Multiple-Meta-Instance Selection. Combining the Properties of Many Instance-Selection Methods.

Marcin Blachnik[1 [0000−0003−3336−4962]],
Piotr Ciepliński[1 [0000−0001−9056−0367]], and
Daniel Dąbrowski[1 [0009−0007−4250−7060]]

Silesian University of Technology,
Department of Industrial Informatics,
40-019 Katowice ul. Krasińskiego 8, Poland
{marcin.blachnik, piotr.cieplinski, daniel.dabrowski}@polsl.pl

Abstract. This study presents two novel approaches for developing a multiple-meta-instance selection method, an advanced algorithm designed for efficient pruning of training sample in classification problems. The proposed meta-instance selection framework reformulates the traditional instance selection problem by introducing a meta-feature space, a problem-agnostic representation space. The transformation enables instance selection to be framed as a classification task in the meta-feature space, facilitating efficient computation with a time complexity of $O(nlog(n))$. A standard classification algorithm, such as Random Forest, can then be employed in the meta-feature space to determine the inclusion or exclusion of individual samples.
To enhance performance, we explore two strategies for combining multiple meta-instance selection algorithms: (1) constructing an ensemble of meta-classifiers and (2) concatenating many meta-sets. Experimental evaluations demonstrate that the meta-set concatenation approach surpasses both classical instance selection techniques and existing meta-instance selection methods. Moreover, the proposed algorithm significantly accelerates the instance selection process—achieving even by two or three orders of magnitude speed-up, depending on dataset size and the reference instance selection method.

Keywords: Instance selection · classification · knowledge distillation · data pruning

1 Introduction

As datasets used in machine learning applications continue to expand in size and complexity, managing, storing, and processing them efficiently is getting more challenging [19]. One of the approaches to tackle these problems is dataset pruning [13]. This is a group of methods used to reduce the size of a dataset while preserving its essential characteristics, making it a valuable tool for improving data management, model training, and overall system performance. More precisely it can be viewed as a process of selecting or constructing a subset of the

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16

most informative and representative data points from a larger dataset, with the goal of retaining the majority of the information and patterns present in the original data [15]. There are two main approaches including model-independent pruning and model-dependent pruning. The first group selects samples based on some external data characteristics [5], while the second one is also called dataset distillation [18] where the final model is used to select or construct new training samples. In both cases, the resulting dataset is typically much smaller. The reduction process involves identifying the most critical data points that capture the underlying structure and relationships within the data and removing redundant or noisy data points that do not contribute significantly to the task the model is built for.

One of the most commonly used techniques for dataset pruning is instance selection. Comprehensive reviews of instance selection methods can be found in [8] and [10]. This family of techniques was initially develped as a tool to improve the k-nearest neighbor (kNN) classifiers, but later it was effectively adopted also to other classification methods [3,7].

However, instance selection suffers from significant computational complexity, as most algorithms iteratively evaluate the nearest-neighbor graph to prune redundant or noisy samples. To address this problem, a method called Meta Instance Selection (MetaIS) was proposed in [2]. This method tackles scalability issues by reframing the instance selection problem as a classification problem in a meta-feature space. In this approach, each sample is mapped into a new meta-feature space that describes the local properties of the nearest-neighbor graph, and a meta-classifier is used to determine whether a given sample should be retained or pruned. The meta-classifier is initially trained to emulate the behaviour of a specific instance selection algorithm. This is achieved by using the output of the classical instance selection method to label samples in the meta-feature space as either "to be kept" or "to be removed" and training a classifier (meta-classifier). Consequently, a single meta-classifier corresponds to a single instance selection algorithm.

In this work, we propose extending the capabilities of meta-instance selection by combining the properties of multiple instance selection methods into a single method which is called multiple-meta-instance selection (MMIS). This approach allows for further improvement of the properties of classical MetaIS solutions and achieving even higher performance without affecting execution time.

In the article, we discuss and empirically compare two methods of constructing MMIS. The first method is based of combining multiple meta-classifiers of MetaIS into a classifiers committee that combines the properties of individual instance selection methods into one system. The second approach is based on concatenating the meta-datasets obtained from the base instance selection algorithms into one large dataset, and consequently training a single meta-classifier on the combined meta-dataset. This assures gaining the knowledge from each base instance selection method.

The process of model evaluation allows for selecting the optimal combination of base-instance selection methods that complement each other. This allows for

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16

achieving superior performance compared to individual methods. Importantly, the gain in performance is achieved with minimal additional computational complexity.

The structure of this paper is as follows: first, we present an overview of meta-instance selection methods (section 2). Next, in section 3 the concept of multi meta instance selection is introduce. Section 4 describes setup of the experiments and results are presented in Section 5. Finally, the concluding section summarizes the findings and outlines potential directions for future research.

2 Meta Instance Selection

Meta-instance selection reformulates the problem of selection or rejection of a sample as a binary classification problem, where positive samples are labeled as "to keep" and negative samples are labeled as "to remove".

To make this process generic and applicable to any dataset, a common feature space is required in which the meta-classifier (a classifier responsible for assessing instance importance) operates — the so-called meta-feature space. In [2] it is suggested to use balanced random forest as a meta-classifier, and the meta-feature space is defined using local properties of the nearest neighbor graph (NNG), where each sample is characterized by statistics derived from the NNG. More details regarding the meta-feature space are provided in subsection 2.1. The basic concept is shown in Figure 1.

Fig. 1: The concept of MetaIS algorithm. The left figure shows the transformation process, where for the input dataset an NNG is constructed and properties of each vertex constitute the meta-feature space. Labeling of samples in the meta-set (marked in yellow) is applied only during the preparation of the meta-training set. The right figure shows the transformation results from regular to the meta-features space.

A schematic overview of the entire system is shown in Figure 2, which is divided into two components: the training phase and the prediction/selection phase. The training phase illustrates the process of constructing the meta-classifier, while the prediction/selection phase represents the process of instance selection using the meta-classifier.

The training phase begins by applying a specific instance selection method, referred to as the reference method. This reference instance selection method is

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16

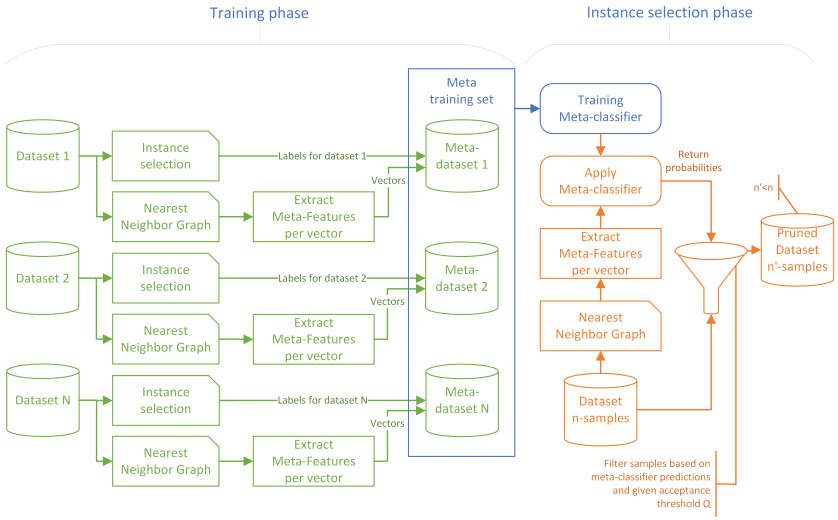


Fig. 2: The scheme of the MetaIS algorithm including preparation of the meta-set (green color), training meta-classifier (blue color), and application of MetaIS for pruning new dataset (orange color).

executed on multiple datasets (the larger the number of datasets, the better the performance), and each sample in these datasets is labeled as either "to keep" or "to remove" based on the results of the reference method. Subsequently, for each sample in each dataset, meta-feature extraction is performed. The resulting pairs, represented as $< \mathbf{x}_{meta}, \{y_{Positive(\text{to keep})}, y_{Negative(\text{to remove})}\} >$, form the training set for the meta-classifier.

Following meta-feature extraction, the features for each sample in each dataset are normalized. This normalization step is crucial because the distances within individual datasets may vary significantly. After normalization, all meta-datasets are concatenated to create a unified training set for a binary classifier.

During the prediction phase, for a new dataset, meta-features are first extracted. Specifically, for each training sample $(\mathbf{x}, y$ in the dataset, its meta-features $\mathbf{x}_{meta}$ are computed. The meta-classifier is then applied to the samples represented using meta-features. Instead of producing a classical binary decision, the model typically outputs a probability score, which represents the importance of each instance rather than a definitive decision. Finally, based on these importance scores, the samples are ranked, and a selection is made according to a user-defined threshold Θ, where 0 indicates no instance selection and 1 indicates prunning the entire dataset, or based on preferences regarding the desired output size of the dataset.

2.1 Meta-Feature Space

As previously indicated, the meta-feature space is determined based on properties derived from the nearest neighbor graph (NNG). These meta-features are

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16

extracted from various attributes that were initially used in the reference instance selection methods. The details are provided in [2], and here, only the names and descriptions of the extracted features are presented:

- Average distance to k nearest neighbors from the same class: Among k nearest neighbors, find samples with the same class label as the query sample and get their average.
- Average distance to k nearest neighbors from the opposite class: Similar to the above, but calculate the average distance to the k nearest neighbors that belong to the opposite class of the query sample.
- Average distance to k nearest neighbors from any class: Compute the average distance to all k nearest neighbors, irrespective of class label.
- Minimum distance to samples from the same class: Determine the distance to the nearest neighbor that belongs to the same class as the query sample.
- Minimum distance to samples from the opposite class: Calculate the distance to the nearest neighbor that belongs to the opposite class of the query sample.
- Minimum distance to samples from any class: Determine the distance to the overall nearest neighbor, irrespective of class label.
- Number of samples from the same class among k nearest neighbors: Perform a vote among the k nearest neighbors to count how many belong to the same class as the query sample.
- Number of samples from the opposite class among k nearest neighbors: Similar to the above, but count how many of the k nearest neighbors belong to the opposite class.

When constructing the meta-feature space, these meta-features are computed for multiple values of k, specifically $k = \{3, 5, 9, 15, 23, 33\}$ as indicated in [2]. As demonstrated, the use of multiple values of k, combined with different types of meta-features, is critical for accurately characterizing the neighborhood of the query sample. This approach enables a more precise determination of the sample's importance.

3 Combining Multiple Instance Selection Methods

The Meta Instance Selection (MetaIS) method offers several advantages over traditional instance selection techniques, particularly in its ability to efficiently combine individual instance selection methods. While the concept of instance selection ensembles was initially introduced in [1] and later refined in [9], the MetaIS approach significantly enhances this process by enabling more efficient combinations of instance selection methods. Instead of running multiple computationally expensive instance selection algorithms, MetaIS leverages two efficient strategies: (1) an ensemble of meta-classifiers and (2) a meta-classifier trained on concatenated meta-datasets representing reference instance selection methods. A detailed explanation of these approaches is provided below.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16

Ensemble of Meta-Classifiers This approach is based on the idea that each meta-classifier is trained to mimic a specific reference instance selection method for example HMN-EI, CCIS, etc. Since these reference methods differ in the subsets they select, the meta-datasets used for training meta-classifiers are distinct in terms of their labeling. Consequently, meta-classifiers trained on different reference methods exhibit variations in outputs (predictions) assuring diversity of the ensemble members. The challenge lies in determining which meta-classifiers should be combined to ensure they complement one another effectively and enhance overall performance.

Concatenation of Meta-Datasets An alternative strategy for improving meta-classifier performance involves merging meta-datasets labeled according to specific reference instance selection methods into a single dataset. A single meta-classifier is then trained on this combined dataset. This approach results in a significantly larger training meta-dataset since it integrates subsets from multiple reference methods. However, the computational cost of training the meta-classifier is incurred only once, making this method practical and efficient in terms of training overhead.

Both approaches provide flexible and scalable solutions for combining multiple instance selection methods while maintaining computational efficiency, thereby addressing the key limitations of traditional instance selection techniques. Since the method combines multiple MetaIS methods into one system it is abbreviated as Multiple-Meta-Instance-Selection (MMIS).

4 Setup of the Experiments

The two proposed MMIS methods were evaluated empirically and validated on multiple datasets of varing size and domain. The details of the experiments are provided below.

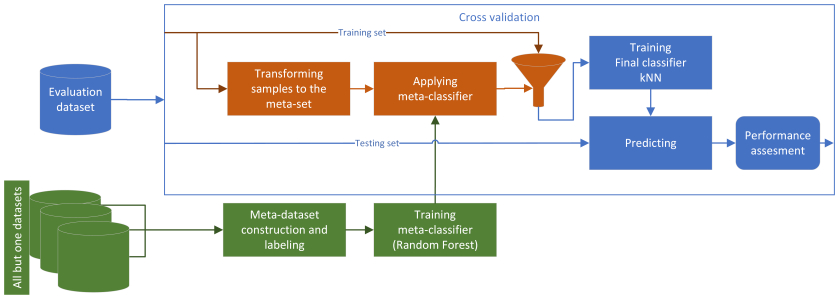


Fig. 3: The procedure used for performance assessment of the MetaIS and MMIS method. MetaIS/MMIS training is marked in green, brown is application of MMIS/MetaIS, and blue is standard cross-validation process.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16

Table 1: Characteristics of the datasets used in the experiments.

	Dataset	# samples	# attr.	# classes
1	abalone	4177	8	28
2	banana	5300	2	2
3	electricity	45312	8	2
4	letter	20000	16	26
5	magic	19020	10	2
6	nursery	12960	8	5
7	optdigits	5620	64	10
8	page-blocks	5472	10	5
9	penbased	10992	16	10
10	phoneme	5404	5	2
11	ring	7400	20	2
12	satimage	6435	36	6
13	shuttle	57999	9	7
14	spambase	4597	57	2
15	texture	5500	40	11
16	twonorm	7400	20	2
17	php89ntbG	488565	8	2

4.1 Evaluation procedure

The experiments were divided into two parts. In the first part, the leave-one-dataset-out methodology was employed to evaluate the performance of meta-instance selection. In the second part, the meta-classifier was tested on a single independent, large dataset containing around 500,000 samples. The leave-one-dataset-out method involves taking a collection of datasets (in this case, datasets with fewer than 100,000 samples), removing one dataset for testing, and using all remaining datasets to train the meta-classifier. Each dataset held out for testing was then pruned and the performance of the final classifier was assessed using 5-fold cross-validation. During the cross-validation procedure, the training data was first filtered using the instance selection method and subsequently evaluated using a 1NN classifier as shown in Fig. 3. The 1NN classifier is a standard practice in instance selection studies, and as meta-classifier the balanced random forest was used.

Since meta-instance selection outputs the probability of retaining a sample a curve representing the relation between reduction-rate and classification accuracy can be constructed using different Θ values. In particular, we used $\Theta = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Therefore, for measuring the overall performance the Area Under the Accuracy-Reduction Rate Curve ($AUARRC$) was used, as described in [1]. This metric is calculated by determining the area defined by the polygon formed between the performance of the 1NN classifier (at

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16

a zero reduction rate) and the results achieved by the instance selection method at a reduction rate greater than zero ($\Theta > 0$).

4.2 Datasets used in the experiments

The experiments were conducted on 17 datasets of various sizes. These datasets were obtained from the Keel Project repository [14], where they were preprocessed and provided in a format suitable for 5-fold cross-validation. While the repository contains additional, smaller datasets, only datasets with at least 4,000 samples were included in the study. This decision was made because removing redundant samples is not particularly useful for small datasets. In the experiments one larger datasets the *php89ntbG* dataset was obtained. It was obtained from the OpenML project [6].

4.3 Evaluation parameters

In Section 3, it was noted that the effectiveness of combined meta-instance selection (MMIS) depends on the choice of reference instance selection methods. Five methods were used to label meta-set samples: *Edited Nearest Neighbor* ENN [16], Drop3 [17], *Interactive Case Filtering* ICF [4], *Hit Miss Network Editing* HMN-EI [11], and *Class Conditional Instance Selection* CCIS [12]. Each method has distinct behavior, so selected methods must complement each other to avoid performance degradation. Various combinations were tested (see Table 4, bottom rows) to find the optimal set. The five evaluated combinations were: CCIS, ICF, CCIS, ICF, Drop3, CCIS, ICF, Drop3, HMN-EI, CCIS, ICF, HMN-EI, and CCIS, ICF, HMN-EI, ENN.

The primary criterion for selecting the optimal combination was the performance of the individual methods, where CCIS and ICF consistently achieved the highest accuracy and compression rates. These two methods served as the foundation, with additional methods (HMN-EI, Drop3, ENN) incorporated sequentially in order of their individual performance.

The results for the individual models were obtained for $\Theta = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, with performance assessment conducted using a 1NN classifier. For all reference instance selection methods (ENN, Drop3, CCIS, ICF and HMN-EI), the number of nearest neighbors was set to 3, as wherever parameter setup was required (3 is the default value suggested by the authors of particular methods). The experiments were carried out using our MetaIS library, implemented in Python and available at `https://github.com/mblachnik/MetaIS`. The reference instance selection methods were implemented in the RapidMiner Information Selection Extension `https://github.com/mblachnik/infoSel`, along with implementations from the Keel Project for selected algorithms (HMN-EI, CCIS).

The experiments consisted of two parts. In the first part, various combinations of reference methods were compared to identify the most effective combination. In the second part, the best-performing combination was compared against the individual reference instance selection methods.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16

Table 2: Results representing AUARRC performance measure obtained for two type of MMIS approaches - the meta-classifier based ensemble and concatenated meta-dataset - for various combination of base members. The last three rows summarize the obtained results. Statistically significant results are marked in bold ($\alpha = 0.05$)

	(ICF,CCIS)				(ICF,CCIS,Drop3)				(HMEI,ICF,CCIS)				(HMEI,ICF,CCIS,Drop3)				(HMEI,ENN,ICF,CCIS)			
	models ensemble		concatenated meta-datasets		models ensemble		concatenated meta-datasets		models ensemble		concatenated meta-datasets		models ensemble		concatenated meta-datasets		models ensemble		concatenated meta-datasets	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
plp89mthG	0.9282	0.0013	**0.9349**	0.0006	0.9061	0.0014	**0.9235**	0.0006	0.9440	0.0009	**0.9493**	0.0003	0.9205	0.0015	**0.9478**	0.0003	0.9426	0.0008	**0.9478**	0.0005
abalone	**0.2093**	0.0125	0.1897	0.0122	**0.2123**	0.0130	0.1938	0.0119	0.2228	0.0106	0.2183	0.0123	0.2205	0.0122	0.2153	0.0108	0.2222	0.0111	0.2152	0.0113
banana	0.8416	0.0111	0.8397	0.0086	0.8284	0.0104	0.8409	0.0124	0.8674	0.0078	**0.8796**	0.0042	0.8561	0.0069	**0.8745**	0.0073	0.8745	0.0052	**0.8836**	0.0050
letter	0.8015	0.0035	**0.8678**	0.0032	0.7686	0.0035	**0.8585**	0.0046	0.8178	0.0031	**0.8678**	0.0043	0.7887	0.0025	**0.8697**	0.0031	0.8108	0.0027	**0.8617**	0.0046
magic	0.7299	0.0044	**0.7716**	0.0070	0.6900	0.0038	**0.7585**	0.0064	0.8044	0.0086	0.8098	0.0033	0.7462	0.0032	**0.8054**	0.0043	0.8119	0.0067	0.8132	0.0041
nursery	0.8764	0.0106	0.8805	0.0093	0.8504	0.0173	**0.8767**	0.0124	0.8643	0.0093	**0.8826**	0.0123	0.8655	0.0100	**0.8862**	0.0145	0.8519	0.0107	**0.8859**	0.0140
optdigits	0.9241	0.0059	**0.9371**	0.0066	0.8748	0.0048	**0.9267**	0.0053	0.9243	0.0093	**0.9371**	0.0044	0.9087	0.0036	**0.9400**	0.0035	0.9135	0.0083	**0.9473**	0.0032
page-blocks	0.9072	0.0144	0.9092	0.0088	0.8649	0.0115	**0.9040**	0.0174	0.9480	0.0068	0.9494	0.0050	0.9174	0.0112	**0.9545**	0.0033	0.9440	0.0047	**0.9536**	0.0061
penbased	0.9773	0.0009	0.9774	0.0019	0.9584	0.0019	**0.9765**	0.0014	0.9700	0.0022	**0.9767**	0.0030	0.9682	0.0013	**0.9763**	0.0024	0.9605	0.0017	**0.9760**	0.0018
phoneme	0.8174	0.0076	**0.8288**	0.0090	0.7850	0.0055	**0.8203**	0.0088	0.8370	0.0099	0.8391	0.0089	0.8097	0.0072	**0.8361**	0.0118	0.8389	0.0051	0.8373	0.0094
ring	0.6840	0.0075	**0.6921**	0.0100	0.6738	0.0057	**0.7298**	0.0183	**0.7136**	0.0179	0.6348	0.0057	**0.7038**	0.0085	0.6780	0.0086	**0.6877**	0.0091	0.6142	0.0038
satimage	0.7944	0.0122	**0.8241**	0.0029	0.7406	0.0062	**0.8023**	0.0076	0.8630	0.0039	**0.8806**	0.0066	0.7890	0.0088	**0.8745**	0.0050	0.8622	0.0025	**0.8829**	0.0038
spambase	0.7158	0.0043	**0.7507**	0.0096	0.7198	0.0092	**0.7438**	0.0048	0.7907	0.0045	**0.8427**	0.0141	0.7338	0.0067	**0.8445**	0.0120	0.8105	0.0119	**0.8547**	0.0102
texture	0.9217	0.0075	**0.9396**	0.0070	0.8700	0.0073	**0.9321**	0.0052	0.9389	0.0036	**0.9471**	0.0047	0.9141	0.0058	**0.9489**	0.0074	0.9306	0.0032	**0.9485**	0.0065
twonorm	0.9357	0.0052	0.9379	0.0050	0.9081	0.0042	**0.9281**	0.0056	0.9511	0.0052	0.9502	0.0019	0.9314	0.0055	**0.9456**	0.0038	0.9533	0.0047	0.9508	0.0039
shuttle	**0.9964**	0.0007	0.9771	0.0081	**0.9924**	0.0018	0.9743	0.0091	0.9962	0.0009	0.9955	0.0002	0.9943	0.0017	0.9951	0.0006	**0.9952**	0.0007	0.9939	0.0009
electricity-normalized	0.6847	0.0044	**0.7623**	0.0026	0.6761	0.0039	**0.7610**	0.0025	0.7873	0.0031	**0.7946**	0.0024	0.7361	0.0040	**0.7963**	0.0009	**0.8013**	0.0032	0.7868	0.0031
Mean	0.8086		0.8247		0.7835		0.8206		0.8377		0.8444		0.8120		0.8464		0.8360		0.8443	
Mean diff.	-0.0162				-0.0371				-0.0067				-0.0344				-0.0084			
p-value	0.0129				0.0004				0.0129				0.0005				0.0887			

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16

Table 3: Summary statistics - *mean difference* and *p-value* representing the comparison of the leading solution (see Table 2) DE(CCIS,ICF,Drop3,HMN-EI) method with the remaining solutions obtained using *concatenated datasets* approach. The statistics were obtained using data available in Table 2

	(CCIS, ICF, Drop3, HMN-EI)	(CCIS,ICF, HMN-EI)	(CCIS, ICF, HMN-EI,ENN)	(CCIS,ICF)	(CCIS,ICF, Drop3)
Mean diff.	ref.	0.0020	0.0021	0.0217	0.0258
p-value	ref.	0.8536	0.7563	0.0007	0.0026

5 Results

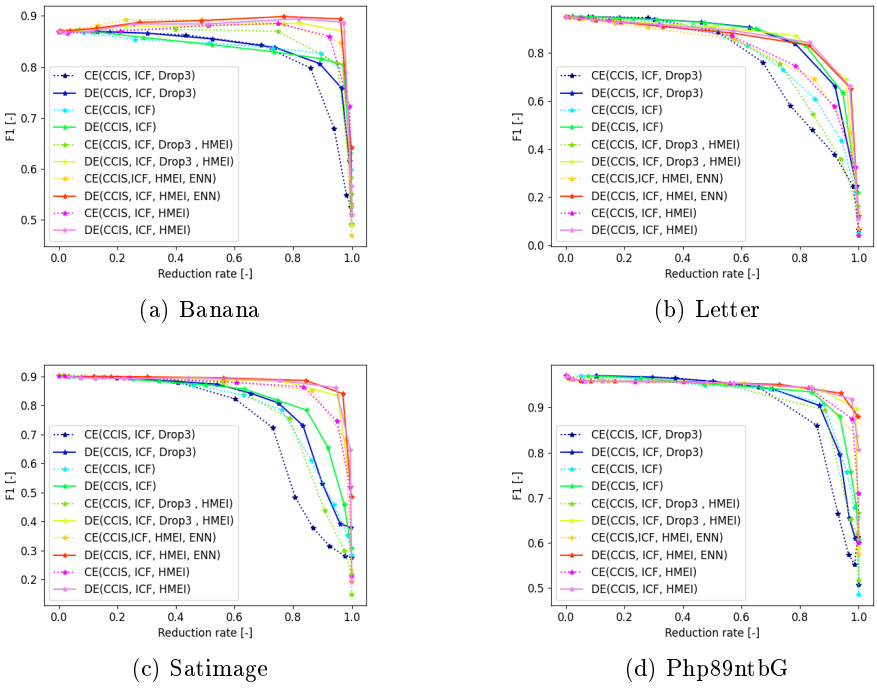
The first set of experiments was devoted to comparing the two approaches of creating MMIS, namely the meta-classifiers ensemble and concatenated meta-datasets. Within this comparison also the members of the ensemble including various IS models discussed in 4.3 were compared. The obtained results are presented in Table 2. The values represent the average AUARRC obtained using the given MMIS method for pruning training samples. For each dataset, the results are grouped according to the combination of ensemble members and according to the type of ensemble.

The last three rows of Table 2 summarize the obtained results. These are the mean AUARRC obtained by aggregating performances for each dataset, the average difference between the obtained results, and the p-value obtained using the Wilcoxon signed rank test obtained when comparing the two MMIS approaches. The average difference was calculated analogously to the Wilcoxon test, that is by $mean(left - right)$, where $left$ are the results of *models ensemble* and $right$ represent the column with *datasets ensemble*. A negative value of this indicator suggests that the *datasets ensemble* outperforms *models ensemble*, and a positive value indicates the opposite, that the *models ensemble* outperforms *datasets ensemble*.

The obtained results indicate that in all cases the mean difference is negative indicating that the *combined meta-datasets* performs better, and in all cases the difference is statistically significant, assuming $\alpha = 0.1$. When comparing the solutions among various members of the ensemble for the *combined meta-datasets* approach the best results are obtained by (CCIS, ICF, Drop3, HMN-EI), therefore this solution was used as a reference when comparing with the other methods.

Based on the results presented in Table 2, we conducted additional statistical analyses comparing the best-performing model (CCIS, ICF, Drop3, HMN-EI) with the remaining MMIS models from the *concatenated meta-dataset* families. These statistics are provided in Table 3. According to the results, the second-best performance is achieved by (CCIS, ICF, HMN-EI), followed by (CCIS, ICF, HMN-EI, ENN) in third place. While the differences among these three methods are not statistically significant, the positive values of the *mean differences* suggest that the (CCIS, ICF, Drop3, HMN-EI) based solution is the leading combination of reference instance selection methods. For the remaining approaches (CCIS,

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16



(a) Banana (b) Letter

(c) Satimage (d) Php89ntbG

Fig. 4: Comparison of the prediction performance vs reduction rate obtained for all evaluated MMIS methods on selected datasets. The CE( ) indicates classifiers ensemble and DE( ) indicates concatenated meta-datasets.

ICF) and (CCIS, ICF, Drop3), the differences in performance are statistically significant and worse then the best method.

A more detailed comparison of the differences is presented in Figure 4, which illustrates the *accuracy-reduction rate* plots for four selected datasets. The results indicate that, in all cases, the models highlighted in solid red, solid purple, and solid light green outperform the competitors. These models correspond to the *concatenated meta-dataset* approach based on concatenation of (CCIS, ICF, HMN-EI, ENN), (CCIS, ICF, HMN-EI), and (CCIS, ICF, Drop3, HMN-EI), respectively. Furthermore, in each case, the area under the *F1-reduction rate* curve is the largest, confirming their superior performance. Conversely, the approach based on (CCIS, ICF, Drop3) members yields the weakest results.

Next, we compared the best-performing MMIS model with the standard MetaIS models. The results, presented in Table 4, demonstrate that in all cases, the MMIS model is statistically significantly superior to each individual MetaIS model. The mean performance differences further highlight the advantages of the MMIS-based approach. A more detailed visualization is provided in Figure 5, where *performance-reduction rate* plots are shown for four datasets, including the best MMIS method and the standard MetaIS models. The dominance

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16

Table 4: Results represent the standard MetaIS models and the best MMIS model. The last three rows summarize the results indicating the mean performance over all datasets, the mean differences in performance considering the MMIS model denoted as (CCIS, ICF, Drop3, HMN-EI) as a reference for the comparison and the p-value of the Wilcoxon signed-rank test.

Dataset	DE(CCIS,ICF, Drop3,HMN-EI)	Meta HMN-EI	Meta ENN	Meta CCIS	Meta ICF	Meta Drop3
	mean ± std	mean ± std	mean ± std	mean ± std	mean ± std	mean ± std
banana	0.8745 ± 0.0073	0.6350 ± 0.0134	0.2839 ± 0.0055	0.8263 ± 0.0091	0.8610 ± 0.0076	0.8385 ± 0.0107
electricity-norm.	0.7963 ± 0.0009	0.7002 ± 0.0032	0.4727 ± 0.0030	0.6861 ± 0.0024	0.7401 ± 0.0034	0.7174 ± 0.0019
letter	0.8697 ± 0.0031	0.5845 ± 0.0033	0.3574 ± 0.0014	0.8611 ± 0.0025	0.7773 ± 0.0031	0.7381 ± 0.0062
magic	0.8054 ± 0.0043	0.5963 ± 0.0051	0.3526 ± 0.0034	0.7553 ± 0.0062	0.7009 ± 0.0036	0.6818 ± 0.0006
nursery	0.8862 ± 0.0145	0.7858 ± 0.0212	0.4021 ± 0.0305	0.8707 ± 0.0121	0.8817 ± 0.0125	0.8130 ± 0.0103
optdigits	0.9400 ± 0.0035	0.4241 ± 0.0053	0.1528 ± 0.0038	0.9427 ± 0.0057	0.9113 ± 0.0036	0.8549 ± 0.0049
page-blocks	0.9545 ± 0.0033	0.3769 ± 0.0047	0.1135 ± 0.0025	0.9369 ± 0.0093	0.9063 ± 0.0127	0.8563 ± 0.0189
penbased	0.9763 ± 0.0024	0.3319 ± 0.0070	0.0505 ± 0.0012	0.9687 ± 0.0011	0.9646 ± 0.0025	0.9591 ± 0.0038
phoneme	0.8361 ± 0.0118	0.6084 ± 0.0114	0.3822 ± 0.0014	0.8242 ± 0.0071	0.8033 ± 0.0084	0.7651 ± 0.0036
ring	0.6780 ± 0.0086	0.7857 ± 0.0095	0.3258 ± 0.0018	0.7492 ± 0.0084	0.6563 ± 0.0100	0.7133 ± 0.0035
satimage	0.8745 ± 0.0050	0.5032 ± 0.0041	0.2694 ± 0.0011	0.8491 ± 0.0041	0.7716 ± 0.0082	0.7386 ± 0.0041
shuttle	0.9951 ± 0.0006	0.5646 ± 0.0075	0.0060 ± 0.0003	0.9965 ± 0.0005	0.9922 ± 0.0012	0.9969 ± 0.0005
spambase	0.8445 ± 0.0120	0.6508 ± 0.0113	0.3416 ± 0.0112	0.7509 ± 0.0085	0.7479 ± 0.0087	0.7781 ± 0.0070
texture	0.9489 ± 0.0074	0.5360 ± 0.0072	0.1571 ± 0.0032	0.9539 ± 0.0043	0.8959 ± 0.0081	0.8529 ± 0.0125
twonorm	0.9456 ± 0.0038	0.5655 ± 0.0068	0.1954 ± 0.0044	0.9415 ± 0.0041	0.9208 ± 0.0039	0.9008 ± 0.0026
php89ntbG	0.9478 ± 0.0003	0.4754 ± 0.0032	0.1660 ± 0.0009	0.9456 ± 0.0005	0.9278 ± 0.0012	
Mean	**0.8858**	**0.5703**	**0.2518**	**0.8662**	**0.8412**	**0.8137**
Mean diff.		**0.3156**	**0.6340**	**0.0197**	**0.0447**	**0.0681**
p-value		**0.0002**	**0.0000**	**0.0182**	**0.0000**	**0.0004**

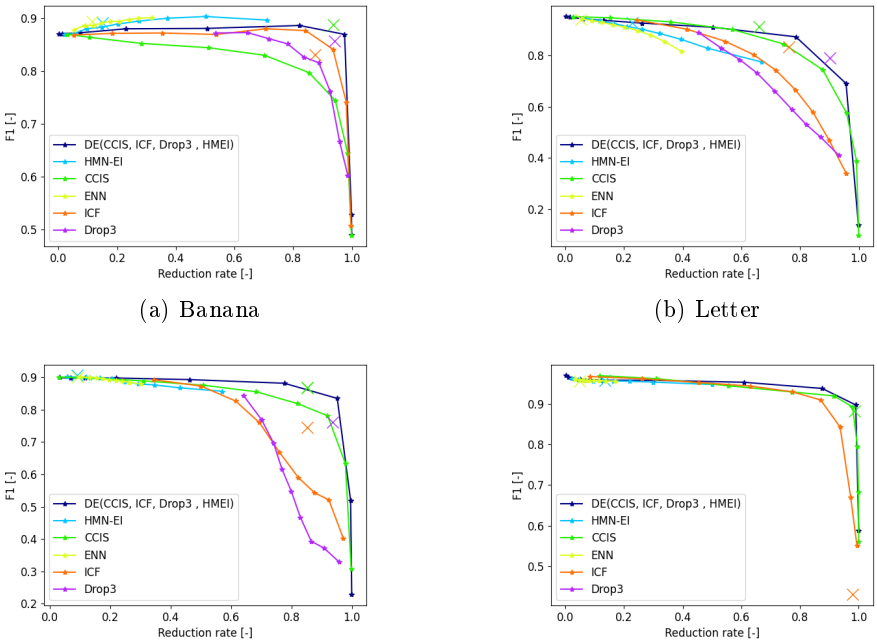
of the MMIS approach is evident, as indicated by the dark blue curve, which consistently outperforms all other methods. Additionally, the MMIS method frequently surpasses the baseline reference instance selection methods, marked with an $\times$, which were originally used for labeling the meta-datasets in the training process.

5.1 Execution time comparison

The final experiment demonstrates the efficiency of the proposed method in accelerating instance selection (Figure 6). Speedup was calculated as the ratio of the execution time of base instance selection methods to that of MMIS (solid line) and MetaIS (dotted line). Results show that speedup scales with the logarithm of sample size or better, with values below 1 only for small datasets (4,000–5,000 samples). The highest speedup—165×—was observed for Drop3.

This gain results from reduced time complexity, as MMIS requires only a single pass over the data. It includes meta-feature extraction in $O(n \log n)$ time and classification using a balanced random forest in $O(n \log n^*)$. In the concatenated meta-dataset approach, execution time depends only on classifier prediction time $O(\log n^*)$, where n^* is tree depth. For large datasets, MMIS and MetaIS have

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16



(a) Banana (b) Letter

(c) Satimage (d) Php89ntbG

Fig. 5: Comparison of the prediction performance vs reduction rate obtained for classical MetaIS methods and the best performing (CCIS, ICF, Drop3, HMN-EI) method based on concatenated meta-datasets approach. The DE( ) indicates concatenated meta-datasets.

similar runtimes due to shared meta-feature extraction and single classifier usage. Differences appear mainly for Drop3 and small datasets, where MetaIS built shallower trees, resulting in higher speedup for MMIS.

A difference between MetaIS and MMIS appears during the meta-classifier training. Here, for the *concatenated meta-datasets* approach used in MMIS the training dataset becomes larger since it is obtained by concatenation of multiple MetaIS datasets. But since the balanced random forest is used as a meta-classifier, the training time is acceptable because it scales by $O(k \cdot (n \log n))$ where k is the number of concatenated datasets. That is k- times slower than the MetaIS, but this process is conducted only once, and the meta-classifier can be used for any dataset. In the experiments, a single training took a couple of minutes.

6 Conclusions

This study introduced a novel approach to meta-instance selection, integrating multiple MetaIS methods into a unified framework called Multiple Meta Instance

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16

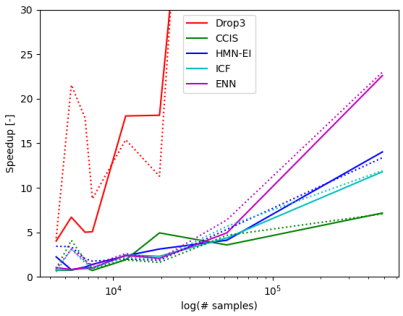


Fig. 6: Speed-up of the best proposed MMIS methods over the reference instance selection methods - marked with a solid line, and speed-up of MetaIS over the same reference instance selection, marked with a dotted line.

Selection (MMIS). Two strategies were explored: one leveraging an ensemble of meta-classifiers and another combining multiple individual methods by concatenating meta-training sets into a single dataset used to train meta-classifier. Experimental results demonstrate that the concatenated meta-dataset approach outperforms other approaches, both in terms of accuracy and reduction rate.

Furthermore, in the study we identified the most effective base-members. Among the tested methods, three demonstrated superior performance namely (CCIS, ICF, Drop3, HMN-EI), (CCIS, ICF, HMN-EI, ENN), and (CCIS, ICF, HMN-EI), with the (CCIS, ICF, Drop3, HMN-EI) yielding the best overall results.

In addition to improved selection accuracy, the proposed method significantly accelerates the instance selection process compared to baseline reference methods, offering substantial computational efficiency gains. These findings highlight the potential of meta-instance selection ensembles as a promising direction for optimizing data reduction in classification tasks. The proposed approach can be easily adapted also to other tasks such as support vector selection for the kernel-based methods, where instead of instance selection-based labeling support vectors can be used for labeling the meta-set.

Acknowledgments. The work was supported by the Excellence Initiative – Research University at the Silesian University of Technology, project number 11/040/SDW/10-21-01 and the research project BK-227/RM4/2025 also funded by the Silesian University of Technology

Disclosure of Interests. The authors declare no relevant conflicts of interest.

References

1. Blachnik, M.: Ensembles of instance selection methods. a comparative study. International Journal of Applied Mathematics and Computer Science **29**(1) (2019)
2. Blachnik, M., Ciepliński, P.: Meta-instance selection. instance selection as a classification problem with meta-features. arXiv preprint arXiv:2501.11526 (2025)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16

3. Blachnik, M., Kordos, M.: Comparison of instance selection and construction methods with various classifiers. Applied Sciences **10**(11), 3933 (2020)
4. Brighton, H., Mellish, C.: Advances in instance selection for instance-based learning algorithms. Data mining and knowledge discovery **6**(2), 153–172 (2002)
5. Cunha, W., Viegas, F., França, C., Rosa, T., Rocha, L., Gonçalves, M.A.: A comparative survey of instance selection methods applied to non-neural and transformer-based text classification. ACM Computing Surveys **55**(13s) (2023)
6. Feurer, M., Van Rijn, J.N., Kadra, A., Gijsbers, P., Mallik, N., Ravi, S., Müller, A., Vanschoren, J., Hutter, F.: Openml-python: an extensible python api for openml. Journal of Machine Learning Research **22**(100), 1–5 (2021)
7. García, S., Luengo, J., Herrera, F.: Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. Knowledge-Based Systems **98**, 1–29 (2016)
8. García, S., Derrac, J., Cano, J.R., Herrera, F.: Prototype selection for nearest neighbor classification: Taxonomy and empirical study. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34**(3), 417–435 (2012)
9. de Haro-García, A., Cerruela-García, G., García-Pedrajas, N.: Instance selection based on boosting for instance-based learners. Pattern Recognition **96**, 106959 (2019)
10. Malhat, M., El Menshawy, M., Mousa, H., El Sisi, A.: A new approach for instance selection: Algorithms, evaluation, and comparisons. Expert Systems with Applications **149**, 113297 (2020)
11. Marchiori, E.: Hit miss networks with applications to instance selection. Journal of Machine Learning Research **9**(Jun), 997–1017 (2008)
12. Marchiori, E.: Class conditional nearest neighbor for large margin instance selection. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(2), 364–370 (2010)
13. Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., Morcos, A.: Beyond neural scaling laws: beating power law scaling via data pruning. Advances in Neural Information Processing Systems **35**, 19523–19536 (2022)
14. Triguero, I., González, S., Moyano, J.M., García, S., Alcalá-Fdez, J., Luengo, J., Fernández, A., del Jesús, M.J., Sánchez, L., Herrera, F.: Keel 3.0: an open source software for multi-stage analysis in data mining. International Journal of Computational Intelligence Systems **10**(1), 1238–1249 (2017)
15. Wang, T., Zhu, J.Y., Torralba, A., Efros, A.A.: Dataset distillation. arXiv preprint arXiv:1811.10959 (2018)
16. Wilson, D.: Assymptotic properties of nearest neighbour rules using edited data. IEEE Trans. on Systems, Man, and Cybernetics **SMC-2**, 408–421 (1972)
17. Wilson, D., Martinez, T.: Reduction techniques for instance-based learning algorithms. ML **38**, 257–268 (2000)
18. Yu, R., Liu, S., Wang, X.: Dataset distillation: A comprehensive review. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)
19. Zha, D., Bhat, Z.P., Lai, K.H., Yang, F., Hu, X.: Data-centric ai: Perspectives and challenges. In: Proceedings of the 2023 SIAM International Conference on Data Mining (SDM). pp. 945–948. SIAM (2023)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_16