Applied Intelligence (2023) 53:26050–26066
https://doi.org/10.1007/s10489-023-04878-w

Neural network architecture with intermediate distribution-driven layer for classification of multidimensional data with low class separability

Weronika Borek-Marciniec[1] · Pawel Ksieniewicz[1]

Accepted: 7 July 2023 / Published online: 17 August 2023

Abstract

Simple neural network classification tasks are based on performing extraction as transformations of the set simultaneously with optimization of weights on individual layers. In this paper, the *Representation 7* architecture is proposed, the primary assumption of which is to divide the inductive procedure into separate blocks – transformation and decision – which may lead to a better generalization ability of the presented model. Architecture is based on the processing context of the typical neural network and unifies datasets into a shared, generically sampled space. It can be applicable in the case of difficult problems – defined not as imbalance or streaming data but by low-class separability and a high dimensionality. This article has tested the hypothesis that – in such conditions – the proposed method could achieve better results than reference algorithms by comparing the R7 architecture with state-of-the-art methods, raw mlp and *Tabnet* architecture. The contributions of this work are the proposition of the new architecture and complete experiments on synthetic and real datasets with the evaluation of the quality and loss achieved by R7 and by reference methods.

Keywords Representation learning · Intermediate distribution · Classification · Difficult data

1 Introduction

Appropriate analysis tools are an increasingly pressing problem in an information-saturated world. Statistical tools dedicated to big data processing or intelligent agents in machine learning help to support the data processing and reasoning, but they can only meet some needs in this area.

Research in the field of classification must be based on labeled data, thanks to which learning models can gain generalization abilities. Jamain and Hand [1], at the beginning of the period of increased interest in learning representations, analyzed what sets are most often used in papers, which led them to a disturbing conclusion – there is a considerable gap in conducting experiments on large and difficult datasets. The reasons for this lack, which they described as an unexpected discovery, are found in the general unavailability of such collections in popular repositories such as UCI [2]. Their conclusions were confirmed ten years later by Shand et al. [3], which proves that this problem has not been solved yet, despite the development of existing repositories and the emergence of new, which today are listed on a par with *University of California, Irvine* (UCI) – Kaggle [4] or *Knowledge Extraction based on Evolutionary Learning* (KEEL) [5].

An additional problem in this regard is the overall low reproducibility of research in machine learning [6]. Many published works need descriptions of the source datasets, which means there is no information about the proper difficulty of the problems considered in them.

It is also worth noting that the definitions of difficult problems differ depending on the context in which the phrase is used. It is common to interpret this way sets with a high degree of class imbalance [7] or problems with a potentionally infinite volume described by data streams [8]. In other cases, explicitly accepted [9] metrics are used as a determinant, allowing numerical determination of a given set's complexity level.

Pawel Ksieniewicz contributed equally to this work.

✉ Weronika Borek-Marciniec
weronika.borek-marciniec@pwr.edu.pl

Pawel Ksieniewicz
pawel.ksieniewicz@pwr.edu.pl

[1] Department of Systems and Computer Networks, Wroclaw University of Science and Technology, Wyb. Wyspiaskiego 27, 50370 Wroclaw, Poland

However, the abovementioned analyses considered criteria more visible at first glance. The initial selection was based on the error rate measure, but later the size of the set, its dimensionality, and the number of considered classes was analyzed. At this stage, the authors pointed out that in the scale of all examined articles, there is a clear gap for multi-cluster sets with high dimensionality and a significant overall volume of observations.

The method proposed in this article is intended for difficult binary classification problems described by large, multidimensional datasets, which are additionally characterized by multiple clusters and low class separability. It is possible to use the existing methods of generating synthetic sets to reflect the existence of the listed features, but real data primarily characterize more complex and diverse distributions. Synthesizers, usually based on generic distributions, will not reflect a similar level of complexity [10], which may be one of the reasons for the gap in research on data that is difficult in a described sense.

The literature in the field offers few methods to deal with all the indicated characteristics simultaneously. It makes it necessary to combine existing algorithms into hybrid solutions, allowing only gradual transformations [11].

Due to its simplicity and relatively low computational complexity, the method most often used in practice to reduce the number of features is *Principal Component Analysis* (pca) [12]. It performs a linear combination of existing attributes. The primary advantage of using this approach is that it removes the redundant influence of correlated problem descriptors. However, when only part of the components is used, pca is not a lossless method in terms of information and the interpretability of the original feature space. Features lose their original meanings after transformation, which is caused by parameterization and its dependence on the user.

A newer and lossless method is t-sne. It is based on nonlinear transformations that preserve local structures and do not lose global information, such as the number of clusters. In turn, it pays with high computational complexity and thus also long processing time, which is incompatible with the use of this method in effectively reducing the dimensionality of large datasets for purposes other than visualization [13].

Both pca and t-sne are also methods used for problems with low class separability, as their dispersion is an additional effect of compressing the data to a smaller dimension [14]. Note, however, that the second of these methods is nondeterministic. It can cause potential problems during experiments due to the lack of results replication [15]. In addition, the distortions introduced to the geometry of the problem here are specific to the set for which the analysis is carried out, which increases the risk of overfitting the model by distorting the original representation of the problem space.

Decreasing the number of samples is solved by resampling. Its simplest form assumes removing the indicated number of randomly selected patterns from the set. However, there are also more complex procedures, with particular emphasis on those that perform stratification, so as not to violate the prior probability of the problem [16].

The construction of the architecture proposed in this article does not require separate transformations to reduce the cardinality and dimensionality or increase the separability of classes. It replaces them with a unitary transformation of the original problem space to an alternate representation defined by a generic distribution of the target space. This concept is based on the previously presented cpte method [17], which brings considered problems to the feature space of one of them. It uses the foundations of ensemble and transfer learning to transform one dataset into the feature space of another. However, it is not intended for use with difficult data understood as multidimensional data with low class separability. For this reason, the cpte algorithm was not used as a reference method during experiments.

The solution presented in this paper develops the cpte paradigm, also taking multitasking as a basis to enable models to gain the ability to generalize against several problems simultaneously [18], which naturally reduces the number of learning components needed. In its basic construction, the proposed *Representation 7* method uses the classic *Neural Network* (nn) architecture, which, as demonstrated by the experimental evaluation, cannot effectively extract the problems difficult to the given definition.

A spectrum of multitask learning methods is based on sharing parameters by models, sets transformations, and encoding-decoding procedures [19]. The first and second groups are used less and less because encoders replace them. This phenomenon is related to a common problem in the field – the need to optimize the target model so that it does not discriminate against any of the component tasks.

Procedures based on the use of encoders-decoders allow for the transfer of two sets to be represented in one space [20], thanks to which the discussed problem does not occur. In other cases, methods based on multi-objective optimization are used, for example, Pareto solutions [21].

Representation 7, like encoders, does not require an additional mechanism to balance the importance of problems for the algorithm because it assumes the transformation of the input sets to a homogeneous representation, which will be described in the Section 2.

The three main contributions of this work can be highlighted.

1. The *Representation 7* architecture was proposed. It uses transformations to unify data sets into a common, generically sampled space. The whole procedure is kept in the context of Multi-layer Perceptron (mlp) processing so it can benefit from its advantages.
2. Proposition was evaluated experimentally on synthetic data compared to state-of-the-art methods for dealing

with difficult data. It verifies the proposed architecture's general properties by measuring the algorithms' quality and loss.

3. The hypotheses were tested on real sets characterized by high cardinality, high dimensionality, and low separability of classes. Real datasets are considered more difficult for recognition models because of their irregular distributions.

2 Methods

2.1 Method motivation

The mlp allows for learning with the use of one or more hidden layers, which means that in the case of many hidden layers, this model structurally corresponds to the basic feed-forward architecture of a deep neural network (dnn). However, at the same time, such a standard structure does not assume separating the data transformation step – nowadays most often defined as representation learning [22] – from optimizing the loss function [23]. In practice, this means that training mlp – regardless of the number of hidden layers – simplifies the definition of the analyzed dataset by minimizing the distance between the decision boundary and incorrectly classified patterns [24].

The method proposed in this paper is based on the assumption that the separation of the extraction stage and – appropriate for the classification task – optimization of the loss function to the given labels may be associated with a significant increase in the quality of the recognition model. Particularly great achievements of this type of strategy are observed for signal data, where the currently most popular approaches introduce neural processing blocks specialized for the task of extracting attributes, i.e., in the form of convolutional layers (cnn) [25]. In the case of classical recognition problems, represented as spatially independent tabular data, they do not find a broader application, and data of this kind is sometimes even referred to as "*the last bastion not yet conquered by deep learning*" [26].

This paper proposes the *Representation 7* (R7) architecture. Its basic assumption is to separate the available memory of the neural model into processing blocks dedicated to the subprocesses of the recognition procedure, which are connected through an intermediate representation. Such internal task-specialization of layers, combined with the implicit propagation of an optimized representation, sampled by more than one problem in each iteration of updating the model weights, may allow both to consider many problems simultaneously and to obtain better results than with a solid model. According to the authors' assumptions, this difference should be observable mainly for the so-called difficult data, especially problems with high class overlapping, where the proper transformation of space ceases to be just a trivial task of dimensionality reduction [27]. This hypothesis is the basis for the conducted experimental evaluation.

The default solution for multidimensional tabular problems is to conduct preliminary attributes engineering, selecting or extracting the features with the greatest predictive potential [28]. In the case of selection methods, for example, based on the *Chi-squared* (Chi2) test or *Analysis of Variance* (ANOVA), the attributes with the most significant dependence or covariance with the class label are identified in the defined problem space. It does not violate the original interpretation of the factors of the set of observations. However, it does not allow a mutual combination of independent attributes, which makes such a strategy highly inefficient for attributes with low linear separability of classes [29]. In the case of extraction methods, for example, based on the pca and its derivatives, or the t-sne algorithm, the procedure looks for a set of mutually orthogonal projections of the original problem attributes with a maximized variance [30] or a neighborly embedded manifolds based on a stochastic process [13]. Such approaches are often agnostic methods, independent of the given problem labeling. In this case, a common alternative is using autoencoders that allow learning an effective, low-dimensional representation of the original problem by building a neural network reproducing its instances [31].

However, each approach has limitations, which do not allow for building an adequate representation of problems described by a high-dimensional set of factors with deficient but strongly independent predictive ability. To illustrate extraction problems, let us assume there is a binary classification problem sampled over a thousand instances with fifty independent attributes and a controllable scale of class-separability. For additional difficulty and to prevent a simple linear separation with a margin of error, each class is built on five clusters of objects. Thanks to the Madelon generator proposed by Guyon, an appropriate illustrative problem can be obtained, where the separability of attributes is configurable by the class_sep parameter [32].

Figure 1 presents such problems. Six columns are consecutive values of class_sep decreasing linearly from 2.0 (high separability) to 0.001 (very low separability). The consecutive rows of the figure show the two-dimensional representations of the problem obtained by:

ANOVA Analysis of variance – selected two best features from the set based on the analysis of their covariance with the class label.

PCA Principal Components Analysis – presenting the two components that explain the largest percentage of variance in the problem.

SAE Shallow Autoencoder – presenting the result of the transformation of a shallow encoder (with an

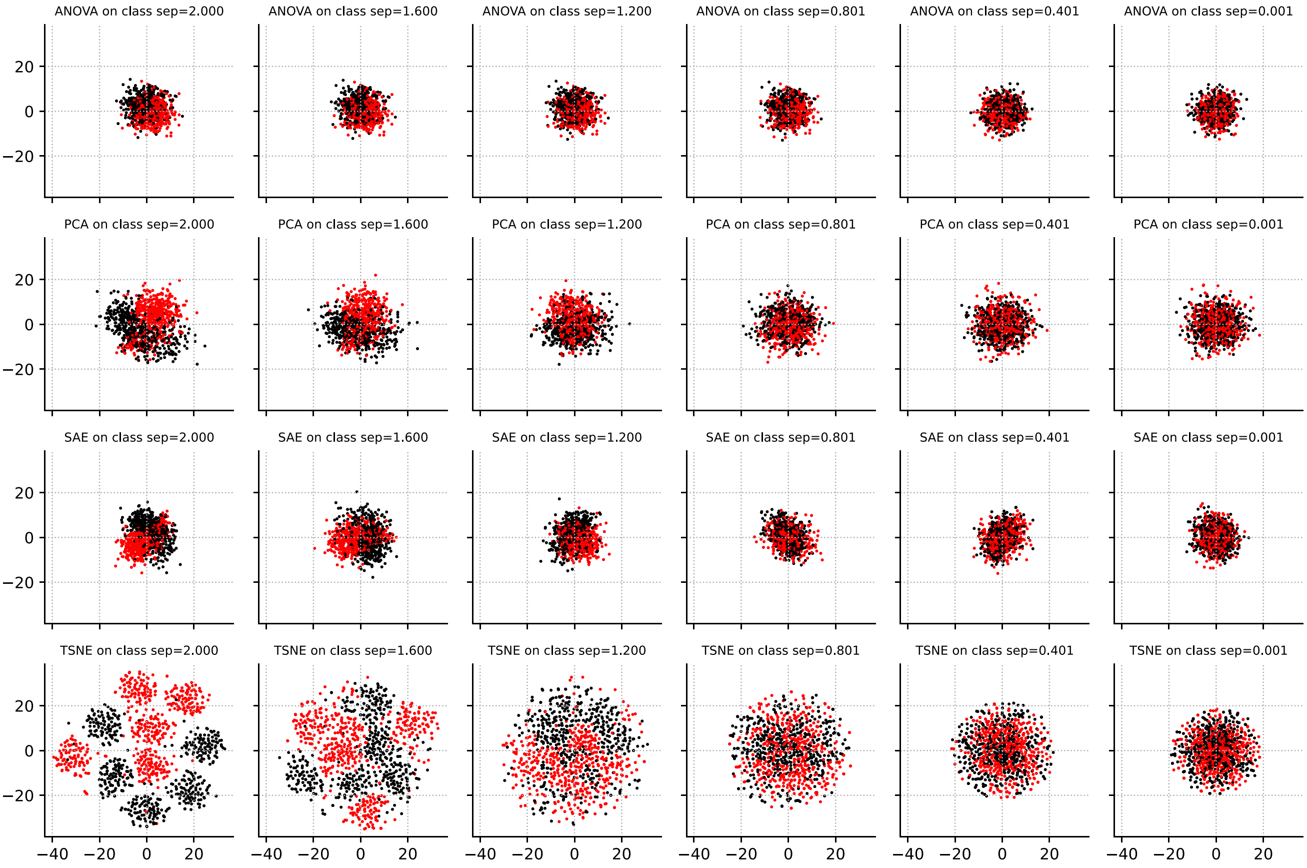


Fig. 1 Reducing the dimensionality of difficult problems using state-of-the-art methods

additional hidden layer consisting of 100 neurons), whose task was to represent the original problem in two dimensions.

t-SNE T-distributed Stochastic Neighbor Embedding – demonstrating the effect of embedding a problem into two dimensions using the t-sne procedure.

As can be seen, the best-differentiating factors, according to the ANOVA test with high dimensionality and a large number of clusters, do not allow for a low-dimensional representation enabling linear separation of classes without a significant margin of error, even in the case of a problem with relatively large size of the hypercube. Pca in this task allows for a significant improvement in the efficiency of the representation, which quickly loses its potential along with the decrease in the separability. Similar, and sometimes even identical, results are achieved by autoencoders, which additionally – due to the random specificity of neural networks – do not constitute a deterministic approach to building a representation. The best one here is t-sne, which, due to constructing a space that only locally reflects Euclidean, is not subject to the same geometrical restrictions as other transformations. It allows a good separation of classes and a proper identification of clusters with a large size of the hypercube forming the Madelon set. Nevertheless, even this approach is highly ineffective when its size rapidly decreases, also building a representation that prevents correct recognition of a defined problem.

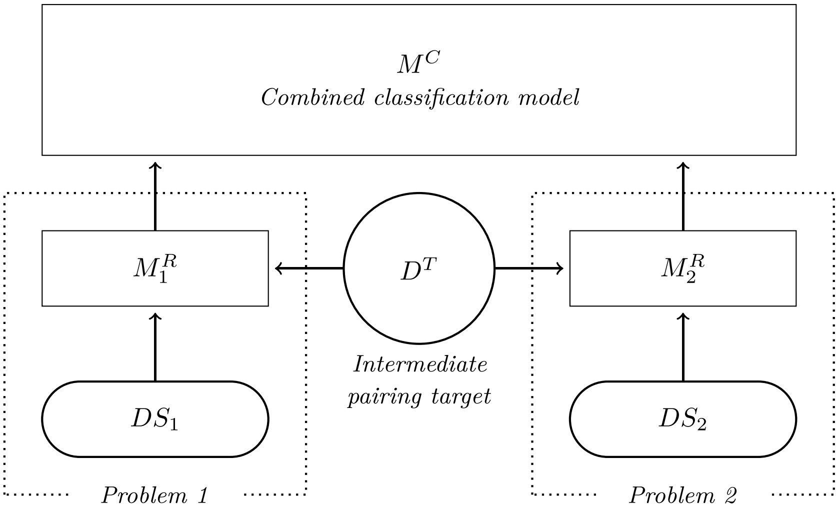
2.2 Representation 7

The R7 architecture proposed in this paper breaks down the available depth of the neural network, for each considered problem, into two blocks of constant width. For illustrative and experimental purposes, the R7 version combining two recognition problems in a coherent architecture is used here, but extending it to larger structures analogically is permissible.

Each recognition problem passed to the R7 model is processed by two logical blocks. The first one, M^R, is a transformation tool to a shared domain of problems, interpreted by the second – M^C – which already solves the classification task. The M^C block is an element shared by all problems in the unified domain, which in the illustrated example Fig. 2 means that it must accept both M_1^R and M_2^R outputs.

In order to allow the M^C block to operate on the outputs of all given M_i^R blocks, two conditions must be met. The first, relatively simple to ensure, is the same dimensionality of

Fig. 2 Schema of the Representation 7 architecture



the output layers of the M_i^R blocks. These blocks are implemented as regression models with a joint output width but admissibly different input width. Thanks to that, both problems can be interpreted together, despite potentially different dimensionality, by ensuring a constant representation width.

Theoretically, it would be possible to apply the autoencoders of both problems to common dimensionality here. However, more is needed to ensure the fulfillment of the second condition – a common definition of the transformed problem. The coexistence of two or more independent definitions of a concept in one space could lead – in the vast majority of practical cases – to the mutual overlapping of classes, significantly hindering the combined problem and thus leading to a significant degeneration of the mixed model concerning separate dedicated solutions. To avoid leading to such a phenomenon, the R7 model introduces an element ordering the transformations carried out in parallel in the M_i^R blocks, in the form of *intermediate pairing target*, which in the illustration is represented as the D^T block.

The D^T block is defined by a simple, generic distribution with a given number of dimensions (*td*), which determines the intermediate problem space representation size. For the cases of binary classification considered in this paper, two clusters distributed co-distantly on the diagonal of feature space were assumed. A parameter equivalent sets the distance from the origin of the coordinate system to the expected value with a constant standard deviation. At each optimization step of the M_i^R models, it is possible to sample from such a distribution a set with prior probability defined by the class distribution of the underlying problem DS_i. This results in a synthetic intermediate set DS_i^T.

Then, the DS_i set is transformed by the M_i^R model, which builds the resulting set M_i^R – represented in the same dimensionality as the D^T distribution. Its instances are further paired with DS_i^T using the matching rule *intermediate pairing target R*(), which orders DS_i^T to $DS_i^{T}{'}$, whose labels thus match labels DS_i. This introduces a regularization factor for all M_i^R transformation models to target the same generic distribution, mitigating the problems that using an autoencoder would cause here. This approach makes it possible to perform the M_i^R optimization step for each problem, taking DS_i as input and $DS_i^{T}{'}$ as the target. After this operation, the M^C model is trained on all concatenated representations of DS_i'.

For a complete picture of how the method works, looking at the matching rule $R()$ is necessary. There are three strategies based on randomness available. The simplest method, also used in the previously mentioned cpte algorithm, is random sampling (RAN). The other two approaches are its modifications, in which the noise generated from the normal distribution (RND) or generated from the target set distribution (RPD) is added to the patterns. In the case of RPD, distribution information is obtained from KDE's kernel density estimator.

In general, the matching rule $R()$ can be described in the three simple steps as follows:

1. Extract the classes of the source problem DS_i and their counts.
2. For each sample in DS_i draw a sample from target set D^T which belongs to the same class.
3. If using RND or RPD matching rule, add noise generated from normal distribution or problem distribution respectively to samples in the transformed set DS_i^T.

After executing the procedure for both DS_i problems, they are concentrated to the common set.

The usage of the matching rule is a part of the R7 architecture. The whole processing procedure described above can be generalized to the following steps:

1. Assume:
   - a finite set of source recognition problems $\mathcal{DS} = \{DS_1, DS_2, \ldots, DS_n\}$,
   - dimensionality of the intermediate representation *td*,

- intermediate distribution D^T with number of dimensions *td*,
- matching rule of intermediate pairing target $R()$,
- parametrization of the base model of the neural network cc.

2. Initialize the model weights for M^C and for the models set $\mathcal{M}^{\mathcal{R}} = \left\{M_1^R, M_2^R, \dots, M_n^R\right\}$ with the configuration cc.
3. Until the stop condition of all models occurs:

   (a) Initialize empty set $\mathcal{DS}' = \{\}$.

   (b) For each model M_i^R from the set $\mathcal{M}^{\mathcal{R}}$:

      (i) Sample the D^T intermediate distribution into DS_i^T, generating cases with the same cardinality and equal prior distribution as DS_i.

      (ii) Propagate the objects of DS_i through the model M_i^R to DS_i set and append it to $\mathcal{DS}'$ set.

      (iii) Using the matching rule $R()$ and the output set DS_i, order the set DS_i^T to the set $DS_i^{T\prime}$, where $DS_i^{T\prime} = R\left(DS_i^T, DS_i'\right)$.

      (iv) Update the weights of the M_i^R model with DS_i as its input and $DS_i^{T\prime}$ as its target.

   (c) Update the weights of the M^C model, taking as its input the concatenation of the sets contained in $\mathcal{DS}'$.

For the method, a constant and common configuration of all M^R models and the M^C model was assumed, excluding only the size of the input and output layers, where the input layers of the M^R models depend on the dimensionality of the $\mathcal{DS}$, their output layers and the input layer of the M^C model depend on the dimensionality of the intermediate representation. The M^C output layer depends on the number of problem classes. The description assumes only binary recognition problems.

3 Experiments

The experiments were implemented using the *Python* language, relying on the *scikit-learn* library dedicated to tabular data [33] and on *PyTorch* for deep learning experiments. Cross-validation 5×2 was used to ensure adequate reliability and repeatability of the results. The quality assessment metric was balanced_accuracy_score. The code of the method, experiments, and their analysis can be found on the GitHub repository.[1]

For the first three experiments, the Multi-layer Perceptron was used as the base model of the neural network in implementing the method. Therefore, any R7 results are also compared to those obtained by the raw, unmodified mlp classifier separately for both source datasets.

The number of layers and neurons in methods – R7 and mlp – was selected, so their architecture was identical regarding memory consumption. Therefore, mlp consists of four layers of 100 neurons. The method proposed in this article consists of transformation and decision blocks, each containing two layers of 100 neurons.

3.1 Datasets

The Madelon procedure generated the synthetic datasets for the first and second experiments [32] implemented in the *scikit-learn* package. It allowed appropriate parameter configuration – the size of the set, its dimensionality, the number of clusters and classes, and the level of their separability.

The used real dataset is *concept-metafeatures* dataset from Kaggle repository,[2] which contains 5000 samples described by 118 attributes. For the needs of analysis, the feature dispersion was conducted, introducing eight variations of this set with a linear increase of noise, reducing the linear separability of its classes.

3.2 Experiment 1 – reference methods

3.2.1 The aim of the experiment

In the first experiment, the performance of the proposed algorithm was compared with the reference methods: raw mlp and mlp with pca preprocessing. The source problems were two synthetic datasets with 2000 samples, 50 dimensions, a variable number of clusters in each class (from 1 to 5), and a variable class separability parameter (five values from the range from 0.01 to 1 were selected). In order to better visualize the differences between the analyzed solutions, the experiments were also carried out for the low- and high-dimensional intermediate distribution. In the first part of the research, the number of intermediate set patterns was twice the samples of a single source set and the exact width of the intermediate representation as in the source set. In the second part – a comparison of the efficiency of the solution for transforming problems to lower dimensionality – the width of the intermediate representation was always 1. Ten repetitions were carried out for each variant to ensure a statistically correct verification of the hypotheses.

3.2.2 Results evaluation

The proposed method applies to problems with high complexity in terms of a large number of features and low separability of classes. Therefore it was decided to compare its

[1] https://github.com/w4k2/r7.

[2] https://www.kaggle.com/datasets/joannakomorniczak/concept-metafeatures-dataset.

operation with the existing state-of-the-art tabular feature extraction algorithm – pca. It was chosen because it is the best standard solution that does not rely on manifolds. In addition, unmodified, raw mlp was used as a reference.

The pca configuration assumed the preservation of components describing at least 70% of the variance after extraction. Then mlp with four layers with one hundred neurons each was used in the same configuration as the raw classifier. This decision was motivated by the fact that pca, despite the simplification of the problem in terms of the number of features and the level of separability, is limited to only a single layer of transformations, which makes it impossible to build non-linear geometric structures of the decision boundary.

The flows of accuracy and losses are shown in Figs. 3 and 4 for high dimensionality of the target set, and in Figs. 5 and 6 for low dimensionality. On all, the blue line corresponds to the R7 method, the red line – raw mlp, and the green line – mlp with pca preprocessing. Each column corresponds to one source problem.

Line intensities refer to the number of clusters used in a given experiment (for 4 and 6 plots) or to the class separability (for 3 and 5). In order to enable the comparison against one factor at the same time and the presentation of the results in the form of waveform graphs, in the first case averaging – flattening – of accuracy and losses concerning separability was made, and in the second – concerning the number of clusters.

The first observation from the visualization is that for both low and high dimensionality of the problem, the flow of accuracy and loss takes a similar shape. The main difference, however, is the maximum threshold of achieved results for each variant varying up to about 5%. It can also be observed that although R7 performs better than the reference methods in all cases, the difference between it and raw mlp is much more evident when using higher dimensionality in both – cluster count and separability – analysis.

It is also worth paying attention to figures showing classifier losses on a logarithmic scale, which for mlp and mlp with pca preprocessing do not change noticeably and reach stability earlier than the R7 loss. They remain at a similar level for all checked parameters, proving that the network has already converged at the lower discriminative level in each case. The R7 architecture allows it to achieve better results in the indicated metric.

Observing heatmaps can further confirm this conclusion (Fig. 7). Each cell shows the result for a specific configuration – the number of clusters should be read from the vertical axis and the separability coefficient from the horizontal axis. Like in the flows, the top row shows a balanced accuracy score, and the bottom row – loss. The letter M stands for raw mlp, P is mlp with pca preprocessing, and R is the R7 method proposed in this paper.

In addition, color coding was used, where shades of purple indicate the advantage of R7 results over other methods. The darker the color, the more significant the difference in favor of the proposed method. The dependence is also more apparent here, already visible in the flows – the expected decrease in the quality of all models with the increasing level of problem complexity can be noted. The difference between R7 and raw mlp is more significant than between R7 and mlp with pca, which means that the actual use of preprocessing with this method harms accuracy. Pca evenly distributes the potentials on all attributes, so eliminating even one of them causes information loss and thus makes the problem harder, which is the reason for lowering the maximum achievable recognition quality.

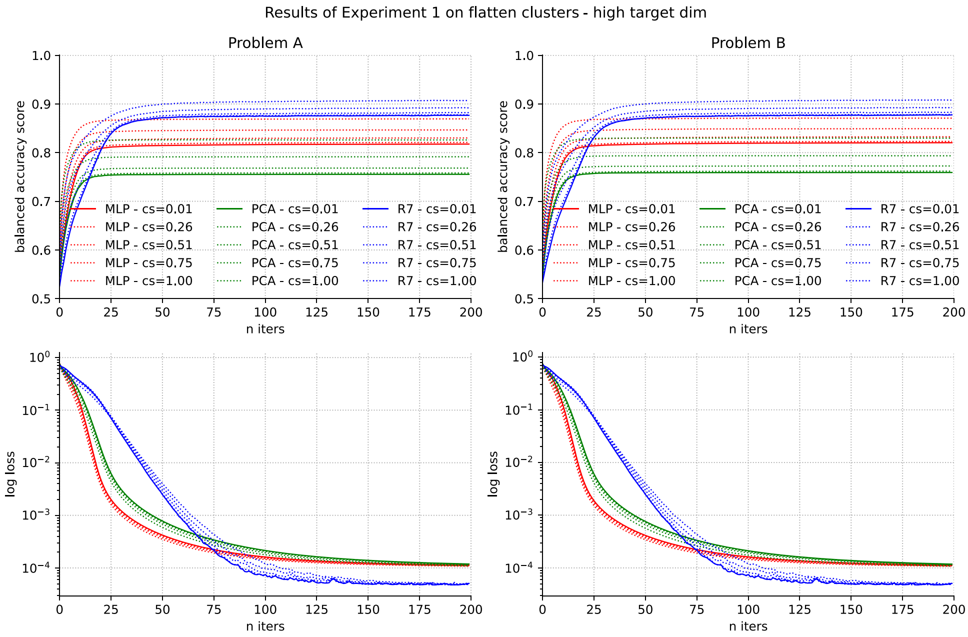
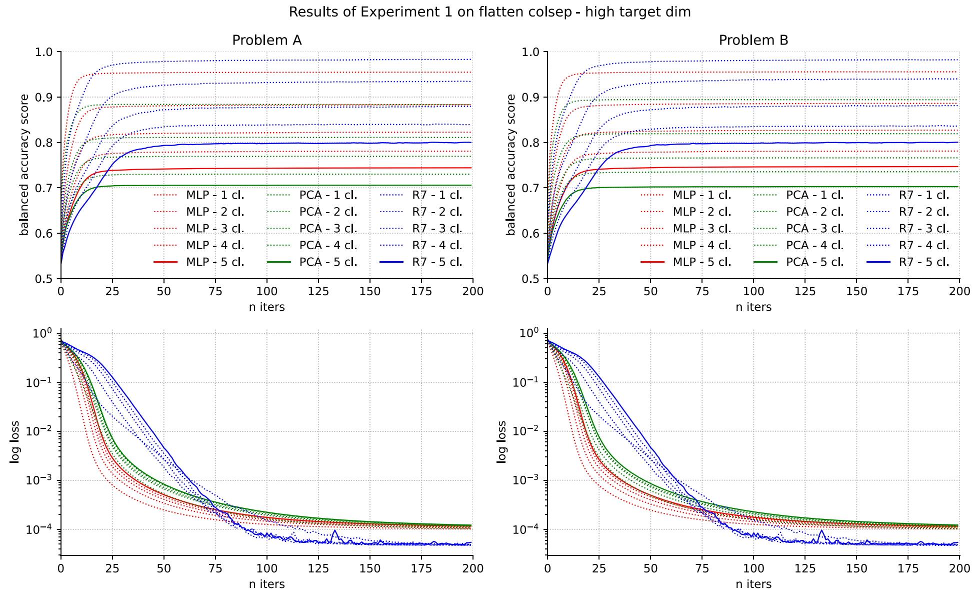


Fig. 3 *Experiment 1* – balanced accuracy score and loss shown on cross-section of classes separability for high target dim

Fig. 4 *Experiment 1* – balanced accuracy score and loss shown on cross-section of number of clusters for high target dim



The heatmap also shows that the number of clusters has a more significant impact on the difficulty of the set than the class separability, which can be seen by the degree by which the color intensity changes in individual columns and rows for both source problems. The same is also indicated by the heatmaps in the bottom row, where the green color means that the R7 achieves less loss than the other two methods. Increasing color brightness is associated with increasing this difference. The R7 minimizes loss better, especially when using more complex problems.

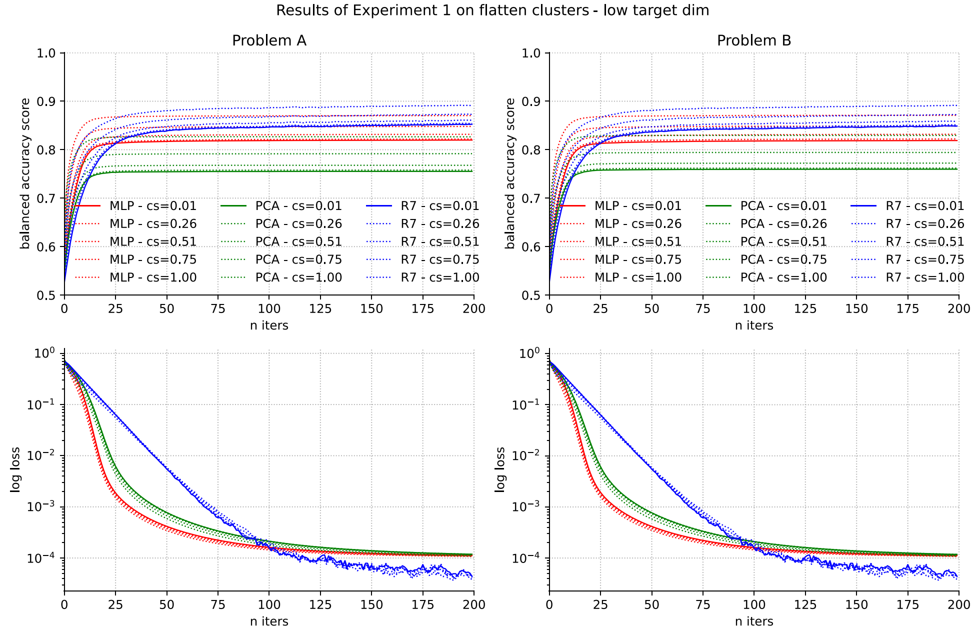
The observations described above show that mapping the mlp architecture in the way that R7 does can lead to better results than using simple state-of-the-art methods, and its potential increases with increasing the problem's difficulty.

3.3 Experiment 2 – cardinality and dimensionality of the source datasets

3.3.1 The aim of the experiment

The second experiment aimed to review the cardinality and dimensionality as the parameters of the generated problems concerning mlp as a reference method. In each of the ten replications, two source datasets with fifty attributes, five clusters per class, and constant, low class separability of 0.1 were generated. The number of samples was consecutively 1000, 2500, and 5000. The transient representation in each repetition contained twice the patterns of the source problem, and two dimensionalities were checked – 25 and 50.

Fig. 5 *Experiment 1* – balanced accuracy score and loss shown on cross-section of classes separability for low target dim



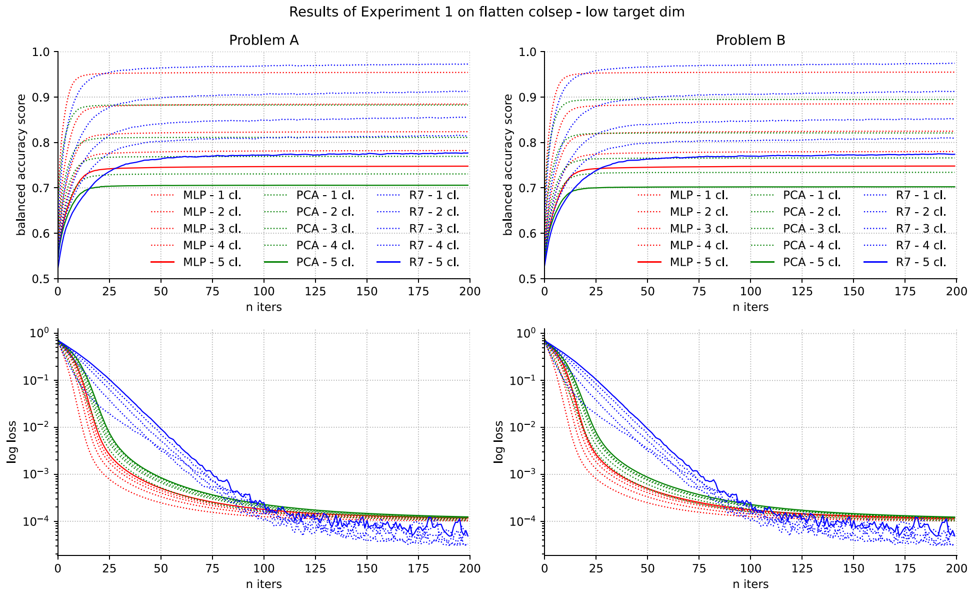


Fig. 6 *Experiment 1* – balanced accuracy score and loss shown on cross-section of number of clusters for low target dim

3.3.2 Results evaluation

The balanced accuracy score results are shown in Fig. 8, where each column refers to a different size of the set – 1000, 2500, and 5000 samples, respectively – and each row to a different dimensionality – 25 and 50 attributes. One of the source sets is presented with a solid line, the other with a dashed line, but what can be observed immediately, the results achieved for them do not differ significantly. The course of accuracy is presented for the test set over successive iterations after averaging against replication and cross-validation folds.

Losses of transformation blocks (Fig. 9) and classifiers (Fig. 10) were analyzed similarly. A logarithmic scale was used for the presentation.

In the first of the figures, the flows took a similar shape. From this fact, the first conclusion can be drawn – models from $\mathcal{M}^{\mathcal{R}}$ training are similar in estimator loss, regardless of the set size parameters. Moreover, there is no distinction between training the models from $\mathcal{M}^{\mathcal{R}}$ for both source problems – solid and dashed lines are indistinguishable.

Greater diversity can be seen in the case of classifier losses, although the mutual relations of the results achieved by individual R7 variants and by mlp are similar. Changes are most easily observed for a random strategy with noise with the characteristics of the transitional representation where the period of fluctuations, their amplitude, and the initial epoch change. The smoothest fluctuations are for a thousand samples and only for this variant, and for 50 dimensions, it can be observed that the loss stabilizes around the 175th epoch. The flow of losses of the remaining strategies and mlp is smooth, and the random and random strategies with normal noise coincide during most of the iterations. It is also seen that although the losses of R7 start to decline at a later epoch than mlp, they eventually reach lower values.

This observation is consistent with the presented accuracy, which is approximately 5 % higher for the R7 than for the reference method in each case. While for 1000 samples, it achieves an accuracy of 70%. For 2500 it is already 80%, and for 5000–85%. With the increase in the dataset size, the random strategy with noise with the characteristics of the transitional representation stands out more and more, slightly outperforming the quality of the other method variants in the last case.

On the other hand, there is no significant difference when changing the dimensionality of the transitional representation – for both 25 and 50 attributes, the accuracy remains at a similar level. Considering this and the classifier loss flows, it can be concluded that the proposed method has already saturated its predictive capabilities, reaching suboptimal values, and further increasing the number of features will not increase the model's accuracy.

From all the above observations, it can be concluded that, according to the adopted quality measure, R7 performs better thanmlp for difficult data. As the number of samples increases, the mlp loss gets closer to the R7 loss, while the proposed method still achieves significantly better accuracy, as shown in the illustrative examples.

3.4 Experiment 3 – real datasets

3.4.1 The aim of the experiment

The third experiment tested the performance of the R7 algorithm for real data. Mlp was used as a reference, as in the other cases. The dimensionality of the intermediate representation was set to 25, and its cardinality is the sum of the

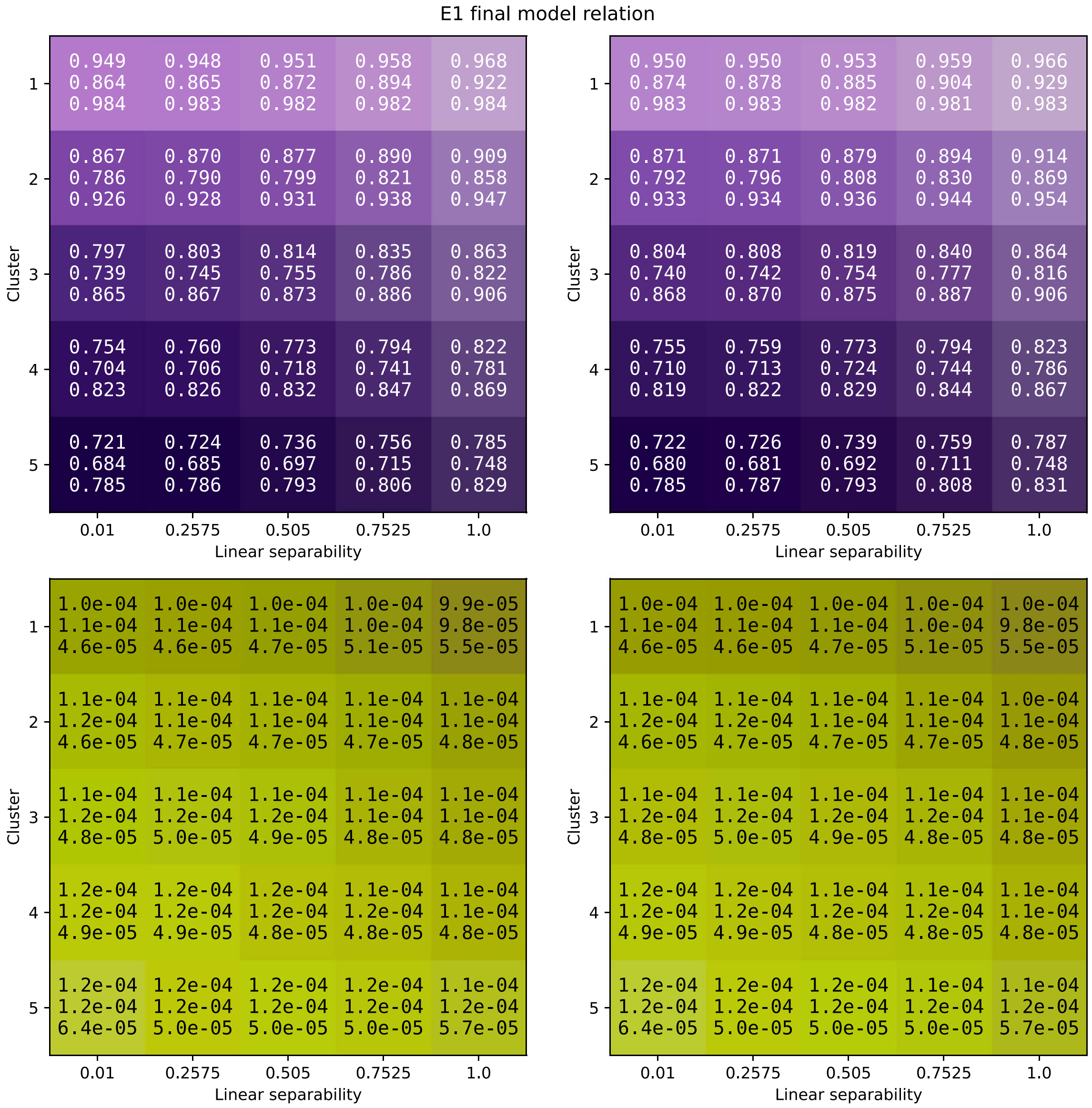


Fig. 7 *Experiment 1* – heatmap of balanced accuracy scores and loss for all variants of class separability and number of clusters

cardinality of the source sets. Source problems and obtaining them are described in detail in Subsection 3.1.

3.4.2 Results evaluation

According to the conclusions drawn from the second experiment, the transient representation for R7 consisted of a number of samples equal to the sum of the samples of the source problems and 25 dimensions. All three pairing strategies have been compared with mlp.

The results of the experiment are presented in Fig. 11, where the balanced accuracy obtained by the tested methods for both the training set (left column) and the test set (right column) can be seen. The horizontal axis shows the dispersion of the information from the source sets – zero corresponds to the unmodified original set and seven to the highest level of problem deconstruction. The error bars are the recorded standard deviation, and the methods are color-coded in the legend, where RND stands for normal noise and RPD – for problem noise.

Thanks to the accuracy plot for the training set, the susceptibility of particular methods to overfitting can be discussed. It can be seen that mlp consistently achieves results above 90%, although the quality on the test set decreases as the problem's difficulty increases. Similar results are obtained for the R7 without noise and with the noise of normal characteristics, where both fall below the threshold in only one case. The behavior of R7 with noise generated from

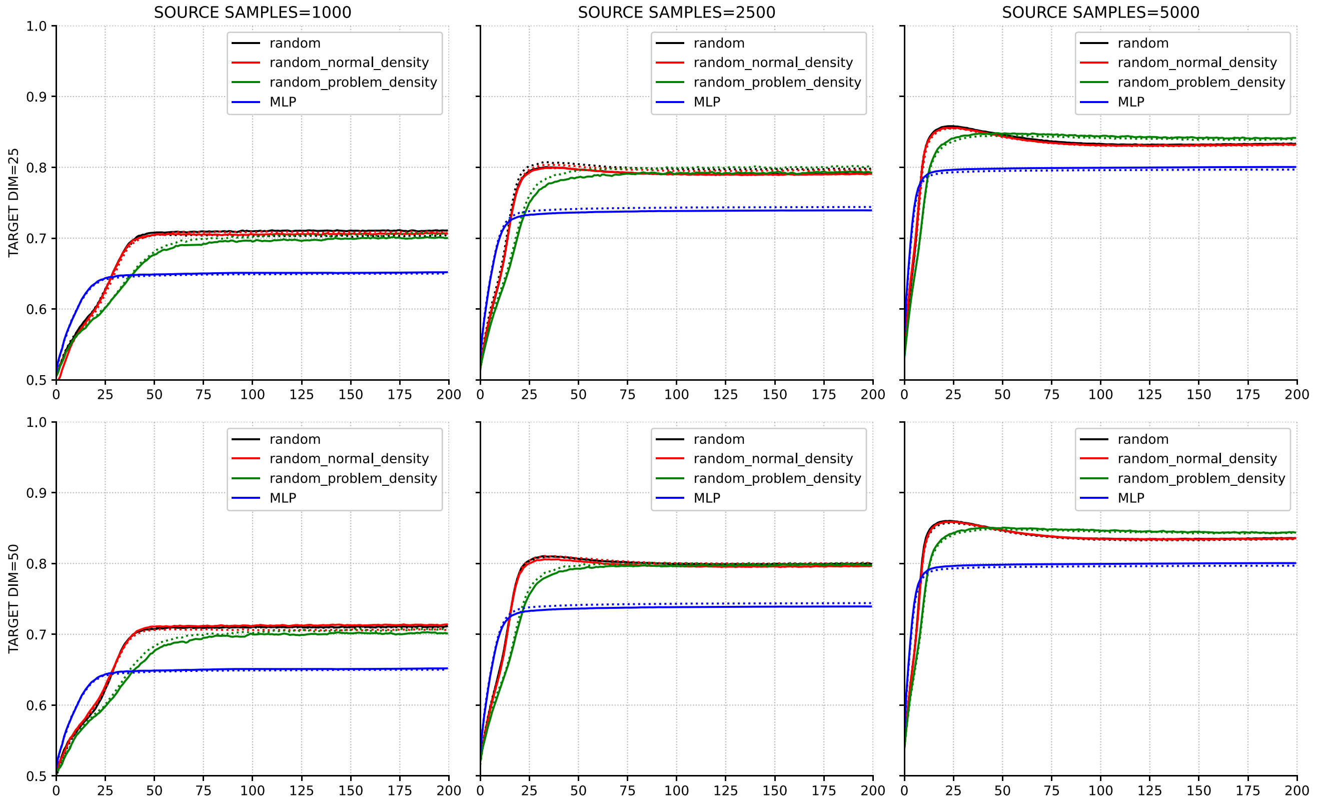


Fig. 8 *Experiment 2* – balanced accuracy score for all variants of samples and dimensionality

the distribution of the intermediate problem differs from this trend. At the same time, it achieves the best results for the test set – at the first three levels, the accuracy is close to the other two strategies, but later it starts to stand out.

Meanwhile, mlp, although it starts with a similar level of accuracy (about 85%) as the other methods, in each subsequent case, achieves significantly worse results for the test set when the quality of the training is still at the highest level. Moreover, this difference is aggravated for the R7 with problem-distributed noise.

From the above observations, this strategy has the most potential when applied to real data. It can be caused by the fact that the distributions of synthetically generated problems, although far from typical distributions, such as normal or uniform, still preserve some of their properties. In contrast, the actual distributions of features are more complicated. Disturbing the coefficients with transitional representation noise can partially compensate for these differences.

In turn, the advantage of the proposed method over raw mlp is related to the separation of the set transformation stage from the prediction stage, which strengthens the generalization ability of the model, and at the same time, tends to reduce its susceptibility to overfitting, avoiding this common problem with neural networks.

3.5 Experiment 4 – deep learning

3.5.1 The aim of the experiment

In the fourth experiment, the performance of the R7 strategy was compared to the state-of-the-art method of deep learning for tabular data – the TabNet classifier [34]. For this purpose, an R7 implementation using the PyTorch library was deployed,[3] where each block – both transformation and decision – consists of two hidden layers with 114 neurons each. Experiments were conducted on synthetic sets, where the number of objects (5000) and class separability (0.7) were not altered. Parameters such as (a) the number of clusters per class (1, 5, and 10), (b) the number of attributes of the input sets (from 10 to 1000 dimensions on a logarithmic scale and 32 quants, all of which were informative features) were investigated. In each iteration, R7 considered five binary source datasets, and the transitional representation assumed a dimensionality of 2 and a number of objects equal to the sum of all learning set objects. Different learning rate configurations for the neural networks were also tested – $1e-3$ and $1e-2$ using 250 epochs for R7 and the default number of epochs for TabNet.

[3] https://github.com/w4k2/rockets.

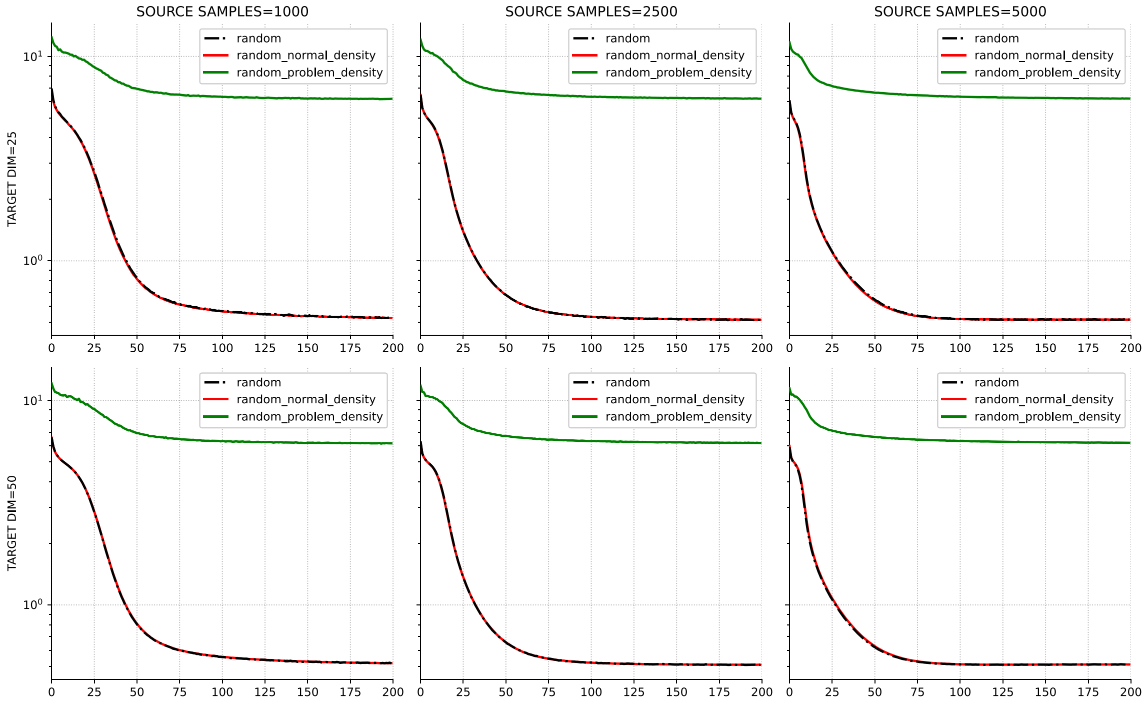


Fig. 9 *Experiment 2* – regression loss for all variants of samples and dimensionality

3.5.2 Results evaluation

Figure 12 presents the balanced accuracy score for both tested methods. The learning parameters – learning rate, features, and number of clusters per class – can be read from the header of each graph. The left column shows a relative comparison of the learning curves, where the black line refers to TabNet and the red lines to R7 – one is the accuracy comparison for the training and test sets, and the other, located on the diagonal, is directly the quality of the test set. The right column, in

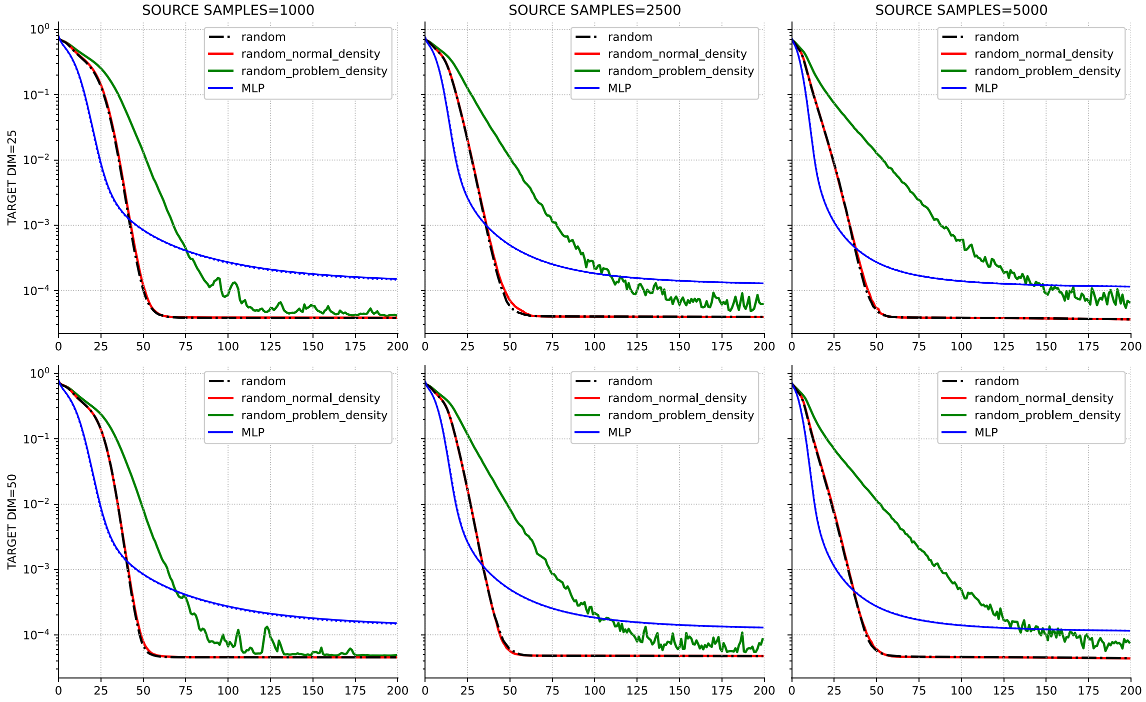


Fig. 10 *Experiment 2* – classifier loss for all variants of samples and dimensionality

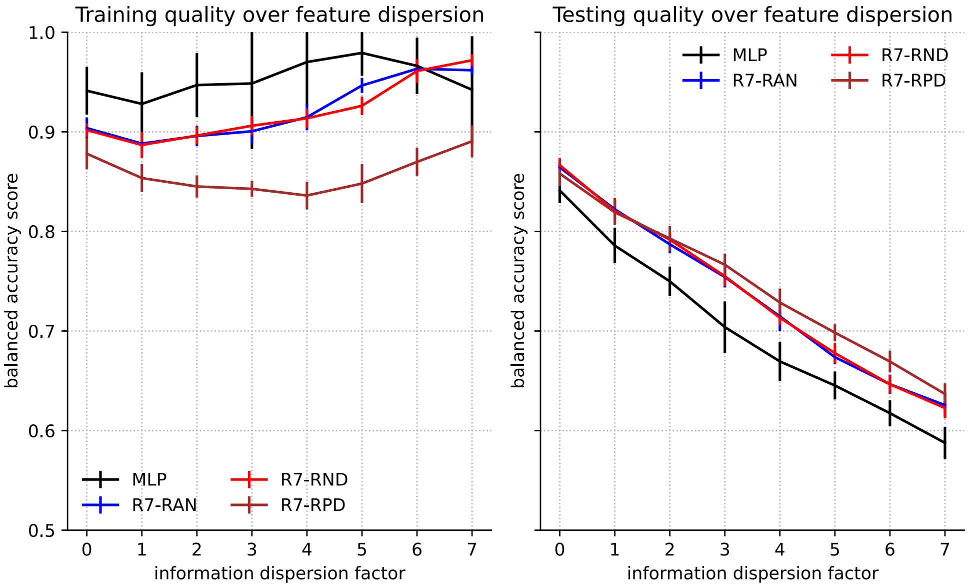


Fig. 11 *Experiment 3* – classifier accuracy for all R7 strategies and mlp

turn, shows the learning curve. In addition, Fig. 13 presents the differences between the accuracy of R7 and TabNet. The advantage in favor of the proposed method is marked in red, and the advantage in favor of the reference method – in blue.

As can be observed, the TabNet model has the advantage for sets with lower dimensionality. However, when the number of features exceeds 40, better accuracy is achieved by the R7 architecture proposed in this article, regardless of the number of clusters per class.

An interesting conclusion about the method can be drawn from observing the change in the quality difference between the models depending on the learning rate used (middle column of Fig. 13) – for a value of 0.001 the advantage of TabNet over R7 is 39.7%, while for a value of 0.01 it drops to 7.3%. The reason for this difference is that a large learning rate ignores geometric nuances in the data distribution. However, as the number of attributes increases, these nuances become less and less critical – which is taken into account by the R7 architecture, which intentionally has a mechanism to ignore them by seeking a near-threshold separation of input factors.

Such an approach will not work in low dimensionality, where geometric nuances are critical. However, the saturation of red with an increasing number of attributes shows that for problems with more complex characteristics, the proposed rule and processing structure – strictly by its assumptions – is resistant to the already redundant search for category shapes and instead treats the set as a collection of factors to be sorted out.

In addition, it can be seen on the runs of quality change in successive epochs that the accuracy of the TabNet method remains almost constant for both the training and test sets, while for the specified experimental conditions and with the indicated number of initial epochs, TabNet had about four times longer learning time than R7.

Based on the study, it can be observed that although TabNet as a pre-trained model performs significantly better for low-dimensional data (at best, by 40%), as the problem's dimensionality increases, R7 gains the advantage (at best, by 26%). It should also be taken into account that for the architecture proposed in this article, this result reflects the simultaneous processing of five data sets while TabNet processes one of them. Thus, the experiment has given credence to the hypothesis that R7 performs better when processing sets that are difficult in terms of a large number of features and low class separability.

4 Conclusions

This paper presents a new method – R7 – that unifies datasets into a shared space in the context of neural network optimization. Its performance for difficult data represented by multidimensional sets with low class separability was tested. Its performance was compared (a) to the raw mlp architecture with pca extraction, as a state-of-the-art method for reducing the feature space, (b) to the raw mlp, as this architecture is the foundation of the R7 implementations and (c) to the TabNet network, as the state-of-the-art method in deep learning processing for tabular data. The experiments were conducted on synthetic and real data with increasing information dispersion.

The research showed that the R7 based on mlp outperformed the reference methods regarding accuracy and other factors, such as loss function and susceptibility to overfitting. This effect is achieved by separating the set's transformation stage to a simpler feature space from the stage of learning the classifier and by performing those steps simultaneously using standard neural networks. Separation of these stages allows for better specialization of methods of individual neural networks.

A comparative experiment with TabNet, in turn, showed that although the R7 architecture using networks achieves

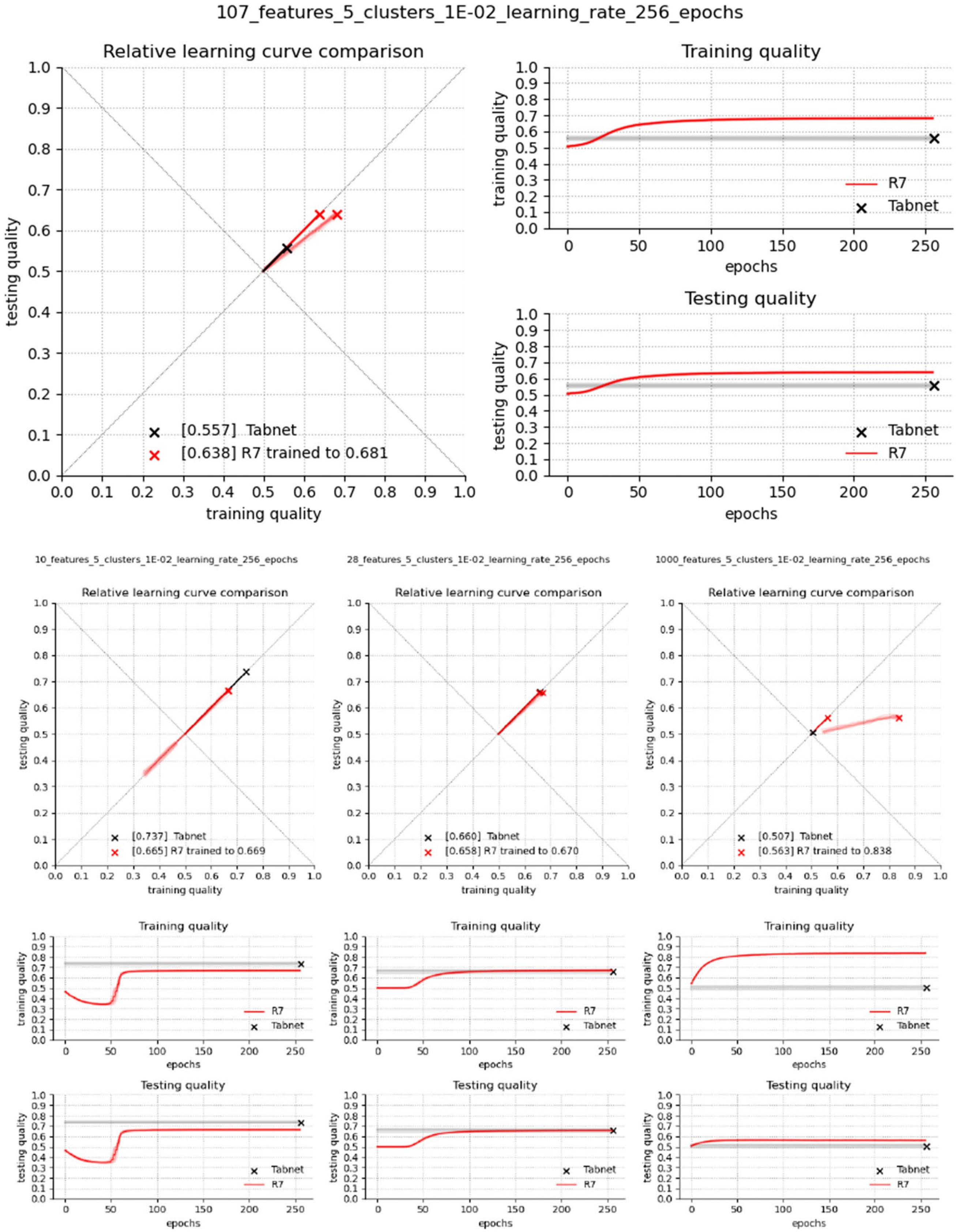


Fig. 12 *Experiment 4* – classification accuracy for R7 and TabNet

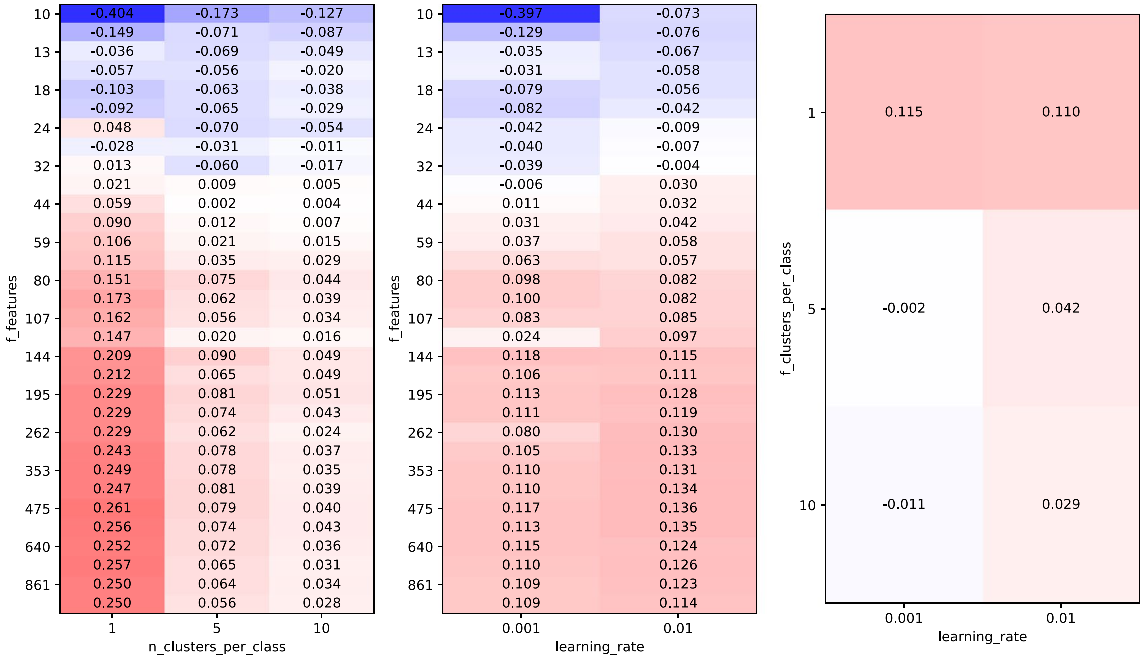


Fig. 13 *Experiment 4* – differences between accuracy of R7 and TabNet

worse quality for low-dimensional data, in difficult problems - as predicted - it is characterized by much better generalization ability, which is caused by less sensitivity to subtle deviations within the data set.

In addition, R7 can be used to analyze many datasets simultaneously, which meets the basic paradigm of the multitasking field and can be a way to reduce the memory complexity of the final model due to the shared classifier block. As part of future works, this thesis should be verified using a broader review of real data, such as multimodal data, in which simultaneous modality analysis can offset the negative effects of separate feature extraction and thus positively affect quality. Using sets with different views (modalities) for the same objects will be interesting.

An area for improving the method is a more detailed analysis of the phenomenon of loss fluctuations for a random strategy with noise generated from the intermediate problem observed in Experiment 2. Their source can be traced to the learning process, where some of the samples are alternately assigned to the class positive and negative – they are close to decision boundary. A closer look at this phenomenon and its source will allow us to introduce corrections in the method, further increasing its potential effectiveness.

Another aspect in which there is still room for improvement in the method is the pairing of samples between the source sets and the intermediate representation. Until now, all strategies are based on random methods with modifications that could be replaced by precise criteria – for example, based on distance. The research can also be extended to using other learning modes currently implemented only on transformed source sets. Including the representation of the transient problem in this representation could improve the recognition quality. Testing the generalization abilities of the model that learns only on the intermediate representation would be an interesting research aspect.

More complex intermediate distributions could also be used. Introducing the aspect of the variability of these distributions and optimizing them with the criterion of class separation may allow solving problems with different characteristics – even those containing an enclosing class.

R7 also has the potential to be used in solving multi-class problems where the source sets contain both equal and different numbers of classes. In such a case, an asymmetric configuration of the transformation and decision blocks could be useful as differentiation in the number of hidden layers would allow for a better adjustment to the problem.

Analysis conducted within the deep learning environment also led to the observation that the initial prediction quality does not always reflect the behavior of the random classifier. The introduction of an element regulating the initial state and its parameters into the architecture could have a positive impact not only on the final quality of the model but also on its learning time. This is an additional field of research, during which the reasons for the decrease in the value of the metric towards negative labels at the beginning of the learning process and the subsequent compensation, which can be seen in Fig. 12, should also be analyzed.

Acknowledgments This work was supported by the statutory funds of the Department of Systems and Computer Networks, Faculty of Information and Communication Technology, Wroclaw University of Science and Technology.

Funding This work was supported by the statutory funds of the Department of Systems and Computer Networks, Faculty of Information and Communication Technology, Wroclaw University of Science and Technology.

Declarations

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of interest The authors declare that they have no conflict of interest.

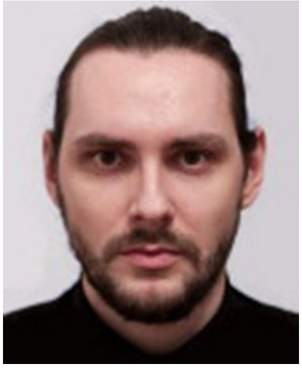
References

1. Jamain A, Hand DJ (2009) Where are the large and difficult datasets? ADAC 3(1):25–38
2. Dua D, Graff C (2017) UCI Machine Learning Repository. https://archive.ics.uci.edu/ml. Accessed 15 Apr 2023
3. Shand C, Allmendinger R, Handl J, Webb A, Keane J (2019) Evolving controllably difficult datasets for clustering. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp 463–471
4. Bojer CS, Meldgaard JP (2021) Kaggle forecasting competitions: An over-looked learning opportunity. Int J Forecast 37(2):587–603
5. Derrac J, Garcia S, Sanchez L, Herrera F (2015) Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. J Mult Valued Logic Soft Comput 17:255–287
6. Heil BJ, Hoffman MM, Markowetz F, Lee S-I, Greene CS, Hicks SC (2021) Reproducibility standards for machine learning in the life sciences. Nat Methods 18(10):1132–1135
7. Komorniczak J, Zyblewski P, Ksieniewicz P (2021) Prior probability estimation in dynamically imbalanced data streams. In: 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, pp 1–7
8. Komorniczak J, Zyblewski P, Ksieniewicz P (2022) Statistical drift detection ensemble for batch processing of data streams. Knowl-Based Syst 252:109380
9. Lorena AC, Garcia LP, Lehmann J, Souto MC, Ho TK (2019) How complex is your classification problem? a survey on measuring classification complexity. ACM Comput Surv (CSUR) 52(5):1–34
10. Assefa SA, Dervovic D, Mahfouz M, Tillman RE, Reddy P, Veloso M (2020) Generating synthetic data in finance: opportunities, challenges and pitfalls. In: Proceedings of the First ACM International Conference on AI in Finance, pp 1–8
11. Ardabili S, Mosavi A, Várkonyi-Kóczy AR (2020) Advances in machine learning modeling reviewing hybrid and ensemble methods. In: International Conference on Global Research and Education, Springer, pp 215–227
12. Hotelling H (1933) Analysis of a complex of statistical variables into principal components. J Educ Psychol 24(6):417
13. Anowar F, Sadaoui S, Selim B (2021) Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne). Comput Sci Rev 40:100378
14. Cardona LAS, Vargas-Cardona HD, Navarro González P, Cardenas Peña DA, Orozco Gutiérrez ÁÁ (2020) Classification of categorical data based on the chi-square dissimilarity and t-sne. Computation 8(4):104
15. Liu C, Gao C, Xia X, Lo D, Grundy J, Yang X (2020) On the replicability and reproducibility of deep learning in software engineering. arXiv preprint arXiv:2006.14244
16. Gerber M, Chopin N, Whiteley N (2019) Negative association, ordering and convergence of resampling methods. Ann Stat 47(4):2236–2260
17. Borek W, Ksieniewicz P (2022) Inductive parallel learning for multiple classification problems. In: 2022 International Joint Conference on Neural Networks (IJCNN), IEEE, pp 1–8
18. Zhang N, Gupta A, Chen Z, Ong Y-S (2021) Evolutionary machine learning with minions: A case study in feature selection. IEEE Trans Evol Comput 26(1):130–144
19. Vandenhende S, Georgoulis S, Van Gansbeke W, Proesmans M, Dai D, Van Gool L (2021) Multi-task learning for dense prediction tasks: A survey. IEEE Trans Pattern Anal Mach Intell 44(7):3614–3633
20. Hu R, Singh A (2021) Unit: Multimodal multitask learning with a unified transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 1439–1449
21. Lin X, Zhen H-L, Li Z, Zhang Q-F, Kwong S (2019) Pareto multitask learning. Advances in neural information processing systems 32
22. Le-Khac PH, Healy G, Smeaton AF (2020) Contrastive representation learning: A framework and review. IEEE Access 8:193907–193934
23. Hinton GE (1990) Connectionist learning procedures, 555–610
24. James G, Witten D, Hastie T, Tibshirani R (2013) An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
25. Li Z, Liu F, Yang W, Peng S, Zhou J (2021) A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Trans Neural Netw Learn Syst 33:6999–7019
26. Kadra A, Lindauer M, Hutter F, Grabocka J (2021) Well-tuned simple nets excel on tabular datasets. Adv Neural Inf Proces Syst 34:23928–23941
27. Espadoto M, Hirata NST, Telea AC (2020) Deep learning multidimensional projections. Inf Vis 19(3):247–269
28. Khalid S, Khalil T, Nasreen S (2014) A survey of feature selection and feature extraction techniques in machine learning. In: 2014 Science and Information Conference, IEEE, pp 372–378
29. Zheng J, Qu H, Li Z, Li L, Tang X, Guo F (2022) A novel autoencoder approach to feature extraction with linear separability for high-dimensional data. PeerJ Comput Sci 8:1061
30. Topolski M (2020) The modified principal component analysis feature extraction method for the task of diagnosing chronic lymphocytic leukemia type b-cll. J Univ Comput Sci 26(6):734–746
31. Sewak M, Sahay SK, Rathore H (2020) An overview of deep learning architecture of deep neural networks and autoencoders. J Comput Theor Nanosci 17(1):182–188

32. Guyon I (2003) Design of experiments of the nips 2003 variable selection benchmark. In: NIPS 2003 Workshop on Feature Extraction and Feature Selection 253:40
33. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12:2825–2830
34. Arik SÖ, Pfister T (2021) Tabnet: Attentive interpretable tabular learning. Proceedings of the AAAI Conference on Artificial Intelligence 35:6679–6687

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Weronika Borek-Marciniec is a PhD student at Wroclaw University of Science and Technology, where she achieved an MSc. degree in 2021. Her research focuses on the problem transformation in pattern recognition tasks, especially in the field of classification.



Paweł Ksieniewicz is an associate professor at Wroclaw University of Science and Technology, where he achieved an M.Sc. degree in 2013, a Ph.D. degree in 2017 and D. Sc. in 2023. His research focuses on the classification of difficult recognition problems represented by imbalanced and drifting data streams, multidimensional data representation, image processing and open set recognition.