An artificial neural network is an adaptive system that changes its structure based on external or internal information it learns during the training phase. Today, ANN is used for nonlinear data modelling and when continuous features describe the data. They can model complex relationships between input data and the target class/feature and find relationships in the data [19]. A simple ANN model consisting of a single neuron is called a perceptron, while a model with a more complex structure is called a multilayer perceptron. An artificial neuron receives signal x_i and weight w_i and an additional offset signal (bias) whose value is always 1. The output of perceptron Y is determined using (3):

$$Y = f(net), \tag{3}$$

where $f(\cdot)$ is the activation function and *net* is calculated based on input value x and weight w as follows:

$$net = \sum_{i=1}^{n} x_i w_i. \tag{4}$$

Activation functions help the neural network to use important information and reduce the weight of inappropriate data points. There are many activation functions (sigmoid, softmax, ReLu, tanh, etc.), and one of the goals is to train the neurons to solve nonlinear tasks. Perceptron training is performed as follows: the weight vector w is iteratively changed until the output values of the model are equal to the desired values, or the termination criterion is reached. Usually, the mean square error is used as the termination criterion [13]. A multilayer perceptron differs from a perceptron in that it can solve complex tasks that a perceptron cannot. A multilayer perceptron has a much more complex structure and consists of the following layers: an input layer, an output layer, and one or more hidden layers. There are many different techniques for training multilayer perceptrons. One of the best known is the backpropagation algorithm, which moves forward and backward through the network and which is used in this study. In the forward direction, the network is fed an example from the training set to be classified. The backward direction consists of recursively updating the weights of all layers based on the calculated errors [20].

Naïve Bayes is a probabilistic learning algorithm that uses Bayes' theorem in conjunction with the strong assumption that features are conditionally independent given the class. Although this independence assumption is often violated in practice, Naïve Bayes still often provides competitive classification accuracy. In Naïve Bayes, the information in the training data is used to estimate the posterior probability $P(C \mid A)$ for each class C given an object A. Once the estimates are made, they can be used for classification. It selects the most likely classification based on the defined features and determines the most likely class using formula (5).

$$C' = \text{argmax}_c\, P(c_j) \prod_{i=1}^{n} P(a_i|c_j). \tag{5}$$

If the feature values are continuous, they can be discretized, or the Gaussian distribution is often used, with the conditional probability calculated as follows (6):

$$P(a_i|c_j) = \frac{1}{\sigma\sqrt{2\pi}}\, exp^{-\frac{(a_i - \bar{a}_i)^2}{2\sigma^2}}, \tag{6}$$

where σ is standard deviation, $\sigma^2$is variance and $\overline{a}_i$ is the mean. Combined with its computational efficiency and many other desirable properties, this leads to the widespread use of Naïve Bayes in practice [13].

B. Feature Selection

For feature selection, there are methods that directly identify features that do not contribute to or even reduce the accuracy of the predicted model and detect the relevant features. These methods are divided into several approaches, two of them are *Filter Feature Selection* and *Wrapper* methods.

Filter Feature Selection methods use a statistical measure to provide an evaluation for each feature. The features are then ranked by their score, with more important features at the top and less important features at the bottom. There are several filtering methods, but the two most popular are the Information Gain and correlation-based feature (CFS) selection. CFS is a simple method of ranking features according to a heuristic evaluation function based on correlations. The evaluation function tends to subsets where the features have a solid connection to the target features and are uncorrelated with each other. Unsuitable features unrelated to the target features are ignored, and the remaining features are tested for their association with other features. Feature Selection based on Information Gain (2) is one of the most popular methods in which features are ranked by evaluating the Information Gain of each feature with respect to the target feature. The filtering method is advantageous because of its speed and good generalization ability.

In the Wrapper approach, one of the machine learning algorithms is used to evaluate subsets of features according to their predictive performance. The use of learning algorithms is associated with the computational cost of calling machine learning algorithms to evaluate each subset of features; however, this method provides better performance results than the filter selection method. Unfortunately, this method is not suitable for small datasets, as it leads to selective bias and overfitting when filtering methods perform independent Feature Selection [21].

C. Evaluation Metrics

The most commonly used metric for evaluating classifiers is Classification Accuracy (CA). CA is expressed as the number of correctly classified examples divided by the total number of classified examples. This evaluation metric is quite simple, but the results can be misleading when applied to datasets where the classes are unbalanced [13].

The Receiver Operating Characteristics curve (ROC) can be used to understand the core performance of the classifier in separating two classes. The ROC curve is usually represented in True Positive Rate (TPR) (7) and False Positive Rate (FPR) (8), and these terms are defined as follows:

$$TRP = \frac{TP}{TP + FN}, \tag{7}$$

$$FRP = \frac{FP}{FP + TN}. \tag{8}$$

In the graph, the x-axis marks FPR, and the y-axis marks TPR. Initially, both rates are 0 because the threshold is such that all samples are classified into negative class. At the opposite end, all samples are classified into the positive class, and both rates are 1. For each problem, an optimal threshold must be chosen that provides the optimal performance under the given conditions. The ROC curve provides a metric that is considered the most important property of the classifier. This metric is the area under the ROC curve (AUROC). The larger the AUROC, the better the performance of the algorithm. If the value of the AUROC for the model is equal to 1, it means that the model correctly predicted the class for all data points. If the value is 0, then the model predicted the wrong class for all data points. If the value is 0.5, it means that the model was not able to find the differences between the data points and classified them randomly. If the value of the curve is between 0.5 and 1, there is a probability that the model is able to distinguish the values of the positive class from the values of the negative class [13].

III. Methods

In order to achieve the goal of this study, we applied six different supervised machine learning algorithms to the HDLSS bioinformatics dataset to determine which machine learning algorithm creates a classification model that performs the most accurate classification of the microbiome and thus obtains high AUROC and CA values. Since we were dealing with a HDLSS dataset and expected variance in models' predictions, we developed a workflow represented in Fig. 1. with hyperparameter tuning to obtain a robust interpretation of the model performance. We used different methods of Feature Selection as well as added new data to the dataset to see how these approaches affected model performance and variance.

We proposed the following experiments:

1. Model training and evaluation using the initial dataset.
2. Perform Feature Selection based on Information Gain in Orange data mining software and select *N* "best" features.
3. Add more data to the dataset. Perform Feature Selection in Weka data mining software. Compare CFS and Wrapper methods to discover which *N* features lead to better model performance when the Naïve Bayes algorithm is applied.

After each experiment, the updated dataset was loaded into the workflow to evaluate the changes in AUROC and CA values and the variance of the models' prediction. In the end mean values of AUROC and CA of each experiment were compared.

A. Data

Biological data containing information on the number of genome sequences of gut microbiome bacteria before and after antibiotic use were available for analysis. Anonymized data were obtained from the ERDF project No. 1.1.1.1/18/A/184 "Optimisation of H. pylori Eradication Therapy for Population-Based Gastric Cancer Prevention".

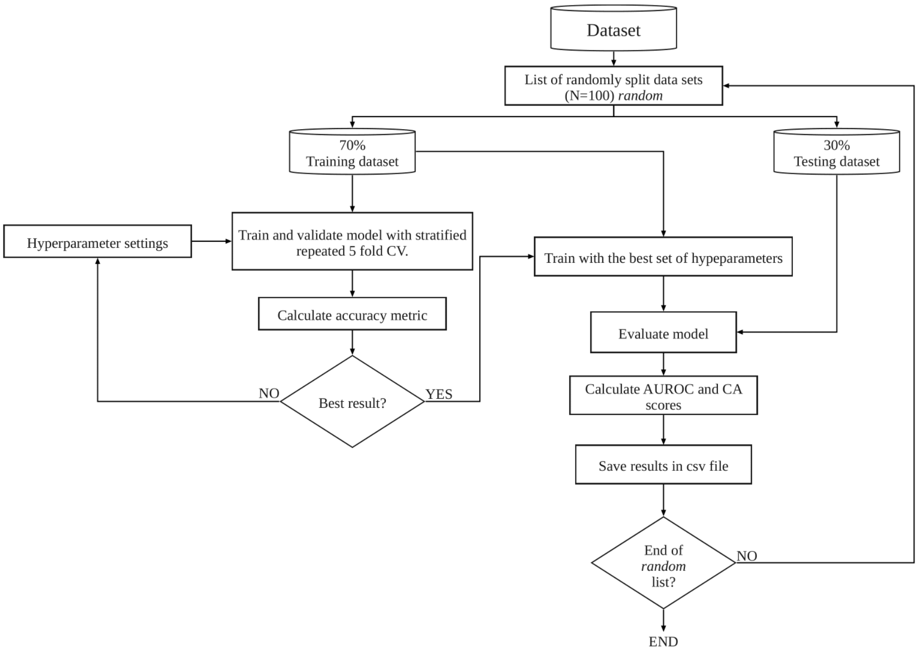
Dataset
List of randomly split data sets (N=100) *random*
70% Training dataset
30% Testing dataset
Hyperparameter settings
Train and validate model with stratified repeated 5 fold CV.
Calculate accuracy metric
Best result?
NO
YES
Train with the best set of hypeparameters
Evaluate model
Calculate AUROC and CA scores
Save results in csv file
End of *random* list?
NO
END

Fig. 1. Developed machine learning workflow. The dataset was randomly split 100 times into a training and a testing dataset, with the training dataset containing 70 % and the testing dataset containing the remaining 30 % of the entire dataset. The stratified split was used to maintain the distribution of classes. Stratified repeated 5-fold cross-validation was used on training dataset to find the best set of hyperparameters, then this set was used to train model on whole training set. For hyperparameter tuning, we used Grid Search for RF, SVM, *k*-NN, Decision Tree and Bayesian Optimization for ANN. The final model was applied to the testing dataset to evaluate its predictive performance on unseen data. Overall, six different machine learning models were trained and evaluated. Workflow was developed in PyCharm Professional v2020.3.3 using Python 3.8 programming language and machine learning packages.

Originally, the dataset consisted of 71 samples and was divided into two groups: C_1 - the *before* class (consisting of 32 samples or 45 % of all samples), and C_2 – the class *after* the use of antibiotics (consisting of 39 samples or 55 % of all samples). Samples were characterized by 263 microbiome bacteria identifiers at order level (features) and their quantity in each sample.

Later, additional data were added to the dataset, and in the end, it consisted of 117 samples, with 52 samples (44 %) belonging to C_1 and 65 (56 %) to C_2, and 271 features.

B. Feature Selection

For Feature Selection, we used Orange [22] and Weka [23] data mining software. In Orange, we performed Feature Selection based on information gain. In Weka, we performed CFS and used wrapper methods to see if we obtained more robust models in comparison. Since different algorithms were compared in this study, we decided to perform Feature Selection using *WrapperSubsetEval* with five similar algorithms – Naive Bayes, Random Forest, Kstar (similar to *k*-NN algorithm, which uses distance metrics), MultiLayerPreceptron, and REPTree (an algorithm that builds a decision tree using information gain). We first decided to do Feature Selection with RF as the classifier, with CA as the evaluation measure, *Best-First* as the search method, and the entire dataset was searched for the "best" feature subset. The same Feature Selection was performed using NB as the classifier. Then we applied the NB algorithm to both reduced datasets to compare the results.

Finally, we decided to perform Feature Selection with each algorithm as the classifier, performing 5-fold cross-validation so that the best features were searched in five subsets. Each time this method returned the number of times (out of a total of five), a particular feature was selected. Next, we evaluated which features appeared most frequently in all results and selected the features that appeared at least five times or more.

IV. Results

In this section, we have collected and compared the results of three experiments and provided approaches to possibly reduce the variance of the results and improve the accuracy of the described methods.

A. First Experiment

We originally intended to perform model training and evaluation on the entire dataset. However, after inspecting the dataset, we found that many bacteria (features) appeared only in a minimal amount in a small number of samples (in total, each sample had about 30 million sequences). Such features do not contribute to the creation of an accurate predictive model and will only increase the computational cost of modelling. Therefore, we decided to keep only the features whose sequences were found in more significant quantities and satisfied one of two conditions:

1. The feature with the mean of 500 or more sequences across all samples.
2. The feature with the mean of at least 100 sequences across all samples and 500 sequences or more in one sample.

After selection, the dimensionality of the dataset was reduced from 263 to 42 features. The models of six methods were trained and evaluated with the reduced dataset. As can be seen in Fig. 2, the prediction results of the models were unstable for all methods and varied between very low and very high results. The highest variance was found in the performance results of ANN models, where the values of AUROC could vary from 0.107 to 0.839 and the values of CA from 0.200 to 0.867. As shown in Table I, the predictive performance of the RF model was higher than the other models, with a mean AUROC value of 0.627 ± 0.019 (95 % CI) and a CA value of 0.629 ± 0.019 (95 % CI).

The highest AUROC value was obtained by the RF model with a value of 0.889 and the lowest AUROC value was obtained by the ANN model with a value of 0.107. The highest CA results were obtained by the ANN models with 0.867, but these method models also obtained the lowest results of 0.200.

B. Second Experiment

In this experiment, we intended to see how the performance results of all six methods would change after Feature Selection. We performed Feature Selection based on Information Gain in Orange mining software. This filter method returned ranked features, where we selected the first twenty features. Six models were trained and evaluated with the selected features.

TABLE I
1ST EXPERIMENT RESULTS

Accuracy metric	Value	*k*-NN	SVM	NB	DT	RF	ANN
AUROC	Mean	0.586 ± 0.009 (95 % CI)	0.576 ± 0.017 (95 % CI)	0.538 ± 0.019 (95 % CI)	0.529 ± 0.020 (95 % CI)	0.627 ± 0.019 (95 % CI)	0.539 ± 0.032 (95 % CI)
	Maximum	0.847	0.750	0.750	0.792	0.889	0.839
	Minimum	0.333	0.236	0.250	0.222	0.375	0.107
CA	Mean	0.615 ± 0.008 (95 % CI)	0.569 ± 0.016 (95 % CI)	0.519 ± 0.018 (95 % CI)	0.542 ± 0.021 (95 % CI)	0.629 ± 0.019 (95 % CI)	0.528 ± 0.025 (95 % CI)
	Maximum	0.810	0.762	0.714	0.857	0.857	0.867
	Minimum	0.333	0.238	0.238	0.286	0.333	0.200

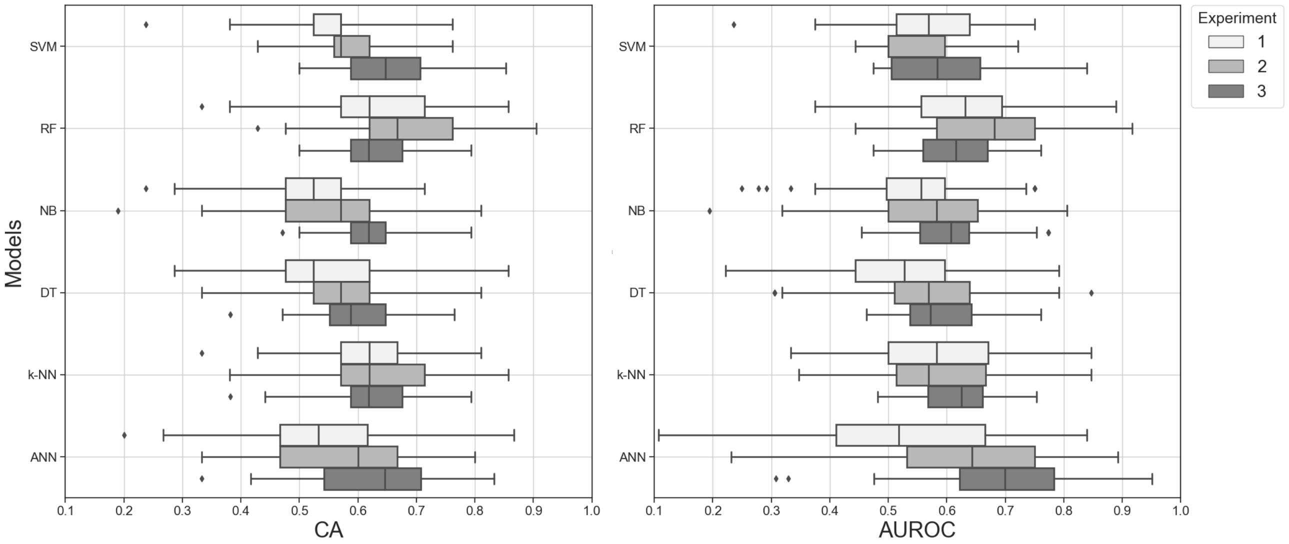

Fig. 2. Performance results of machine learning models on test data using CA and AUROC within three experiments.

The variance of the results of most models decreased slightly, although it remained high (Fig. 2). The variance of the CA results of the *k*-NN models increased, although the variance of the AUROC results somewhat decreased. The variance of the NB results for both AUROC and CA also increased slightly. The variance of the performance results of the ANN models continued to be the highest among the performance results of the other methods, although it decreased. The overall performance of the ANN models improved. As shown in Table II, the prediction performance of the five methods improved. The performance of the RF models was still higher than the other models, with a mean AUROC value of 0.673 ± 0.022 (95 % CI) and a CA value of 0.681 ± 0.021 (95 % CI). Unfortunately, the performance of the SVM models deteriorated for both the AUROC and CA values. The highest AUROC and CA values were obtained by the RF models with values of 0.917 and 0.905, respectively. The lowest AUROC and CA values were obtained by the NB models with values of 0.194 and 0.190, respectively.

C. Third Experiment

In the final experiment, we added more data to the dataset and performed different Feature Selection methods in Weka mining software to choose the one with better NB model performance.

Firstly, we added new data to the dataset, resulting in a dataset with 271 features and 117 samples.

After inspecting the dataset, we faced the same problem as in the first experiment – it consisted mainly of features present in a small number of samples. Similar to the first experiment, we kept only the features that satisfied the two conditions mentioned previously, reducing the dimensionality of the dataset to 42 features. Then, we applied the Wrapper Feature Selection method with RF as the classifier since the models of this machine learning algorithm performed better than the models of other algorithms in previous experiments. After performing Feature Selection, *WrapperSubsetEval* with RF returned 13 features. Then, we applied the NB algorithm to a reduced dataset, since this machine learning algorithm had no hyperparameters to tune, and the results were the following – mean AUROC value of 0.520 ± 0.014 (95 % CI) and mean CA value of 0.537 ± 0.015 (95 % CI). Because the results of NB deteriorated in comparison with the second experiment results, we performed Feature Selection using NB as the classifier, and in the end selected seven features. Once again, we applied the NB algorithm on the reduced dataset, and the results were the following – mean AUROC value of 0.614 ± 0.015 (95 % CI) and mean CA value of 0.634 ± 0.014 (95 % CI). The performance results of the NB models were much better when the features were selected using NB as the classifier in the Wrapper method. Such results indicate that the features selected with *WrapperSubsetEval* could be more adjusted to the machine algorithm used for the Feature Selection. For this reason, we used five machine learning algorithms similar to this study as classifiers and collected all the Feature Selection results to finally select 13 features that occurred most frequently.

Similarly, we applied the NB algorithm on the reduced dataset, and the results were the following – mean AUROC value of 0.530 ± 0.012 (95 % CI) and mean CA value of 0.566 ± 0.014 (95 % CI). The results we got were not better in comparison with the second experiment results. Thus, the last step was to perform the *CorrelationAttributeEval* (CFS) method to evaluate changes in NB model performance once the algorithm was applied to the reduced dataset. This filter method returned ranked features, and we selected the first fifteen features. In the end, the NB model performance was the following – mean AUROC value of 0.604 ± 0.011 (95 % CI) and mean CA value of 0.628 ± 0.013 (95 % CI).

The results were better than the results of the second experiment, and the Wrapper method when it was used with five different algorithms as classifiers. Considering all the results, we decided to train and evaluate models with features selected using the CFS method since there was an improvement in the results, and we were sure that the selected features were not adapted to a specific machine learning algorithm.

TABLE II
2ND EXPERIMENT RESULTS

Accuracy metric	Value	*k*-NN	SVM	NB	DT	RF	ANN
AUROC	Mean	0.586 ± 0.019 (95 % CI)	0.545 ± 0.014 (95 % CI)	0.580 ± 0.020 (95 % CI)	0.567 ± 0.022 (95 % CI)	0.673 ± 0.022 (95 % CI)	0.623 ± 0.029 (95 % CI)
	Maximum	0.847	0.722	0.806	0.847	0.917	0.893
	Minimum	0.347	0.444	0.194	0.306	0.444	0.232
CA	Mean	0.621 ± 0.018 (95 % CI)	0.580 ± 0.013 (95 % CI)	0.566 ± 0.020 (95 % CI)	0.567 ± 0.021 (95 % CI)	0.681 ± 0.021 (95 % CI)	0.579 ± 0.022 (95 % CI)
	Maximum	0.857	0.762	0.810	0.810	0.905	0.800
	Minimum	0.381	0.429	0.190	0.333	0.429	0.333

In Fig. 2, it is visible that after adding additional data and performing CFS, the variance of the performance results of almost all models decreased noticeably, except for the results of the SVM models. Their performance results had the highest variance of AUROC, and CA results compared to previous experiments. As shown in Table III, most of the performance results improved. The ANN models achieved the best mean AUROC value among all experiments – 0.698 ± 0.024 (95 % CI). Nevertheless, the variance of the performance results of the ANN models continued to be the highest. The variance of AUROC results decreased, but the variance of CA scores increased. Interestingly, the performance of the RF models decreased in both AUROC and CA values. Other machine learning models improved their predictive performance. The highest AUROC results as well as the lowest results were obtained by the ANN models. The highest CA result was obtained by the SVM model with a value of 0.853 and the lowest was obtained by the ANN model with a value of 0.333.

V. CONCLUSION

This study aimed at developing a classification model that would classify the microbiome into two groups as accurately as possible and reduce variability in the classifier accuracy measures.

It cannot be clearly stated that a classifier induced using a particular algorithm made more accurate predictions than other algorithms. In the first two experiments, the classifiers of RF algorithm showed better performance than other predictive machine learning models, but after adding more samples to the dataset and performing Feature Selection with a different method, the classification results of RF deteriorated significantly.

The classification results of the ANN classifiers improved with each experiment, and in the last experiment, the mean value of AUROC was the highest among all classifiers and experiments yet it also had the most significant variance in accuracy measures.

After performing the experiments, it has been concluded:

1. Feature Selection is an essential step in the analysis of HDLSS data. The results have shown that it has an impact on the model performance and leads to better results on the accuracy measures of the classifier. Performing Feature Selection for data with a small sample size using the Wrapper method, the selected features can be adapted to a specific machine learning algorithm, so it is better to use an algorithm-independent selection method such as the Filter Selection method for result comparison.
2. The variance of classifiers' predictions decreases as the size of the dataset increases.

This can be explained by the fact that by removing data that are not significant for classification, we remove noise and therefore improve the quality of the information held by the dataset. This step can lead to overfitting more quickly by selecting the information (features) significant for the specific dataset, but it can be addressed with larger data sets and/or an appropriate iterative data sampling procedure to test the classification process on different training and test sets.

For further research, we suggest adding more data to the dataset if possible and try different Feature Selection methods to obtain more robust classifiers and reduce the variance of the accuracy measures.

ACKNOWLEDGEMENTS

We would like to thank the Institute of Clinical and Preventative Medicine for giving the samples.

The samples were sequenced within the ERDF project No. 1.1.1.1/18/A/184 "Optimization of H. pylori Eradication Therapy for Population-Based Gastric Cancer Prevention".

TABLE III
3RD Experiment Results

Accuracy metric	Value	*k*-NN	SVM	NB	DT	RF	ANN
AUROC	Mean	0.620 ± 0.013 (95% CI)	0.594 ± 0.017 (95% CI)	0.604 ± 0.015 (95% CI)	0.590 ± 0.014 (95% CI)	0.617 ± 0.014 (95% CI)	0.698 ± 0.024 (95% CI)
	Maximum	0.754	0.840	0.774	0.761	0.761	0.951
	Minimum	0.482	0.474	0.454	0.463	0.475	0.308
CA	Mean	0.629 ± 0.015 (95% CI)	0.651 ± 0.014 (95% CI)	0.628 ± 0.013 (95% CI)	0.593 ± 0.014 (95% CI)	0.628 ± 0.013 (95% CI)	0.637 ± 0.019 (95% CI)
	Maximum	0.794	0.853	0.794	0.765	0.794	0.833
	Minimum	0.382	0.500	0.471	0.382	0.500	0.333

References

[1] D. M. Camacho, K. M. Collins, R. K. Powers, J. C. Costello, and J. J. Collins, "Next-generation machine learning for biological networks," *Cell,* vol. 173, no. 7, pp. 1581–1592, June 2018. https://doi.org/10.1016/j.cell.2018.05.015

[2] X.-B. Qian *et al.*, "A guide to human microbiome research: study design, sample collection, and bioinformatics analysis," *Chinese Medical Journal,* vol. 133, no. 15, pp. 1844–1855, June 2020. https://doi.org/10.1097/CM9.0000000000000871

[3] M. Oh and L. Zhang, "DeepMicro: deep representation learning for disease prediction based on microbiome data," *Sci. Rep.,* vol. 10, no. 1, p. 6026, Apr. 2020. https://doi.org/10.1038/s41598-020-63159-5

[4] H. Li and H. Li, "Introduction to special issue on statistics in microbiome and metagenomics," *Statistics in Biosciences,* vol. 13, no. 2, pp. 197–199, Mar. 2021. https://doi.org/10.1007/s12561-021-09307-5

[5] C. F. A. Ribeiro, G. Silveira, E. S. Candido, M. H. Cardoso, C. M. Espinola Carvalho, and O. L. Franco, "Effects of antibiotic treatment on gut microbiota and how to overcome its negative impacts on human health," *ACS Infect. Dis.,* vol. 6, no. 10, pp. 2544–2559, Jul. 2020. https://doi.org/10.1021/acsinfecdis.0c00036

[6] A. Golugula, G. Lee, and A. Madabhushi, "Evaluating feature selection strategies for high dimensional, small sample size datasets," in *2011 Annu. Int. Conf. of the IEEE Eng. in Med. and Biol. Soc*., Aug. 2011, pp. 949–952. https://doi.org/10.1109/IEMBS.2011.6090214

[7] S. Bang, D. Yoo, S.-J. Kim, S. Jhang, S. Cho, and H. Kim, "Establishment and evaluation of prediction model for multiple disease classification based on gut microbial data," *Scientific Reports,* vol. 9, no. 1, Jul. 2019, Art. no. 10189. https://doi.org/10.1038/s41598-019-46249-x

[8] B. D. Topcuoglu, N. A. Lesniak, M. Ruffin, J. Wiens, and P. D. Schloss, "A framework for effective application of machine learning to microbiome-based classification problems," *mBio,* vol. 11, no. 3, Jun. 2020. https://doi.org/10.1128/mBio.00434-20

[9] L. J. Marcos-Zambrano *et al.*, "Applications of machine learning in human microbiome studies: A review on feature selection, biomarker identification, disease prediction and treatment," *Frontiers in Microbiology,* Review vol. 12, no. 313, Feb. 2021. https://doi.org/10.3389/fmicb.2021.634511

[10] M. Ziemski, T. Wisanwanichthan, N. A. Bokulich, and B. D. Kaehler, "Beating naive Bayes at taxonomic classification of 16S rRNA gene sequences," *Front. Microbiol.,* vol. 12, p. 644487, Jun. 2021. https://doi.org/10.3389/fmicb.2021.644487

[11] A. Vabalas, E. Gowen, E. Poliakoff, and A. J. Casson, "Machine learning algorithm validation with a limited sample size," *PLoS One,* vol. 14, no. 11, p. e0224365, Nov. 2019. https://doi.org/10.1371/journal.pone.0224365

[12] D. Brain and G. Webb, "On the effect of data set size on bias and variance in classification learning," *Proceedings of the Fourth Australian Knowledge Acquisition Workshop,* Jun. 2000, pp. 117–128.

[13] A. V. Joshi, *Machine Learning and Artificial Intelligence*. Switzerland: Springer, Cham, 2020. https://doi.org/10.1007/978-3-030-26622-6

[14] C. Sammut and I. G. Webb, *Encyclopedia of Machine Learning and Data Mining*. New York: Springer Nature, 2017. https://doi.org/10.1007/978-1-4899-7687-1

[15] H. Zhou, "Decision trees," in *Learn Data Mining Through Excel: A Step-by-Step Approach for Understanding Machine Learning Methods*. Berkeley, CA: Apress, 2020, pp. 125–148. https://doi.org/10.1007/978-1-4842-5982-5_9

[16] L. Igual and S. Seguí, *Introduction to Data Science. A Python Approach to Concepts, Techniques and Applications* (Undergraduate Topics in Computer Science). Switzerland: Springer, 2017. https://doi.org/10.1007/978-3-319-50017-1

[17] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning* (Springer Texts in Statistic). New York: Springer-Verlag, 2013. https://doi.org/10.1007/978-1-4614-7138-7_1

[18] H. Rajaguru and S. K. Prabhakar, "kNN Classifier," in *KNN Classifier and K-Means Clustering for Robust Classification of Epilepsy From EEG Signals. A Detailed Analysis*. Hamburg: Anchor Academic Publishing, 2017, ch. 3, pp. 31–38.

[19] K. Ashley, "Neural networks," in *Applied Machine Learning for Health and Fitness: A Practical Guide to Machine Learning with Deep Vision, Sensors and IoT*. Berkeley, CA: Apress, 2020, pp. 73–91.

[20] A. Meyer-Baese and V. Schmid, "Foundations of neural networks," in *Pattern Recognition and Signal Analysis in Medical Imaging*, A. Meyer-Baese and V. Schmid, Eds. Oxford: Academic Press, 2014, pp. 197–243.

[21] V. Bolón-Canedo and A. Alonso-Betanzos, "Feature selection," in *Recent Advances in Ensembles for Feature Selection*, vol. 147. Cham: Springer International Publishing, 2018, pp. 13–37. https://doi.org/10.1007/978-3-319-90080-3_2

[22] J. Demšar *et al.*, "Orange: data mining toolbox in Python," *Journal of Machine Learning Research,* vol. 14, pp. 2349–2353, 2013. [Online]. Available: http://jmlr.org/papers/v14/demsar13a.html

[23] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., 2016.

Jana Buša received a Bachelor's degree in Information Technology from Riga Technical University in 2021. After graduation she is continuing her education at the University of Latvia as a Master degree student in bioinformatics. Her research interests include data analysis, data mining, supervised and unsupervised machine learning methods.
E-mail: janabusa@outlook.com

Inese Poļaka received her Doctoral degree (Dr. sc. ing.) in Information Technology from Riga Technical University in 2014. She has since worked at Riga Technical University and holds the position of Associate Professor. Her fields of interest are machine learning, data mining, especially in medical applications, evolutionary algorithms, transparent and interpretable supervised and unsupervised learning methods, as well as bioinformatics (main focus on prokaryotic metataxonomics and metagenomics).
E-mail: inese.polaka@rtu.lv
ORCID iD: https://orcid.org/0000-0002-9892-7765