BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 69(1), 2021, Article number: e136044
DOI: 10.24425/bpasts.2021.136044

Ensemble selection in one-versus-one scheme – case study for cutting tools classification

Izabela ROJEK[1*], Robert BURDUK[2], and Paulina HEDA[2]

[1] Institute of Computer Science, Kazimierz Wielki University, ul. Chodkiewicza 30, 85-064 Bydgoszcz, Poland
[2] Faculty of Electronic, Wroclaw University of Science and Technology, ul. Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland

Abstract. The binary classifiers are appropriate for classification problems with two class labels. For multi-class problems, decomposition techniques, like one-vs-one strategy, are used because they allow the use of binary classifiers. The ensemble selection, on the other hand, is one of the most studied topics in multiple classifier systems because a selected subset of base classifiers may perform better than the whole set of base classifiers. Thus, we propose a novel concept of the dynamic ensemble selection based on values of the score function used in the one-vs-one decomposition scheme. The proposed algorithm has been verified on a real dataset regarding the classification of cutting tools. The proposed approach is compared with the static ensemble selection method based on the integration of base classifiers in geometric space, which also uses the one-vs-one decomposition scheme. In addition, other base classification algorithms are used to compare results in the conducted experiments. The obtained results demonstrate the effectiveness of our approach.

Key words: ensemble of classifiers; ensemble selection; one-vs-one decomposition; cutting tool.

1. Introduction

Machine learning (ML) is a subset of Artificial Intelligence techniques and can be defined as the ability to learn through the use of training data [1]. There is a variety of ML methods which depend, among other things, on what data we have [2]. For a supervised classification method, the training set must have a class label. The training set therefore contains the values of the features (feature vector) of the described objects together with the class label of each object (class label vector). Therefore, the goal of supervised classification is to build a mathematical model of a real problem using a labeled dataset. This mathematical model is used to map feature space into class label space in the case of a new object which, in general, does not belong to the training set.

The ensemble of classifiers methods (EoC) is a popular approach for improving the possibilities of individual ML algorithms (base learners, base classifiers) by building more stable and accurate classifiers [3, 4]. In general, the procedure for creating an EoC can be divided into three major steps: generation, selection, and integration.

During the selection phase, either a model (classifier selection), or a subset of all classifiers (ensemble selection) learned in the generation phase, is selected to make the final decision of EoC [5]. The taxonomy of the selection methods distinguishes between the static and dynamic selections. The static pruning process selects one classifier or a certain subset of base classifiers that is invariable throughout all feature space or defined feature subspaces. In the case of the dynamic selection, knowledge about the neighborhood of the newly classified object is used (most often defined by a fixed number of nearest neighbors) to determine one or a certain subset of base classifiers for the classification of a new object.

The discussed topic is still up-to-date, as evidenced by the propositions of new ensemble selection methods [6–9]. The ensemble selection can also be considered in the context of the decomposition of a multiclass problem. The competence of each base classifier in an ensemble and the weighted distance of a new object from other objects in its nearest neighborhood are used in the dynamic ensemble selection, which uses the one-vs-one strategy [10]. The use of the nearest neighborhood of a newly classified object to choose the non-competent base classifiers was presented in [11].

ML algorithms are used in many areas for practical tasks. For example, in technical problems: an artificial neural network [12–14] fuzzy logic [15] and other ML methods [16–18] are used. The supervised classification is used for computer diagnostics [19]; decision trees method was used to ecodesign of technological processes [20]; neural networks with radial basis function, Kohonen networks and the Random Forest classifier were used to optimally select compatible materials [21], in medical tasks to computational gait analysis for post-stroke rehabilitation purposes using fuzzy numbers, fractal dimension and neural networks [22].

On the other hand, there is a great interest in using ML methods in the area of cutting tool selection, or the machining parameters for specific cutting tools. Igari et al. [23] presented an optimum selection model for processing tools and parameters based on decision rules generated by decision trees. Another cutting tool application presents the carbide-tool selection expert system for the purposes of a CNC lathe [24].

*e-mail: izarojek@ukw.edu.pl

Manuscript submitted 2020-07-17, revised 2020-11-05, initially accepted for publication 2020-12-01, published in February 2021

The aim of this study was to develop an optimum system for selecting a tool chuck, a cutting tool, and a plate, along with their machining parameters (i.e. feed and cutting speeds) by using decision rules. Some earlier authors' articles also show the selection of tools using simple classifiers in the form of neural networks [25] and the dynamic ensemble selection that uses median and quartile score function of correctly classified objects [26].

In this work we propose a novel algorithm of the dynamic ensemble that works in the one-vs-one scheme. Accordingly, we propose that the new method for the dynamic ensemble selection takes into consideration values of the score function defined by base classifiers. Therefore, our approach provides information about the relative distance of the recognized object to decision boundary defined by base classifier.

Given the above, the main objectives of this work can be summarized as follows:

- A proposal of a new dynamic ensemble selection algorithm using the one-vs-one decomposition scheme that uses selected classifiers whose decision boundary is close to the recognized object.
- Experimental research to compare the proposed method with static ensemble selection based on integration of base classifiers in geometric space, which also used the one-vs-one decomposition scheme and base classifiers on the real classification problem regarding the classification of cutting tools.

The paper is structured as follows: In the next section, the background of classification and decomposition of multi-class classification problems are presented. The proposed algorithms are presented in Section 2. The experiments that were carried out are presented in Section 4, whereas the results and the discussion are presented in Section 5. Finally, we draw conclusions and propose future works in Section 6.

2. Background of classification

2.1. Ensemble of classifiers. The recognition algorithm Ψ maps the feature space X to the set of class labels $\Omega = \{\omega_1, \omega_2, \ldots, \omega_M\}$ (M denotes the number of class labels) according to the general formula:

$$\Psi : X \to \Omega. \tag{1}$$

Therefore, the classification goal is to assign a given object $x \in X$ into one of the predefined class labels $\omega_i \in \Omega$.

The idea of EoC methodology is to build a predictive model by integrating multiple base classification models $\Psi_1, \Psi_2, \ldots, \Psi_K$, where K is the number of classifiers in the EoC. The procedure for creating an EoC can be divided into three major steps [27]:

- Generation – a phase where individual classifiers are trained [28].
- Selection – a phase where only a few (or even one) individual models from the previous step are selected for inclusion in the EoC [5].
- Combining or integration – a process of combining outputs of base classifiers to obtain an integrated model of classification [29].

This article focuses on the problem of ensemble selection.

2.2. One-vs-one decomposition. The most commonly used ML methods are designed to deal with binary classification problems, and their extension to a multi-class task are still unknown [30]. Such an example is the SVM algorithm, which is often used in experimental research. The decomposition strategies [31], like one-vs-one (OVO) and one-vs-all (OVA), are the most common and useful strategies for binarization in the multi-class problem.

In this paper, we focus our attention on the OVO scheme. The OVO scheme divides the original M-class problem into $\frac{M(M-1)}{1}$ binary tasks, i.e. all possible class label pairs (ω_i, ω_j) can be formed from the set of all class labels M. Afterwards, each problem defined by possible class label pairs (ω_i, ω_j) is considered as a binary classification problem. The output of each binary classifier defines its scoring function r_{ij} and $r_{ij} = 1 - r_{ij}$, when r_{ij} is the confidence of the binary classifier learned on dataset containing class label pairs (ω_i, ω_j) in favor of ω_i discriminating the class label ω_i from the class label ω_j. The outputs of all binary classifiers in the OVO scheme are represented by score matrix R:

$$R = \begin{pmatrix} - & r_{12} & \cdots & r_{1M} \\ r_{21} & - & \cdots & r_{2M} \\ \vdots & - & & \vdots \\ r_{1M} & r_{2M} \cdots & - & \end{pmatrix}. \tag{2}$$

Once all the pairs of classifiers are used to construct the score matrix, any of the aggregation methods for the OVO scheme can be used to define the final decision of the EoC.

2.3. Aggregation methods for OVO scheme. The final classification decision of the OVO scheme is derived from the score matrix R using the aggregation method [32]. In this paper, we employed two most popular aggregation methods for the OVO scheme:

- Voting strategy – this aggregation strategy uses the vote of each binary classifier and the class label with the largest number of votes is the final decision in the OVO scheme. If binary classifiers return the score function, then the scoring functions are transposed into votes of these classifiers. The final class label for the voting strategy is obtained as follows:

$$\Psi^V_{OVO} = \arg \max_{i=1,\ldots,M} \sum_{1 \le j \ne i \le M} sf_{ij}, \tag{3}$$

where

$$sf_{ij} = \begin{cases} 1, & \text{if } r_{ij} > r_{ji} \\ 0, & \text{otherwise.} \end{cases}$$

- Weighted voting strategy – this aggregation directly uses the score functions from the score matrix R. The class label with the greatest total values of the score functions is designated as the final class label according to the formula:

$$\Psi^{WV}_{OVO} = \arg \max_{i=1,\dots,M} \sum_{1 \le j \ne i \le M} r_{ij}. \tag{4}$$

Other combination methods are presented, inter alia, in [32]. The choice of the combination method is either arbitrary or depends on the available output type of binary classifiers. For example, if the output of the binary classifier is a class label, we can only use the voting method. We will deal with this case in the static ensemble selection considered in this article.

2.4. Ensemble selection. In the selection phase of the EoC building procedure, one classifier (the classifier selection) or a certain subset of classifiers is selected (the ensemble selection or ensemble pruning) and learned at the generation stage. The classifier selection is a special type of ensemble selection, because only one base classifier is selected from all sets of base classifiers. In general, two different categories of the ensemble selection can be distinguished. The static ensemble selection – in this case, the entire feature space after training the base classifier is divided into disjoint regions of competence. These regions are determined permanently and depend mostly on the value of classifiers' performance measure in these regions. If the newly recognized object belongs to the region of competence, then the selected ensemble of classifiers decides its class label. The dynamic ensemble selection – in this case the competence of base classifiers is determined using the nearest neighborhood of the recognized object. The base classifiers which are the most competent in the nearest neighborhood of this object decide on the class label.

In this article we will the use both the static and dynamic ensemble selection strategies. In the case of the static ensemble selection, the method presented in [33] is adapted to the multiclass problem. In the case of the dynamic ensemble selection, we propose a new approach that takes into account the distance from the base classifier decision boundary to the classified object in the selection process.

3. Ensemble selection in one-vs-one scheme

3.1. The proposed method of dynamic ensemble selection. The distance of the recognized object from the decision boundary is often used to determine the scoring functions or other functions in supervised classification. In linear Support Vector Machines (SVM) algorithms, the distance to the separating decision boundary is used to compute the scoring function. Afterwards, the calibration converts the score functions into a probability measure, or more precisely transforms classifier outputs into values that can be interpreted as probabilities. In this paper, we propose a dynamic ensemble selection that uses score function values directly. Unlike other methods of the dynamic ensemble selection, the proposed approach takes into account the location of the recognized object relative to the decision boundaries of the base classifiers rather than the location of this object relative to other objects from the training dataset. The selection process takes into account the value of the scoring function, which is the output of each classifier from the set of base classifiers obtained after the learning process. The overall procedure of the proposed dynamic ensemble selection in the OVO scheme is shown in Algorithm 1. Figure 2 shows the layered diagram of the OVO method with the selection process that is made in the penultimate layer, i.e. before determining the matrix defined by Eq. (2).

Algorithm 1: Dynamic ensemble selection based on values of score function in the one-vs-one scheme

---

Data: Set of base classifiers $\Psi_1,\dots,\Psi_K$, parameters α and β of the algorithm, recognized object x, training dataset D_{tr}
Result: Ensemble decision after dynamic ensemble selection Ψ_{DESOVO} in the one-vs-one scheme for two aggregations strategy – voting Ψ^{V}_{DESOVO} and weighted voting Ψ^{WV}_{DESOVO}
for *each class labels pair* ω_i, ω_j **do**
  $D^{ij}_{tr} \leftarrow$ a subset of D_{tr} whose class labels are ω_i or ω_j train all classifiers $\Psi^{ij}_k \leftarrow D^{ij}_{tr}$ **for** *each trained classifier* Ψ^{ij}_k **do**
    if $\alpha \le \Psi^{ij}_k(x) \le \beta$ **then**
      $\Psi^{ij}_k(x) = 0$
    end
  end
  build an ensemble of classifiers $Eo\Psi^{ij}_k$ whose score functions are greater than 0
  if $\mid Eo\Psi^{ij}_k \mid = 0$ **then**
    randomly select a classifier from Ψ^{ij}_k to $Eo\Psi^{ij}_k$ $\mid Eo\Psi^{ij}_k \mid = 1$
  end
  $r_{ij} \leftarrow \frac{\sum_{k=1}^{K} \Psi^{ij}_k(x)}{|Eo\Psi^{ij}_k|}$ $r_{ji} \leftarrow 1 - r_{ij}$
end
obtain the final decision of Ψ_{DESOVO}, according to: voting strategy – Ψ^{V}_{DESOVO} weighted voting strategy – Ψ^{WV}_{DSOVO}

3.2. Static ensemble selection. In the static ensemble selection in the OVO scheme, we will adopt the algorithm described in the paper [33]. In the integration phase of building the EoC, this algorithm uses functions that define the decision boundaries of individual base classifiers. The decision boundary of the EoC is therefore built using the decision boundaries of the base classifiers, which have gone through a selection process. The overall procedure of the static ensemble selection in the OVO scheme is shown in Algorithm 2. The visualization of the selection process of the base classifier is presented in Fig. 4.

Algorithm 2: Static ensemble selection based on integration base classifiers in geometric space in the one-vs-one scheme

Data: Set of base classifiers $\Psi_1, \ldots, \Psi_K$, recognized object x, training dataset D_{tr}
Result: Ensemble decision after static ensemble selection Ψ^V_{SESOVO} in the one-vs-one scheme
for *each class labels pair* ω_i, ω_j **do**
  $D^{ij}_{tr} \leftarrow$ a subset of D_{tr} whose class labels are ω_i or ω_j
  select two most important features $D^{ij}_{trf2} \leftarrow D^{ij}_{tr}$
  train all classifiers $\Psi^{ij}_k \leftarrow D^{ij}_{trf2}$
  divide the feature space of D^{ij}_{trf2} in different separable decision regions
  evaluate the base classifiers' Ψ^{ij}_k competence in each decision region based on the classification accuracy
  select 2 best classifiers from all base classifiers for each decision regions
  define the decision boundary of the proposed ensemble of classifiers $Eo\Psi^{ij}_{SES}$ as an average decision boundary of the selected in the previous step classifiers **if** $Eo\Psi^{ij}_{SES}(x) = \omega_i$ **then**
    $r_{ij} \leftarrow 1$ $r_{ji} \leftarrow 0$ **else**
      $r_{ij} \leftarrow 0$
      $r_{ji} \leftarrow 1$
    end
  end
end
obtain the final decision of Ψ_{SESOVO}, according to voting strategy – Ψ^V_{SESOVO}

4. Experimental setup

4.1. Dataset of cutting tools. Examples of cutting tool selection were prepared by technologists in a real production company. The scope of data gathering depends on the type of production. The production of the enterprise was mostly unit, necessitating to the production of a large number of product variants, thus requiring a technologist with a lot of experience when choosing tools. The diversity of these variants leads to a rather low degree of standardization. The right selection of tools is very difficult, because it depends on many earlier selections (selected semi-finished products, technological operations, and machine tool) [34]. This article presents classification models for the selection of cutting tools for milling operations.

The cutting tools dataset contains 564 learning objects ($N = 564$) and 17 class labels ($M = 17$). The features of the object are: machining surface – x_1, symbol of material – x_2, demanded surface roughness – x_3, structure of milling tool – x_4, milling tool clamping – x_5, dimension – x_6, shape of milling cutter – x_7, number of teeth – x_8, milling tool total length – x_9, min. cutting speed – x_{10}, max. cutting speed – x_{11}, cutting depth – x_{12}, milling width – x_{13}, cutting feed – x_{14}, operating cost – x_{15}. The class labels are the milling tool symbols. Table 1 presents examples objects of cutting tools dataset. The examples of cutting tools are presented in Fig. 1.

(a)
(b)

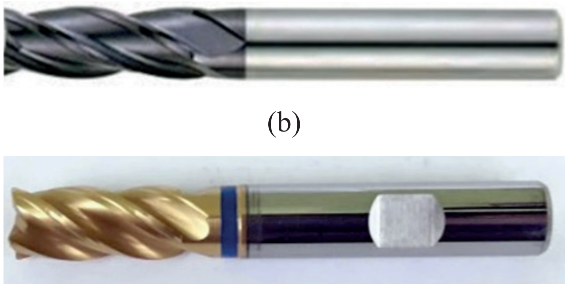

Fig. 1. Examples of cutting tools: a) The cutting tool labeled as Fi16W, b) The cutting tool labeled as 203014_25

The majority of information obtained from the databases was incomplete and raw. To make such data useful for ML purposes, they need to be pre-processed, i.e. transformed and cleaned. Data cleaning entails the identification of extreme points, the supplementation of missing entries, or the unification of records. In turn, data transformation involves coding or normalization.

4.2. Classification performance measures. To evaluate the experiments, the following classification measures are used: average accuracy (ACC), micro-averaged F_1 measure ($F_1\mu$) and macro-averaged F_1 measure (F_1M). ACC represent average per-class effectiveness of a classifier:

$$ACC = \frac{1}{|M|}\sum_{i=1}^{|M|}\frac{\mathrm{TP}_i + \mathrm{TN}_i}{\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i + \mathrm{TN}_i}. \tag{5}$$

The micro-averaged F_1 measure represents the relations between the data's positive labels and the labels given by a classifier that is based on the sums of the per-class decisions. On the other hand, the macro-averaged F_1 measure represents the relations between data's positive labels and the labels given by a classifier based on a per-class average [35]. Macro and micro averaged measures were used to assess the performance for the majority and minority classes. Their use is due to the fact the macro-averaged measures are more sensitive to the performance for minority classes.

The F_1 measures are defined as follows:

$$F_1M = \frac{1}{|M|}\sum_{i=1}^{|M|}\frac{\mathrm{FP}_i + \mathrm{FN}_i}{2\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i}, \tag{6}$$

$$F_1\mu = \frac{\sum_{i=1}^{|M|}\mathrm{FP}_i + \sum_{i=1}^{|M|}\mathrm{FN}_i}{2\mathrm{TP}_i + \sum_{i=1}^{|M|}\mathrm{FP}_i + \sum_{i=1}^{|M|}\mathrm{FN}_i}, \tag{7}$$

where TP_i, TN_i, FP_i, FN_i are class-specific true positive, true negative, false positive, and false negative rates, respectively.

Table 1
Example of cutting tools data sets – before the pre-processing

Feature number															Class labels
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	
contouring the curve	EN-AW 5754	6.30	monolit	pin	16	cylindrical	3	32	250	500	9	1	796	120	Fi16W
groove	316L	10	monolith	pin	20	cylindrical	5	104	96	200	10	0.50	445	260	203012_20
contour	316L	10	monolith	pin	20	cylindrical	5	104	96	200	5	0.25	509	260	203012_20
groove	316L	20	monolith	pin	25	cylindrical	4	125	80	160	25	1	285	250	203014_25
contour	316L	20	monolith	pin	25	cylindrical	4	125	80	160	12.50	0.50	326	250	203014_25
groove	316L	20	monolith	pin	16	cylindrical	4	92	85	170	16	1	338	350	203016_16
contour	316L	20	monolith	pin	16	cylindrical	4	92	85	170	8	0.50	372	350	203016_16
groove	316L	20	monolith	pin	20	cylindrical	4	175	40	80	20	1	178	315	203022_20
contour	316L	20	monolith	pin	20	cylindrical	4	175	40	80	10	0.50	216	315	203022_20
groove	316L	20	monolith	pin	10	cylindrical	4	100	45	90	10	1	229	75	203025_10
contour	316L	20	monolith	pin	10	cylindrical	4	100	45	90	5	0.50	286	75	203025_10
groove	316L	20	monolith	pin	20	cylindrical	4	92	50	100	20	1	318	196	203031_20
contour	316L	20	monolith	pin	20	cylindrical	4	92	50	100	10	0.50	414	196	203031_20
contour	316L	10	monolith	pin	25	cylindrical	6	165	78	160	6.25	0.25	596	730	203540_25
contour	316L	10	monolith	pin	25	cylindrical	6	165	78	160	6.25	0.25	596	730	203540_25
contour	316L	10	monolith	pin	25	cylindrical	8	165	78	160	6.25	0.25	795	730	203540_25
contour	316L	10	monolith	pin	25	cylindrical	8	165	78	160	6.25	0.25	795	730	203540_25
plane	316L	20	folded	sockets	100	sockets	7	55	120	240	30	0.30	722	765	222403_100/7
plane	316L	20	folded	sockets	100	sockets	7	55	120	240	30	0.30	722	765	222403_100/7
plane	316L	20	folded	sockets	40	sockets	4	35	120	240	12	0.30	1032	455	222403_40
plane	316L	10	folded	sockets	40	sockets	4	35	144	300	6	0.15	1032	455	222403_40
plane	S235JR	20	folded	sockets	100	sockets	6	50	180	360	30	0.30	240	595	222800_100/6
plane	S235JR	10	folded	sockets	100	sockets	6	50	216	420	15	0.15	289	595	222800_100/6
plane	316L	20	folded	sockets	125	sockets	7	63	140	280	30	0.30	187	760	222800_125/7
plane	316L	10	folded	sockets	125	sockets	7	63	168	340	15	0.15	187	760	222800_125/7
plane	316L	20	folded	sockets	250	sockets	20	63	140	280	75	0.30	249	1700	222800_250
plane	316L	10	folded	sockets	250	sockets	20	63	168	340	37.50	0.15	249	1700	222800_250
plane	316L	20	folded	sockets	63	sockets	4	40	140	280	19	0.30	198	410	222800_63
plane	316L	10	folded	sockets	63	sockets	4	40	168	340	9.50	0.15	198	410	222800_63
plane	316L	20	folded	sockets	80	sockets	6	50	140	280	30	0.30	187	486	222800_80
plane	316L	10	folded	sockets	80	sockets	6	50	168	340	15	0.15	187	486	222800_80

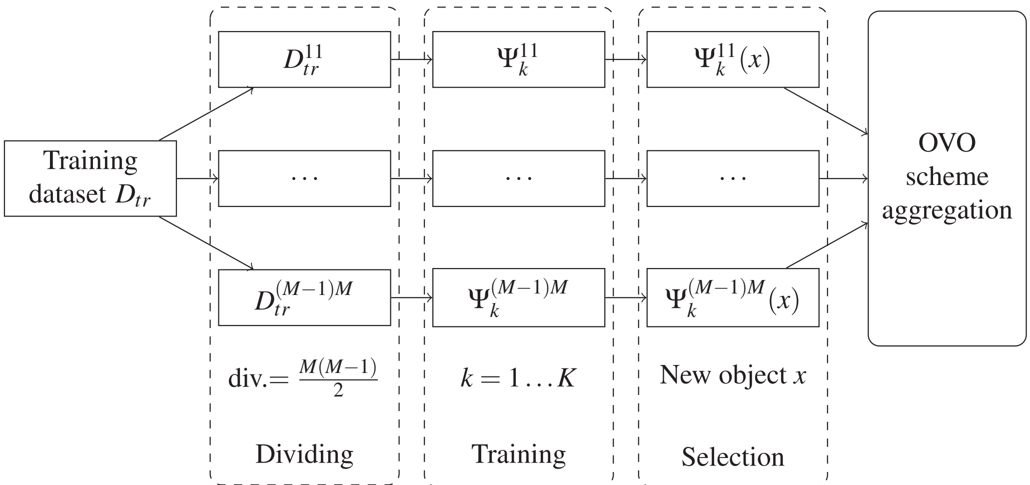

Fig. 2. Layered diagram of the OVO method with the base classifier selection process

4.3. Experiment setup for the dynamic ensemble selection. During the experiment on the dynamic ensemble selection in the OVO scheme, 16 base classifiers were used. We use the SVM model with different kernel types. Table 2 presents kernels and parameters of the SVM classifiers used in the experiment.

Table 2
The parameters of the SVM models

Classifier	Estimation method	Tuning method	Regular parameter	Kernel
Ψ_1	LSVM	Grid	Tuning	Linear
Ψ_2	DQP	Optimal	Tuning	Linear
Ψ_3	LSSVM	Optimal	Tuning	Linear
Ψ_4	FQP	Grid	Tuning	Linear
Ψ_5	DQP	Optimal	Tuning	RBF
Ψ_6	FQP	Optimal	Tuning	RBF
Ψ_7	LSVM	Grid	Tuning	RBF
Ψ_8	LSSVM	Optimal	Tuning	RBF
Ψ_9	DQP	Optimal	Tuning	Polyn.
Ψ_{10}	FQP	Optimal	Constant (0.1)	Polyn.
Ψ_{11}	LSVM	Optimal	Tuning	Polyn.
Ψ_{12}	LSSVM	Optimal	Tuning	Polyn.
Ψ_{13}	DQP	Optimal	Tuning	Sigmoid
Ψ_{14}	LSSVM	Grid	Tuning	Polyn.
Ψ_{15}	DQP	Grid	Tuning	Sigmoid
Ψ_{16}	DQP	Optimal	Constant (0.3)	Linear

Experimental studies for the dynamic ensemble selection were carried out for various sets of parameters α and β. We use sets of those parameters, where P_n, $n = 1 \dots 25$. The parameters α and β take values according to the following formulas $\alpha = (0.5 - n/100)$ and $\beta = (0.5 + n/100)$.

4.4. Experiment setup for static ensemble selection. During the experiment on static ensemble selection in the OVO scheme, 3 base classifiers were used. We use linear base classifiers; that is, classifiers whose decision boundary is a linear function. The use of linear classifiers results from the properties of the integration of the base classifiers, which uses the values of these linear functions. In particular, we used the following classification models: Ψ_{MLP} – single layer MLP classifier, Ψ_{SVM} – SVM classifier with linear kernel and Ψ_{NC} – nearest centroid with the class-specific Mahalanobis distance classifier.

The environment SAS 9.4 and SAS Enterprise Miner were used to perform the experiments. The presented results are obtained via a 10-fold-cross-validation method.

5. Results and discussion

For the ACC performance measure, the best set of parameters for the dynamic ensemble selection is as follows: P_8: $\alpha = 0.42$, $\beta = 0.58$. In the results analyzing the algorithms Ψ^V_{DESOVO} and Ψ^{WV}_{DESOVO}, we used this set of parameters. The results obtained for the algorithms Ψ^V_{DESOVO} and Ψ^{WV}_{DESOVO} were compared with the results obtained by the base classifiers $\Psi_1, \dots, \Psi_{16}$ and ensemble of this base classifier without selection Ψ^{MV}_{OVO}. In addition, the static ensemble selection algorithm Ψ^V_{SESOVO} and base classifiers used in static ensemble selection Ψ_{MLP}, Ψ_{SVM}, Ψ_{NC} were also compared. Each base classifier also worked in the OVO scheme. Table 3 shows the results for three classification measures ACC, $F_1\mu$ and F_1M. The visualization of the obtained results is presented in Fig. 3 where the axis range is $0.804 - 1$. These results means that the smallest value from Table 3, which equals 0.804, is in the middle of the radar plot chart.

Table 3
Results of experimental research for three classification performance measures

Algorithm	ACC	F_1M	$F_1\mu$
Ψ_1	0.972	0.873	0.986
Ψ_2	0.966	0.869	0.979
Ψ_3	0.972	0.873	0.986
Ψ_4	0.972	0.873	0.986
Ψ_5	0.972	0.873	0.986
Ψ_6	0.954	0.858	0.967
Ψ_7	0.972	0.873	0.986
Ψ_8	0.964	0.865	0.977
Ψ_9	0.972	0.873	0.986
Ψ_{10}	0.972	0.873	0.986
Ψ_{11}	0.972	0.873	0.986
Ψ_{12}	0.964	0.865	0.977
Ψ_{13}	0.972	0.873	0.986
Ψ_{14}	0.972	0.873	0.986
Ψ_{15}	0.972	0.873	0.986
Ψ_{16}	0.964	0.865	0.977
Ψ^V_{OVO}	0.971	0.871	0.984
Ψ^{WV}_{OVO}	0.901	0.804	0.918
Ψ^V_{DESOVO}	0.986	0.882	1.000
Ψ^{WV}_{DESOVO}	0.969	0.864	0.982
Ψ_{MLP}	0.973	0.863	0.973
Ψ_{SVM}	0.967	0.865	0.967
Ψ_{NC}	0.962	0.873	0.962
Ψ^V_{SESOVO}	0.975	0.877	0.975

The obtained results indicate that the proposed dynamic ensemble selection in the OVO scheme method significantly improves the quality of the cutting tools classification. In the case of $F_1\mu$, the classification performance measures we proposed obtained a perfect classification result. Also, for performance measures ACC and F_1M, the approach proposed in this paper achieves the best results when compared with the static ensemble selection in the OVO scheme and all base classifiers

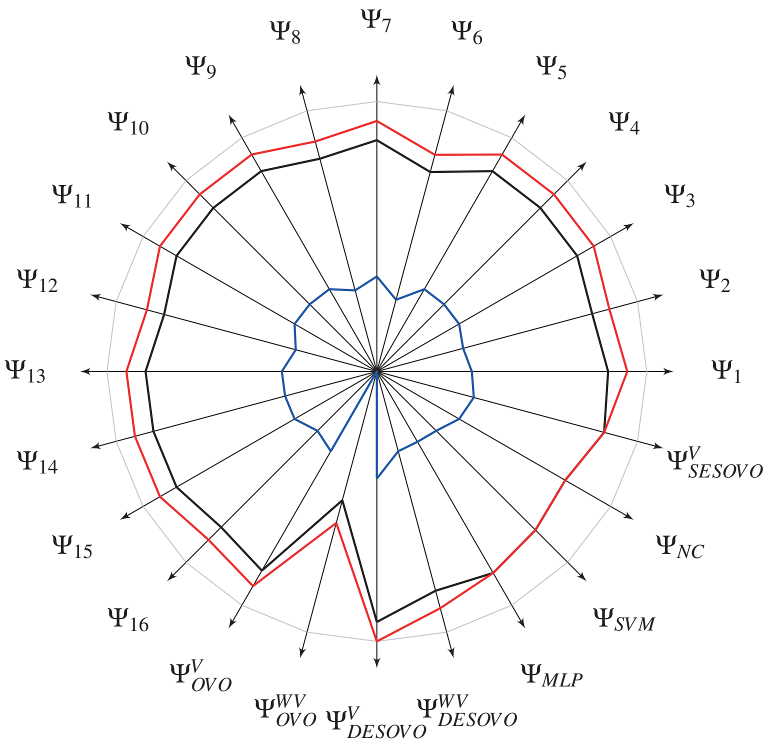

Fig. 3. Radar plot for the three classification performance measures and all classifiers – the red line for $F_1\mu$, the blue line for F_1M and the black line for ACC performance measure

used in the experiment. In addition, the method proposed in this article obtains better classification performance results than the methods described in [26, 36], which are the earlier works of the authors.

6. Conclusions

This paper presented a new dynamic ensemble selection method for the multi-class problem by the application the OVO scheme. The proposed algorithm used the values of the score function in the selection process. The proposed approach took into account the location of the recognized object relative to the decision boundaries of the base classifiers. The results obtained on the real classification problems – tool selections performed during the design of manufacturing processes – indicated clearly that the proposed method achieves very good results compared to other algorithms described in both the community literature and the previous works of the authors.

Acknowledgements. This work was supported in part by the National Science Centre, Poland under the grant no. 2017/25/B/ST6/01750 and statutory funds of Institute of Computer Science, Kazimierz Wielki University.

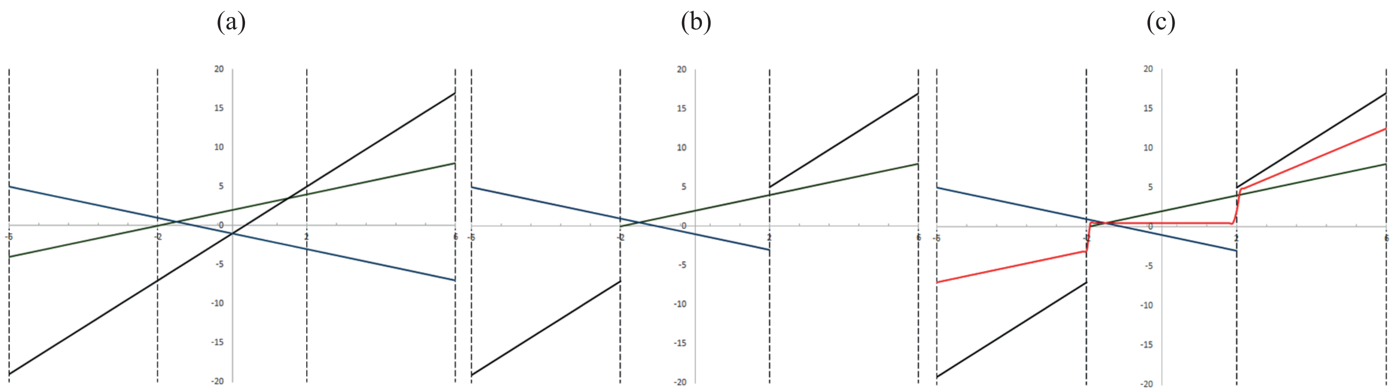

Fig. 4. An example of the selection process for three base classifiers and three regions of competence: a) Decision boundaries of the three base classifiers and three regions of competence, b) Decision boundaries of the base classifiers after selection one base classifier in each region of competence, c) Final decision boundary – red

References

[1] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning and Data Mining*. Springer US, 2016.

[2] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[3] S. Osowski and K. Siwek, "Local dynamic integration of ensemble in prediction of time series", *Bull. Pol. Ac.: Tech.* 67(3), 517–525 (2019).

[4] O. Sagi and L. Rokach, "Ensemble learning: A survey", *Wiley Interdiscip. Rev.-Data Mining Knowl. Discov.* 8(4), e1249 (2018).

[5] R.M. Cruz, R. Sabourin, and G.D. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives", *Inf. Fusion* 41, 195–216 (2018).

[6] O.A. Alzubi, J.A. Alzubi, M. Alweshah, I. Qiqieh, S. AlShami, and M. Ramachandran, "An optimal pruning algorithm of classifier ensembles: dynamic programming approach", *Neural Comput. Appl.* 32, 16091–16107 (2020).

[7] Y. Bian, Y. Wang, Y. Yao, and H. Chen, "Ensemble pruning based on objection maximization with a general distributed framework", *IEEE Trans. Neural Netw. Learn. Syst.* 31(9), 3766–3774 (2020).

[8] R.M. Cruz, D.V. Oliveira, G.D. Cavalcanti, and R. Sabourin, "Fire-des++: Enhanced online pruning of base classifiers for dynamic ensemble selection", *Pattern Recognit.* 85, 149–160 (2019).

[9] T.T. Nguyen, A.V. Luong, M.T. Dang, A.W.-C. Liew, and J. McCall, "Ensemble selection based on classifier prediction confidence", *Pattern Recognit.* 100, 107104 (2020).

[10] Z.-L. Zhang, X.-G. Luo, S. García, J.-F. Tang, and F. Herrera, "Exploring the effectiveness of dynamic ensemble selection in the one-versus-one scheme", *Knowledge-Based Syst.* 125, 53–63 (2017).

[11] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "Dynamic classifier selection for one-vs-one strategy: avoiding non-competent classifiers", *Pattern Recognit.* 46(12), 3412–3424 (2013).

[12] M. Pawlicki, A. Giełczyk, R. Kozik, and M. Choraś, "Fault-prone software classes recognition via artificial neural network with granular dataset balancing", in *International Conference on Computer Recognition Systems 2019*, Springer, 2019, pp. 130–140.

[13] D. Rajeev, D. Dinakaran, and S. Singh, "Artificial neural network based tool wear estimation on dry hard turning processes of aisi4140 steel using coated carbide tool", *Bull. Pol. Ac.: Tech.* 65(4), 553–559 (2017).

[14] D. Więcek, A. Burduk, and I. Kuric, "The use of ann in improving efficiency and ensuring the stability of the copper ore mining process", *Acta Montanistica Slovaca* 24(1), 1–14 (2019).

[15] P. Raja, R. Malayalamurthim, and M. Sakthivel, "Experimental investigation of cryogenically treated hss tool in turning on aisi1045 using fuzzy logic–taguchi approach", *Bull. Pol. Ac.: Tech.* 67(4), 687–696 (2019).

[16] T. Andrysiak and L. Saganowski, "Anomaly detection for smart lighting infrastructure with the use of time series analysis", *J. UCS* 26(4), 508–527 (2020).

[17] A. Burduk, K. Musiał, J. Kochańska, D. Górnicka, and A. Stetsenko, "Tabu search and genetic algorithm for production process scheduling problem", *LogForum* 15, 181–189 (2019.

[18] M. Choraś, M. Pawlicki, D. Puchalski, and R. Kozik, "Machine learning–the results are not the only thing that matters! what about security, explainability and fairness?", in *International Conference on Computational Science*, Springer, 2020, pp. 615–628.

[19] P. Zarychta, P. Badura, and E. Pietka, "Comparative analysis of selected classifiers in posterior cruciate ligaments computer aided diagnosis", *Bull. Pol. Ac.: Tech.* 65(1), 63–70 (2017).

[20] I. Rojek, E. Dostatni, and A. Hamrol, "Ecodesign of technological processes with the use of decision trees method", in *International Joint Conference SOCO'17-CISIS'17-ICEUTE'17,* León, Spain, 2017, Springer, 2018, pp. 318–327.

[21] I. Rojek and E. Dostatni, "Machine learning methods for optimal compatibility of materials in ecodesign", *Bull. Pol. Ac.: Tech.* 68(2), 199–206 (2020).

[22] P. Prokopowicz, D. Mikołajewski, K. Tyburek, and E. Mikołajewska, "Computational gait analysis for post-stroke rehabilitation purposes using fuzzy numbers, fractal dimension and neural networks", *Bull. Pol. Ac.: Tech.* 68(2), 191–198 (2020).

[23] S. Igari, F. Tanaka, and M. Onosato, "Customization of a micro process planning system for an actual machine tool based on updating a machining database and generating a database-oriented planning algorithm", *Trans. Inst. Syst. Control Inform. Eng.* 26(3), 87–94 (2013).

[24] C. Tan and S. Ranjit, "An expert carbide cutting tools selection system for cnc lathe machine", *Int. Rev. Mech. Eng.* 6(7), 1402–1405 (2012).

[25] I. Rojek, "Technological process planning by the use of neural networks", *AI EDAM – AI EDAM-Artif. Intell. Eng. Des. Anal. Manuf.* 31(1), 1–15 (2017).

[26] P. Heda, I. Rojek, and R. Burduk, "Dynamic ensemble selection – application to classification of cutting tools", in *International Conference on Computer Information Systems and Industrial Management LNCS(12133)*, Springer, 2020, pp. 345–354.

[27] L.I. Kuncheva, *Combining Pattern Classifiers*. John Wiley & Sons, Inc., 2014.

[28] E. Santucci, L. Didaci, G. Fumera, and F. Roli, "A parameter randomization approach for constructing classifier ensembles", *Pattern Recognit.* 69, 1–13 (2017).

[29] M. Mohandes, M. Deriche, and S. O. Aliyu, "Classifiers combination techniques: A comprehensive review", *IEEE Access* 6, 19626–19639 (2018).

[30] J. Yan, Z. Zhang, K. Lin, F. Yang, and X. Luo, "A hybrid scheme-based one-vs-all decision trees for multi-class classification tasks", *Knowledge-Based Syst.* 198. 105922 (2020).

[31] P. Chaitra and R.S. Kumar, "A review of multi-class classification algorithms", *Int. J. Pure Appl. Math.* 118(14), 17–26 (2018).

[32] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on onevs-one and one-vs-all schemes", *Pattern Recognit.* 44(8), 1761–1776 (2011).

[33] R. Burduk, "Integration base classifiers based on their decision boundary", in *International Conference on Artificial Intelligence and Soft Computing*, Springer, 2017, pp. 13–20.

[34] M.P. Groover, *Fundamentals of modern manufacturing: materials, processes and systems*, Willey, 2010.

[35] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks", *Inf. Process. Manage.* 45, 427–437 (2009).

[36] I. Rojek, "Classifier models in intelligent capp systems", in *Man-Machine Interactions*, pp. 311–319, Springer, 2009.