Wrapper-based feature selection using Self-adaptive Differential Evolution

Dušan Fister[1[0000−0002−9604−0554]] ⋆, Iztok Fister[2[0000−0002−9964−6957]], Timotej Jagrič[1], Iztok Fister Jr.[2[0000−0002−6418−1272]], and Janez Brest[2[0000−0001−5864−3533]]

[1] University of Maribor, Faculty of Economics and Business, Razlagova 14, SI-2000 Maribor, Slovenia
{dusan.fister1, timotej.jagric}@um.si
[2] University of Maribor, Faculty of Electrical Engineering and Computer Science, Koroška cesta 46, SI-2000 Maribor, Slovenia
{iztok.fister, iztok.fister1, janez.brest}@um.si

Abstract. Knowledge discovery in databases is a comprehensive procedure which enables researchers to explore knowledge and information from raw sample data usefully. Some problems may arise during this procedure, for example the Curse of Dimensionality, where the reduction of database is desired to avoid feature redundancy or irrelevancy. In this paper, we propose a wrapper-based feature selection algorithm, consisting of an artificial neural network and self-adaptive differential evolution optimization algorithm. We test performance of the feature selection algorithm on a case study of bank marketing and show that this feature selection algorithm reduces the size of the database and simultaneously improves prediction performance on the observed problem.

Keywords: data preprocessing · feature selection · self-adaptive differential evolution jDE · NiaPy

1 Introduction

In the era of big data, where more and more data are analyzed, a Data Mining (DM) paradigm [40] has been proposed to stimulate the design and analysis of diverse DM methods to deal with sample data. A wide range of DM methods exist nowadays, from traditional regression analysis to Machine Learning (ML) and symbolic methods. Those can solve problems like classification, regression, clustering, association rule mining and others, in a supervised, unsupervised or reinforcement learning way.

The data are becoming more and more complex. The rise of the Industry 4.0 [19] and the rise of the mobile devices contribute inherently to such trend. As a result, the analysis of data is, according to the Curse of Dimensionality [5] becoming tougher. Typically, the range and volatility of values increases, generalization diminishes and outliers with missing data emerge. In order to avoid

⋆ Corresponding Author: dusan.fister1@um.si

these issues, a data preprocessing step is necessary, which executes the data adjustment procedures to obtain improved predictive abilities of DM methods. Proper data preprocessing step typically tries to decrease or eliminate feature redundancy/irrelevancy and may thus bring lower costs and higher modeling efficiencies. In the following paper, we would like to empirically investigate the practical worth of the data preprocessing step, i.e. what are its benefits, and list its shortcomings.

Literature conceptualizes both the DM and data preprocessing as sub-parts of a more comprehensive Knowledge Discovery in Databases (KDD) procedure [15]. The latter formalizes six phases as shown below:

- problem specification, where the actual problem and the estimated outcome of the KDD are first addressed,
- problem understanding, where the problem tries to become explainable,
- data preprocessing, where the data are prepared, and the complexity is reduced by removing redundant or irrelevant items,
- data mining, where the actual model and task are determined to mine the preprocessed data and to extract discovered knowledge,
- evaluation, where the obtained results are interpreted and
- result exploitation, where the knowledge discovered is visualized for a report to form inference.

The purpose of the mentioned six phases of KDD is to extract hidden patterns, relations and interconnections among data. Data usually consist of explanatory variables (features), i.e., inputs, and the response variables (also target), i.e., outputs. Relations can then be studied for various inference and analysis applications, such as student performance [25], cost reduction [35], satisfying customer expectations [10], healthcare [4], forecasting [7] and others.

The more the interconnections exist in data, the more the redundancies arise. This affects the prediction performance negatively. Feature selection (FS) is a suitable process of eliminating some of the variables, to diminish or avoid redundancy [41]. Three FS methods exist for the supervised tasks [1, 15]: filter-based method, wrapper-based method and embedded-based method. Filter-based method performs the FS separately from the learning algorithm [42], while wrappers do not. They use the learning algorithm to determine the quality of the selected subset [17]. Embedded methods, on the other hand, combine characteristics of the previous two [38]. The following survey outlines many FS applications [8].

Osanaiye et al. [27] show the application of filter-based method for feature elimination in the field of cloud computing. Authors state that number of features can be reduced efficiently from 41 to 13, by using their method. Apolloni et al. [3] claim that their FS methods decrease the number of relevant features for more than 99% on the microarray problem, on the basis of six datasets. Labani et al. [18] design a novel multivariate filter-based method FS for text classification problems, and prove that it can overcome other univariate and multivariate methods.

Mafarja and Mirjalili [24] outline a wrapper-based method using the whale optimization. They use the mutation and crossover operators to increase pop-

ulation diversity and tournament selection to select best individuals. Authors test their FS method on the 18 different datasets. For most of the datasets, they show that their method improves other filter-based FS methods. Al-Tashi et al. [2] deal with combining the hybrid grey wolf optimization (GWO) and particle swarm optimization (PSO) for FS problem. Ramjee and Gamal [31] experiment with wrapper-based FS, where they employ the relevance and redundancy scores. Based on their solution, they report the increase of accuracy from initial 93% to final 99% for the RadioML2016.10b dataset. Lu [23] agrees that data heterogeneity leads to spurious classification and thus proposes the embedded FS method as a remediation. Author uses the synthetic data and three benchmark datasets to confirm that his solution bears fruit, compared to traditional embedded-, or filter-based methods. Liu et al. [21] deal with the embedded FS on the imbalanced data, i.e. fraud detection and cancer diagnosis, where they propose a weighted Gini index GI-FS, specifically designed to handle imbalanced data. ECoFFeS [22] is a comprehensive user-friendly and standalone software intended for automated FS in drug discovery, since it incorporates a set of single-objective and multi-objective evolutionary computation algorithms. Wang et al. [39] shows the application of PSO algorithm for descriptor selection.

This paper is an extension of the [14], where we employed an FS procedure using the logistic regression and hold-out validation. We have employed a threshold mechanism to manipulate with the attendance matrix, and an AUC statistical indicator. Self-adaptive differential evolution was used as optimization algorithm. Here, the general goal is to find out whether FS can contribute to reduce the number of marketing phone calls and thus decrease marketing costs. We employ a custom data reduction algorithm for feature selection, and compare its performance to performance of: (1) complete dataset, and (2) recursive feature elimination (RFE) method. We combine the favourable modeling characteristics of an artificial neural network and valuable optimization characteristics of a self-adaptive differential evolution. As a benchmark, we adopt a known UCI Machine Learning dataset named "Bank Marketing Data Set" [26], composed by Portuguese banking institutions during a campaign of phone calls. The dataset consists of many features (which we even enlarge) that are divided into three major groups: personal, social and financial. The purpose of the dataset is to predict whether a client is willing to make (subscript) a bank deposit or not. The problem we are facing is building a model from the dataset in a way that would maximize prediction performance. An example of bank marketing classification is shown in [32]. Here, we propose an automatic FS optimization process to obtain: (1) simpler modeling problem, (2) higher prediction performance, and (3) lower time complexity. Unlikely to [14], here we evaluate trial solutions using the ten-fold cross-validation procedure. By cross-validating, we avoid any random (bias) effects, which are present if the single hold-out validation is used, and thus expect that the proposed solution will actually increase prediction performance and lower the marketing costs simultaneously.

The outline of the paper is as follows. Section 2 deals with the basic information needed for understanding subjects that follows. Section 3 outlines the

proposed method, while Section 4 presents the obtained results. Section 5 concludes the paper and outlines directions for the future work.

2 Related Information

This section focuses on the background information that is needed for understanding the subjects that follow. In line with this, two prerequisites are necessary, i.e. an artificial neural networks and a differential evolution. Additionally to the original differential evolution, its self-adaptive variant is also illustrated. In the remainder of the paper, the aforementioned topics are presented in detail.

2.1 Artificial Neural Network

An artificial neural network (ANN) is a common ML modelling tool, known for its universal versatility for arbitrary approximations tasks [12]. Typically, ANN consists of multiple layers of perceptrons, i.e. building blocks, which simulate the behaviour of human. A perceptron consists of bias (sum), weight (scaling) and transfer function. Many perceptrons connected into the ANN can address diverse types of problems, such as regression, classification, clustering, dimensionality reduction and others [20].

A two-phase process is typically employed in order to evaluate the ANN: ANN-training (learning) and ANN-validation (evaluation). Those are run sequentially on two non-overlapping samples of the original dataset, i.e. training and validation samples. At first, the training is performed and then validation is executed to derive the predictive ability of the ANN by directly comparing the known and predicted results. However, such inference may be biased, due to the lack of generalization or representativeness of the validation dataset. In order to avoid this problem, a special type of validation is used, i.e. k-fold cross validation. The latter splits the dataset into k equal folds, and uses each of them exactly once for the validation, and the rest of the $k-1$ folds for training [12]. In this way, multiple (k) training and validation processes are run to obtain k results, which can then be averaged to form a consistent and unbiased measure of predictive ability.

2.2 Differential Evolution

Differential Evolution is an evolutionary, population-based, nature-inspired algorithm for global optimization, proposed by Storn and Price [36]. It belongs to a family of Evolutionary Algorithms (EA). Like other stochastic population-based nature-inspired algorithms, DE represents its candidate solutions as D-dimensional real-valued vectors $\mathbf{x}_i^{(t)}$ with elements $x \in [0, 1]$, in other words [34, 37]:

$$\mathbf{x}_i^{(t)} = \{x_{i,1}^{(t)}, \ldots, x_{i,D}^{(t)}\}, \quad \text{for } i = 1, \ldots, NP, \tag{1}$$

where NP denotes the population size, and D is the dimension of the problem.

The principle of the algorithm bases on applying the three genetic operators: mutation, crossover and selection, launched sequentially. The first among them, i.e. mutation, is used to encourage genetic diversity of the solutions, and prevent convergence to the local optimum. It is performed by taking the difference vector between two individuals and scaling its difference to form the mutant vector. For instance, mutation strategy, called 'DE/rand/1/bin', is formalized in Eq. (2):

$$\mathbf{u}_i^{(t)} = \mathbf{x}_{r0}^{(t)} + F \cdot (\mathbf{x}_{r1}^{(t)} - \mathbf{x}_{r2}^{(t)}), \quad \text{for } i = 1, \ldots, NP, \tag{2}$$

where $F \in (0.0, 1.0]$ represents the (stepsize) scaling factor and NP the population size. Thus, indices *r0*, *r1*, and *r2*, are the randomly selected numbers, drawn from uniform distribution in the interval $1, \ldots, NP$. Those denote the corresponding solution that must be different from the target. Crossover is then employed to combine the mutant vector with individuals $\mathbf{x}_{ri}$ to form the trial vector w_i, represented by Eq. (3):

$$w_{i,j}^{(t)} = \begin{cases} u_{i,j}^{(t)}, \text{ if } \mathrm{rand}_j(0,1) \leq CR \vee j = j_{rand}, \\ x_{i,j}^{(t)}, \qquad \text{otherwise}, \end{cases} \tag{3}$$

where $CR \in [0.0, 1.0]$ means the crossover rate and $j = 1, \ldots, D$. The third genetic operator, i.e., selection, is used to compare the two vectors, i.e., trial $\mathbf{w}_i$ and target $\mathbf{x}_i$, and select the better between them. The selection procedure is outlined in Eq. (4):

$$\mathbf{x}_i^{(t+1)} = \begin{cases} \mathbf{w}_i^{(t)}, & \text{if } f(\mathbf{w}_i^{(t)}) \leq f(\mathbf{x}_i^{(t)}), \\ \mathbf{x}_i^{(t)}, & \text{otherwise}. \end{cases} \tag{4}$$

The better among trial and target vector is preserved into the candidate solution vector $\mathbf{x}_i$ which proceeds into the next generation.

Self-adaptive Differential Evolution. DE is a simple and very useful algorithm, which can be hybridized to improve its search performance. For example, Brest et al. propose the self-adapted version of DE (jDE) [6] that self-adapts control parameters F and CR during the search process in order to simulate the "evolution of the evolution". In case of jDE, representation of individuals changes according to Eq. (5):

$$\mathbf{x}_i^{(t)} = (x_{i,1}^{(t)}, x_{i,2}^{(t)}, ..., x_{i,M}^{(t)}, F_i^{(t)}, CR_i^{(t)}), \tag{5}$$

where control parameters $F_i^{(t)}$ and $CR_i^{(t)}$ undergo specific variation operator, formalized in Eq. (6) and Eq. (7):

$$F_i^{(t+1)} = \begin{cases} F_l + \mathrm{rand}_1 * (F_u - F_l) & \text{if } \mathrm{rand}_2 < \tau_1, \\ F_i^{(t)} & \text{otherwise}, \end{cases} \tag{6}$$

and

$$CR_i^{(t+1)} = \begin{cases} \mathrm{rand}_3 & \text{if } \mathrm{rand}_4 < \tau_2, \\ CR_i^{(t)} & \text{otherwise}. \end{cases} \tag{7}$$

Here, $\text{rand}_{i=1...4} \in [0,1]$ represent the random numbers drawn from uniform distribution in interval $[0,1]$, τ_1, τ_2 learning rates and F_l, F_u the lower and upper bounds for scale factor F.

3 Proposed Method

This section presents the practical implementation of the KDD procedure and the synthesis of the proposed FS. As mentioned in the Introduction, this procedure consists of six phases. In this sense, the aims of the phases are discussed in a nutshell.

The first among them, i.e., problem specification, deals with the purpose of the study. Here, the provided database is examined, and the final objectives of the study are considered. The hypotheses to be checked are considered, and the expected results discussed with the help of expert knowledge. Also, relevant information is explored about the problem and related literature.

By completing the problem specification phase, the problem understanding phase follows. Here, the data and any interconnections among the data are checked visually to obtain basic data comprehension. After that, features and response variable(s) are selected to form a reduced representation of the database, i.e. dataset [13]. For continuing the study, only the dataset is relevant.

The third phase of KDD is the data preprocessing. Two sub-phases are contained here, i.e. data preparation and data reduction. During the former subphase, tasks like data cleaning, integration, normalization and transformation are applied. During the latter, the original database is reduced, to eliminate the redundancy or irrelevancy of some of the elements [15]. Here, four methods exist: (1) feature selection (FS), (2) instance selection (IS), (3) discretization and (4) feature/instance extraction. FS deals with eliminating the explanatory variables from the dataset in order to eliminate their redundancy, while IS deals with eliminating the instances from the dataset. The discretization is a data reduction principle, where the domain is simplified into discrete regions. Feature/Instance extraction is a method where new variables are generated.

The fourth KDD phase is the data mining. It is the central and critical phase of the KDD, where actual hidden patterns, relations and interconnections are searched for. First, the appropriate DM task and the DM method are selected. Typically, DM tasks are: regression, classification, clustering and others. The more commonly DM methods are: ANNs, decision trees, support vector machines (SVM), rule-based algorithms and others. It is known that each DM method does not suit each problem well, and that some experimenting might sometimes be necessary.

The fifth KDD phase is the evaluation. Here, the researcher interprets the performance of the KDD. Typically, performance is evaluated by the model's performance using a single criterion or multiple selection criteria. Many selection criteria exist, such as information, distance, dependence, consistency and accuracy measures. Each of them defines a critical evaluation criterion - performance (ability) of the DM.

The last step of KDD is the result exploitation, where the researcher uses the knowledge discovered practically. Visualization, documentation and reporting come into play here, together with comparison to the expectations set in the first phase. Also, built knowledge discovered can be used pro-actively in the daily routine. In the subsections that follow, the aforementioned KDD phases are illustrated from the proposed method point of view in detail.

3.1 Problem Specification

The purpose of our study is to identify and select the best subset, which would assure to find the maximal number of bank clients, willing to subscript the bank deposit. We employ the dataset accessible at UCI Machine Learning, i.e. the Bank Marketing Dataset that was collected by Moro, Cortez and Rita in 2014 [26]. The dataset comes from Portuguese banking institutions which have been advertising their products. Using campaign phone calls, they have been contacting clients and promoting bank deposit subscriptions. The institutions have also, simultaneously, been recognizing the client's personal, social and financial habits, and client's decision to subscript the deposit or not [33].

We are sure that the KDD from this dataset could help us predict interested clients. Banking institutions can nevertheless save enormous efforts and costs of bank marketing if they avoid contacting each client individually again and again. In order to decrease the pool of potential bank depositors, we propose to build a model with which clients with past information could be evaluated, and only most potential among them would be contacted in future.

3.2 Problem Understanding

In the second KDD phase, we discuss briefly the features held in the dataset to obtain basic comprehension of data. Actually, collected data from an original dataset have been accumulated into a table to form an original dataset $\mathbf{X}$, which are depicted in table 1. Besides features, a response variable *deposit_subscription* is added to the table also. The *deposit_subscription* is a binary value with the following meaning: 0 means reject, and 1 means the subscription of the bank deposit.

Let us mention that the original dataset $\mathbf{X}$ consists of 20 features and 41,188 instances describing clients. Although this seems reasonable, much less than 41,188 clients are examined practically, since the majority of them are contacted more than once during the campaign period.

As can be seen from the table 1, the features can be either numerical or categorical. Each numerical variable is defined with a corresponding range of feasible values. On the other hand, the categorical variables are specified with a set of discrete values that are also presented in the table.

3.3 Data Preprocessing

In our study, data preparation consists of two data transformations: dummification, and normalization. Dummification is a transformation, where categorical

Table 1. List of explanatory and dependent variables in the original dataset **X**.

No.	Feature	Type of feature	Range of feature
1.	*age*	numerical	17 - 98 years
2.	*job*	categorical	administrator, blue-collar, entrepreneur, housemaid, management, retired, self-employed, services, student, technician, unemployed, unknown
3.	*marital*	categorical	divorced, married, single, unknown
4.	*education*	categorical	basic.4y, basic.6y, basic.9y, high.school, illiterate, professional.course, university.degree, unknown
5.	*default*	categorical	no, yes, unknown
6.	*housing*	categorical	no, yes, unknown
7.	*loan*	categorical	no, yes, unknown
8.	*contact*	categorical	cellular, telephone
9.	*month*	categorical	March, April, May, June, July, August, September, October, November, December
10.	*day_of_week*	categorical	Monday, Tuesday, Wednesday, Thursday, Friday
11.	*duration*	numerical	0 - 4918
12.	*campaign*	numerical	1 - 56
13.	*pdays*	numerical	0 - 999
14.	*previous*	numerical	0 - 27
15.	*poutcome*	categorical	failure, success, nonexistent
16.	*emp.var.rate*	numerical	-3.4 - 1.4
17.	*cons.price.idx*	numerical	92.201 - 94.767
18.	*cons.conf.idx*	numerical	-50.8 - -26.9
19.	*euribor3m*	numerical	0.634 - 5.045
20.	*nr.employed*	numerical	4963.6 - 5228.1
21.	*deposit_subscription*	binary	0 - 1

or numerical variables are transformed into binary features using one-hot encoding. One-hot encoding is a transformation that encodes categorical variables with multiple classes into a binary vector. Exactly one of the binary vector values is 1 and the rest of them are 0. On the other hand, normalization stands for transformation, which modifies the values to the interval [0,1] proportionally.

Data Preparation. The explanatory variables are dummified and normalized as presented in table 2, where *pdays* although it is numerical, is modified using the dummification. Value 0 presents the situation that the client has never been contacted before, and vice-versa. In this way, the original dataset is widened to 70 variables that form the adjusted dataset $\mathbf{X}'$ [14]. Actually, 62 binary variables and 8 numerical normalized variables are included here. All of them are presented in the correlation analysis plot (Fig. 1). High correlation and, thus, redundancy is reported for the *age*, *months*, *day* and *marital* binary variables. In

Table 2. Transformation of categorical and numerical variables.

Categorical transformations		Numerical transformations	
age	8 dummies	*duration*	[0,1] normalization
job	12 dummies	*campaign*	[0,1] normalization
marital	4 dummies	*previous*	[0,1] normalization
education	8 dummies	*emp.var.rate*	[0,1] normalization
default	3 dummies	*cons.price.idx*	[0,1] normalization
housing	3 dummies	*cons.conf.idx*	[0,1] normalization
loan	3 dummies	*euribor3m*	[0,1] normalization
contact	2 dummies	*nr.employed*	[0,1] normalization
month	10 dummies		
day_of_week	5 dummies		
pdays	1 dummy		
poutcome	3 dummies		

addition, high correlation is examined among the majority of numerical variables and for *months* binary variables. Almost perfect correlation is observed between variables *emp.var.rate* and *euribor3m*.

Data Reduction. The proposed wrapper-based FS is implemented using the modified WFS-jDE that is obtained from the original one by modifying the following jDE elements: (1) representation of individuals, and (2) fitness function evaluation. While the representation of individuals is performed according to the Eq. (1), the fitness function evaluation is calculated as follows: Let us assume that an attendance vector $\mathbf{a}_i^{(t)} = \{a_{i,j}\}$, corresponding to each candidate solution $\mathbf{x}_i^{(t)}$, is given. The elements of the attendance vector are calculated according to Eq. (8), as follows:

$$\mathbf{a}_i^{(t)} = \begin{cases} 0, \text{ if } \mathbf{x}_i \leq 0.5 \\ 1, \text{ otherwise,} \end{cases} \tag{8}$$

Then, the attendance vector $\mathbf{a}_i^{(t)}$ specifies the presence or absence of specific features in the reduced subset $\mathbf{X}^* \subset \mathbf{X}'$.

$$g : \mathbf{X}' \overset{\mathbf{a_i}}{\rightarrow} \mathbf{X}^* \tag{9}$$

where the subset $\mathbf{X}^*$ contains features with an attendance vector value of 1, and omits the features with an attendance vector value of 0. Each subset $\mathbf{X}^*$ is evaluated using the CV, which will be described in Subsection 3.5.

Alternative to the described FS method is the recursive feature elimination algorithm, or shortly RFE. Here, logistic regression is used as an estimator. The purpose of RFE is to detect and remove the least important (one or more) feature in each iteration, using the feature ranking method. The recursive process is run until the desired (predefined) number of features to select is obtained [30].

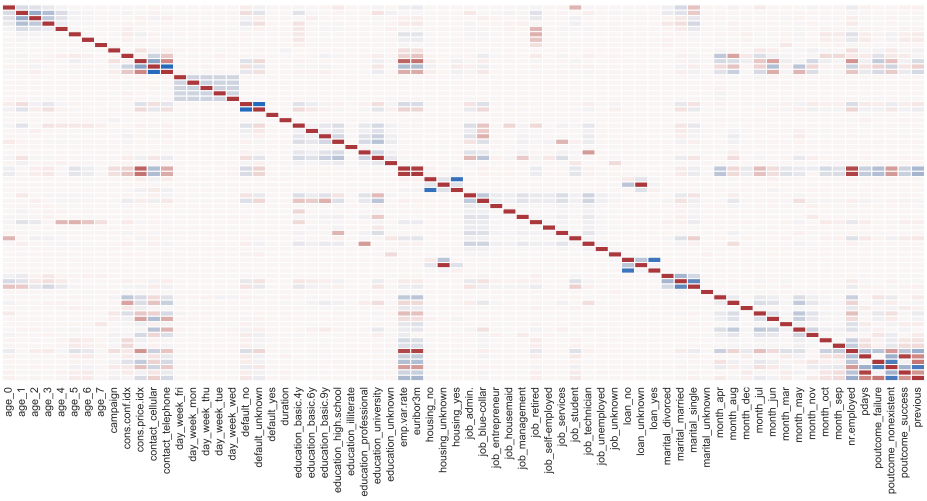

Fig. 1. Correlation analysis plot of adjusted dataset $\mathbf{X}'$.

3.4 Data Mining

The problem to be solved is classification. By taking into account the past explanatory variables, we try to predict whether a client is interested to subscript the bank deposit or not. ANN is used as the DM method. The ANN consists of two phases, i.e., ANN-training and ANN-validation, where the latter is used as the basis for derivation of a confusion matrix, used in the evaluation phase.

ANN-training is an iterative procedure, where the weights and biases of the ANN are modified by the learning algorithm. We use the supervised feed-forward ANN. This means that the ANN-training is two-phase learning process: (1) the ANN is forward-propagated to obtain the analogue output, and (2) the obtained output is compared to the desired (binary) target value. The difference between these is calculated next, which is then back-propagated to modify the weights and biases of the ANN.

ANN-validation consists of the forward-propagation only. Here, the validation data not seen during the training are used. Each forward-propagation gives a classification output that is compared with the actual class (client subscripts/rejects the deposit). This forms four different scenarios, that are summarized in the confusion matrix. ANN-validation is due to the 10-fold cross-validation (CV) mechanism performed 10 times.

3.5 Evaluation

Performance of a classifier is evaluated in the evaluation phase of KDD. Evaluation of the classification results is performed on the basis calculated by the confusion matrix. Selected subset $\mathbf{X}^*$ from the data reduction sub-phase is next split into fixed and non-overlapping $k = 10$ folds for the CV calculations. Nine of

them are used for ANN-training, while the tenth for the ANN-validation. This process is repeated until all of the folds are used exactly once for ANN-validation. Prediction results are recorded afterwards, and saved in the form of confusion matrix (table 6), i.e. a tabular indicator of a goodness-of-classification, consisting of four quadrants, and differentiating between true and false predictions. In

Table 3. An example of a confusion matrix.

	True YES	True NO
Predicted YES	TP	FP (type I error)
Predicted NO	FN (type II error)	TN

a confusion matrix, proper prediction of a deposit subscription means a true positive prediction, while proper prediction of rejection a true negative. Two other cases exist: A false positive for an improper prediction of rejection (type I error), and a false negative for the improper prediction of subscription (type II error). Since the confusion matrix is a universal statistical indicator of prediction (classification) performance, many derivative measures of accuracy can be calculated from it. Specifically, we use the confusion matrix for the calculation of overall accuracy and sensitivity, which can be outlined by Eqs. (10) and (11).

$$accuracy = \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}}, \tag{10}$$

$$sensitivity = \frac{\text{TP}}{\text{TP+FN}}, \tag{11}$$

where both the *accuracy* and *sensitivity* are used for a fitness function computation, as:

$$ff(trial\_solution) = 0.5 \cdot \overline{accuracy} + 0.5 \cdot \overline{sensitivity}. \tag{12}$$

Variables $\overline{accuracy}$ and $\overline{sensitivity}$ present the mean (average) values of accuracy and sensitivity over ten folds, used in CV. We are dealing with an imbalanced dataset, and the accuracy may solely be a biased measure. In order to prevent it, a combination is used with sensitivity. By taking the coefficients of each measure 0.5 we treat both of them equally. The obtained fitness function is next fed as a feedback loop into the optimization algorithm, and the best candidate solution with the best subset $\mathbf{X}^{(best)}$ can be extracted, as follows:

$$\mathbf{X}^{(best)} = max\left(f(\mathbf{X}^{*})\right) \tag{13}$$

Results are exploited on the basis of the obtained $\mathbf{X}^{(best)}$.

3.6 Result Exploitation

The obtained result (built model) can be used on a regular basis to classify interested banking clients. The complete WFS-jDE procedure is shown in the use-case diagram in Fig. 2.

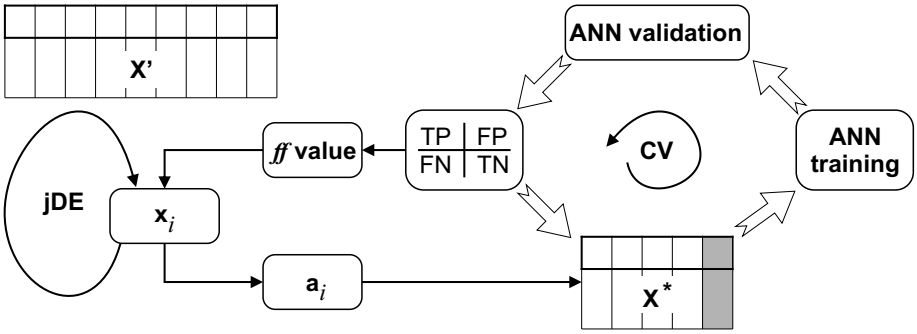

Fig. 2. Use-case diagram of WFS-jDE. Two loops are shown, i.e. a jDE loop, which is controlled by a maximum number of evaluations ($nFES$), and a CV loop, that presents the $k = 10$-fold cross-validation. First, candidate solution vector $\mathbf{x}_i$ is generated, and transformed according to attendance vector $\mathbf{a}_i$ to form a trial subset $\mathbf{X}^*$. This is then trained sequentially, and validated using the CV mechanism, and the results are visualized in a confusion matrix individually for each fold. Fitness function value is calculated once all the folds are evaluated.

4 Experiments and Results

The purpose of the following section is twofold. First, we investigate if we can predict bank clients interested in making deposits. Second, we examine if the proposed WFS-jDE can be used for improving prediction performance.

For the experiments, the widened Bank Marketing Data Set (adjusted dataset $\mathbf{X}'$) introduced in Subsection 3.3 was used. ANN was used as a modeling tool to ensure the confusion matrix calculation and jDE as an optimization algorithm to optimize the attendance vector.

ANN was implemented in the Python programming language using, the Keras deep learning library [9]. The following configuration of ANN was used:

- input layer with number of input neurons N_i varying by the number of features,
- first hidden layer with a number of hidden neurons $N_{h1} = 20$,
- second hidden layer with a number of hidden neurons $N_{h2} = 20$,
- output layer with a single output neuron $N_o = 1$.

A sigmoid function is used in each layer. The architecture of ANN was taken from the paper [11], which proposes 48 input neurons, 20 neurons in the first hidden layer, 15 neurons in the second hidden layer, and a single output neuron. An advanced ANN training algorithm Adam [16] was used with fixed size of epochs. No early-stopping criterion was used. Table 4 presents ANN parameter configuration. The initial configuration of the jDE optimization algorithm is referenced in the table 5. Scikit-learn library was taken to provide RFE procedure [29]. RFE is the alternative to the proposed solution, which requires the setting of the final number of features. An adjusted dataset $\mathbf{X}'$ is used as a basis.

Table 4. Configuration of ANN.

Parameter	Value
Number of epochs N_{epochs}	10
Batch size *batch_size*	32
Learning constant *lr*	0.001
Loss function	*binary_crossentropy*

In what follows, two experimental tests are conducted for the adjusted dataset $\mathbf{X}'$, best subset $\mathbf{X}^{(\mathbf{best})}$ and RFE subset $\mathbf{X}^{(\mathbf{RFE})}$:

- basic evaluation of prediction performance,
- detailed comparison among statistical indicators.

Table 5. Initial configuration of jDE optimization algorithm.

Parameter	Value
Initial scaling factor F	0.5
Initial crossover ratio CR	0.9
Self-adaptive learning rate τ_1	0.1
Self-adaptive learning rate τ_2	0.1
Population size NP	100
Number of evaluations $nFES$	10,000

First, basic evaluation of prediction performance is presented. Second, detailed comparison among statistical indicators is displayed. Confusion matrix is used for basic evaluation, while, for detailed evaluation, statistical indicators (measures of accuracy) are derived from the confusion matrix. Following eight statistical indicators are used: accuracy, sensitivity, specificity, precision, negative predictive value (NPV), type I and type II errors, and F1 score. Additionally, fitness function value (*ff*) is displayed for comparison. Since the CV mechanism is employed, average values are displayed for all of the obtained results. In the remainder of the paper, the aforementioned experiments are explained in detail.

4.1 Basic Evaluation

Basic evaluation is performed for the adjusted dataset $\mathbf{X}'$, best subset $\mathbf{X}^{(\mathbf{best})}$ and RFE subset $\mathbf{X}^{(\mathbf{RFE})}$. The results of the adjusted dataset are obtained by running the classification on the complete adjusted dataset $\mathbf{X}'$. The average confusion matrix is recorded in table 6, which exhibits basic prediction performance. In average, 216.6 clients are predicted properly for bank deposit subscription and 3532.8 clients on average for bank deposit rejection. 247.4 clients are predicted incorrectly, since they actually wish to subscript a deposit, but the model predicts

Table 6. Average confusion matrix for adjusted dataset $\mathbf{X}'$.

	True YES	True NO
Predicted YES	216.60	122.00
Predicted NO	247.40	3532.80

inversely. 122 clients are predicted incorrectly as well, since the model predicts them to subscribe, but they actually reject the bank deposit. Table 6 is thus a benchmark predictive ability, which we would try to improve using the WFS-jDE procedure. Table 7 exhibits results on the best subset $\mathbf{X}^{(\mathbf{best})}$, which are

Table 7. Average confusion matrix for best subset $\mathbf{X}^{(\mathbf{best})}$.

	True YES	True NO
Predicted YES	234.60	141.00
Predicted NO	229.40	3513.80

obtained by applying the iterative WFS-jDE procedure on the complete dataset $\mathbf{X}'$. The dimension is reduced from 70 features to 37, meaning that almost half of the features are omitted. By applying the WFS-jDE, the number of properly predicted subscriptions rises to 234.6, compared to 216.6 in table 6, and the number of properly predicted rejections diminishes from 3532.8 to 3513.8. Although the first characteristic is very welcome - the rise of proper predictions provides more bank deposit subscriptions, this rise is conditioned by a fall of proper rejection predictions. The type I error increases slightly due to that reason, and the type II error decreases. The next subsection quantifies these effects. Categorical variables that are applicable in the best subset $\mathbf{X}^{(\mathbf{best})}$ (those that are not omitted), are outlined in the table 8. Alongside, the following binary and numerical variables are present in the best subset $\mathbf{X}^{(\mathbf{best})}$: *pdays*, *duration*, *emp.var.rate*, *cons.price.idx*, *cons.conf.idx* and *nr.employed*.

Listed variables in the best subset $\mathbf{X}^{(\mathbf{best})}$ are similar to two FS studies. Both of the studies [11, 28] rank the most important feature to be *duration*. [28], who use the information gain and chi-square characteristic, then rank *poutcome*, *month*, *pdays* and *contact*. On the other hand, according to the sensitivity analysis, ranks *month*, *poutcome*, *contact* and *job* to be the other most important features [11]. Third evaluation is the evaluation of the feature selection alternative - RFE method with the number of variables to select set to 35. According to the confusion matrix 9, this alternative is the worst among all. True positive instance is decreased drastically, scoring barely 85 clients. Number of false positives decreases as well, which is desired, but the false negatives increase. True negative instance scores the highest value among all. Table 10 lists the categorical variables, used in the RFE subset $\mathbf{X}^{(\mathbf{RFE})}$. It is noticeable that each feature that is listed in the table, incorporates all of its feature values (except the *day_of_week* feature, where the Tuesday and Wednesday feature values are missing).

Table 8. List of categorical variables in the best subset $\mathbf{X}^{(\mathbf{best})}$.

Features	Feature values
age	17-25, 26-34, 44-52, 53-61, 62-70
job	management, services, technician
marital	unknown
education	basic.4y, basic.9y, professional.course, university.degree, unknown
month	March, May, June, September, October, December
day_of_week	Wednesday, Thursday
default	yes
housing	no, yes, unknown
loan	yes
poutcome	failure, success, nonexistent
contact	telephone

Table 9. Average confusion matrix for RFE subset $\mathbf{X}^{(\mathbf{RFE})}$.

	True YES	True NO
Predicted YES	85.30	44.40
Predicted NO	378.70	3610.40

4.2 Detailed Evaluation

The aim of this section is to provide the detailed evaluation between the adjusted dataset $\mathbf{X}'$, best subset $\mathbf{X}^{(\mathbf{best})}$ and RFE subset $\mathbf{X}^{(\mathbf{RFE})}$. The results displayed in table 11 present the mentioned statistical indicators, where NPV means negative predictive value and *ff* means fitness function value. The displayed statistical indicators are averaged for the ten folds used in the CV mechanism. Examination of the detailed results identifies a slight drop in overall accuracy by applying the proposed WFS-jDE solution. Substantial increase in sensitivity is indicated, which causes fitness function value *ff* to increase as well. An increase of fitness function value indicate the successfulness of the WFS-jDE. Specificity is due to the drop of proper rejection predictions lowered marginally compared to adjusted dataset, and thus increases type I errors proportionately. Specificity is the highest for the RFE subset. On the other hand, type II error lowers drastically which causes that far less clients are "ignored". After all, regardless of any statistical indicator, the number of bank deposit subscriptions increases strongly. The RFE procedure decreases the overall accuracy and drastically decreases the sensitivity. Due to such a low value of sensitivity and consequently fitness function value, this FS procedure is not appropriate for our problem. A good quality of the WFS-jDE comes in the slight increase of type I errors but significant drop of type II errors. The dataset is imbalanced and thus biased in some way. A special treatment is necessary, which comes in the form of fitness function. The latter

Table 10. List of categorical variables in the RFE subset $\mathbf{X}^{(\mathbf{RFE})}$.

Features	Feature values
age	17-25, 26-34, 35-43, 44-52, 53-61, 62-70, 71-79, 80+
job	administrator, blue-collar, entrepreneur, housemaid, management, retired, self-employed, services, student, technician, unemployed, unknown
marital	single, married, divorced, unknown
day_of_week	Monday, Thursday, Friday
default	no, yes, unknown
poutcome	failure, success, nonexistent
contact	telephone, cellular

Table 11. Detailed evaluation of prediction results.

Avg. indicator	Adjusted dataset $\mathbf{X}'$	Best subset $\mathbf{X}^{(\mathbf{best})}$	RFE subset $\mathbf{X}^{(\mathbf{RFE})}$
No. of features	70	37	35
Accuracy	0.9103	0.9101	0.8973
Sensitivity	0.4668	0.5056	0.1838
Specificity	0.9666	0.9614	0.9879
Precision	0.6407	0.6272	0.6572
NPV	0.9346	0.9388	0.9051
Type I error	0.0334	0.0386	0.0121
Type II error	0.5332	0.4944	0.8162
F1 score	0.5381	0.5584	0.2869
ff value	0.6886	0.7078	0.5406

tries to take the imbalance into account at least a bit. Different fitness functions would establish different results.

The results can be compared to those obtained in [14]. Type II error there is remarkably lower, i.e. 27 for the "original database" and 28 for the "reduced database". The authors report that their accuracy increases by almost 2% by employing the FS, while we experienced a drop in accuracy. However, the authors there employed a simple hold-out validation and a logistic regression. Additionally, they implemented a special procedure for threshold (TH), which was used for mapping candidate solutions. They used simultaneous optimization of threshold TH during the optimization process and thus improved the reaction of FS significantly on the imbalanced dataset. Instead of custom fitness function from accuracy and sensitivity, they optimized the AUC score. All those changes call for a difficult comparison between the [14] and our study. Another study that deals with the bank marketing is [28]. The authors here compared the F1 score. By employing several traditional FS applications, the authors show that maximal increase of F1 score is 0.01. In our case, an increase of more than 0.02 is applicable.

5 Conclusion

In this paper, we have proposed a wrapper-based feature selection using the ANN and jDE. We have tested its efficiency on a Bank Marketing Data Set. We figured out that modeling the past information about bank clients is a suitable task that might come handy on an everyday basis.

We have shown that the proposed FS procedure reduces the size of the dataset successfully. From an initial 70 variables, we formed the best subset of 37 variables, i.e. almost half less variables. Such a decrease of the number of variables not only reduces the complexity of the dataset, but also improves the predictive performance of a classifier. The RFE alternative, which is more common in practice, caused the predictive ability to suffer. Although, we can say that on this case study, FS procedure can be used beneficially to improve the predictive ability by eliminating some redundant or irrelevant features.

The proposed WFS-jDE is especially suitable for imbalanced datasets, due to the arbitrary selection of fitness function. However, self-adaptation of the threshold TH is highly desired in such cases.

In future, we would like to test WFS-jDE with the self-adaptation of TH on several diverse datasets. We would like to employ universal fitness function, and run the jDE with a higher number of evaluations, to ensure the convergence. Constrained optimization should be utilized as well, to control the number of features better in the reduced dataset.

References

1. Charu C Aggarwal. *Data classification: algorithms and applications*. CRC press, 2014.
2. Qasem Al-Tashi, Said Jadid Abdul Kadir, Helmi Md Rais, Seyedali Mirjalili, and Hitham Alhussian. Binary Optimization Using Hybrid Grey Wolf Optimization for Feature Selection. *IEEE Access*, 7:39496–39508, 2019.
3. Javier Apolloni, Guillermo Leguizamón, and Enrique Alba. Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments. *Applied Soft Computing*, 38:922–932, 2016.
4. David W Bates, Suchi Saria, Lucila Ohno-Machado, Anand Shah, and Gabriel Escobar. Big data in health care: using analytics to identify and manage high-risk and high-cost patients. *Health Affairs*, 33(7):1123–1131, 2014.
5. Richard Bellman. Adaptive control processes: a guided tour princeton university press. *Princeton, New Jersey, USA*, 1961.
6. Janez Brest, Sašo Greiner, Borko Boškovič, Marjan Mernik, and Viljem Žumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE transactions on evolutionary computation*, 10(6):646–657, 2006.
7. Laura Cardona and Liana Amaya Moreno. Cash management cost reduction using data mining to forecast cash demand and lp to optimize resources. *Memetic Computing*, 4(2):127–134, 2012.
8. Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

9. François Chollet et al. Keras. https://keras.io, 2015.
10. Catherine Da Cunha, Bruno Agard, and Andrew Kusiak. Selection of modules for mass customisation. *International Journal of Production Research*, 48(5):1439–1454, 2010.
11. Hany A Elsalamony and Alaa M Elsayad. Bank direct marketing based on neural network. *International Journal of Engineering and Advanced Technology (IJEAT)*, 2(6), 2013.
12. Wolfgang Ertel. *Introduction to Artificial Intelligence*. Springer, 2018.
13. Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37, 1996.
14. Dušan Fister, Iztok Fister, Timotej Jagrič, Iztok Fister Jr., and Janez Brest. A novel self-adaptive differential evolution for feature selection using threshold mechanism. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 17–24. IEEE, 2018.
15. Salvador García, Julián Luengo, and Francisco Herrera. *Data Preprocessing in Data Mining*. Springer, New York, NY, 2015.
16. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
17. Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.
18. Mahdieh Labani, Parham Moradi, Fardin Ahmadizar, and Mahdi Jalili. A novel multivariate filter method for feature selection in text classification problems. *Engineering Applications of Artificial Intelligence*, 70:25–37, 2018.
19. Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industry 4.0. *Business & information systems engineering*, 6(4):239–242, 2014.
20. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.
21. Haoyue Liu, Meng Chu Zhou, and Qing Liu. An embedded feature selection method for imbalanced data classification. *IEEE/CAA Journal of Automatica Sinica*, 6(3):703–715, 2019.
22. Zhi-Zhong Liu, Jia-Wei Huang, Yong Wang, and Dong-Sheng Cao. ECoFFeS: A Software Using Evolutionary Computation for Feature Selection in Drug Discovery. *IEEE Access*, 6:20950–20963, 2018.
23. Meng Lu. Embedded feature selection accounting for unknown data heterogeneity. *Expert Systems with Applications*, 119:350–361, 2019.
24. Majdi Mafarja and Seyedali Mirjalili. Whale optimization approaches for wrapper feature selection. *Applied Soft Computing*, 62:441–453, 2018.
25. Pankhurhi Mallik, Chandrima Roy, Ekansh Maheshwari, Manjusha Pandey, and Siddharth Rautray. Analyzing student performance using data mining. *Ambient Communications and Computer Systems: RACCCS-2018*, page 307, 2019.
26. Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.
27. Opeyemi Osanaiye, Haibin Cai, Kim-Kwang Raymond Choo, Ali Dehghantanha, Zheng Xu, and Mqhele Dlodlo. Ensemble-based multi-filter feature selection method for ddos detection in cloud computing. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):130, 2016.
28. Tuba Parlar and Songül Kakilli Acaravci. Using Data Mining Techniques for Detecting the Important Features of the Bank Direct Marketing Data. *International Journal of Economics and Financial Issues*, 7(2):692–696, 2017.

29. Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Matieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, M. Perrot, and Eduard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
30. Rajasheker Pullanagari, Gabor Kereszturi, and Ian Yule. Integrating airborne hyperspectral, topographic, and soil data for estimating pasture quality using recursive feature elimination with random forest regression. *Remote Sensing*, 10(7):1117, 2018.
31. Sharan Ramjee and Aly El Gamal. Efficient wrapper feature selection using autoencoder and model based elimination. *arXiv preprint arXiv:1905.11592*, 2019.
32. Magdalena Scherer, Jacek Smolag, and Adam Gaweda. Predicting success of bank direct marketing by neuro-fuzzy systems. In *International Conference on Artificial Intelligence and Soft Computing*, pages 570–576. Springer, 2016.
33. Yosimar Oswaldo Serrano-Silva, Yenny Villuendas-Rey, and Cornelio Yáñez-Márquez. Automatic feature weighting for improving financial decision support systems. *Decision Support Systems*, 107:78–87, 2018.
34. Dan Simon. *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.
35. Uma Srinivasan and Bavani Arunasalam. Leveraging big data analytics to reduce healthcare costs. *IT professional*, 15(6):21–28, 2013.
36. Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.
37. Adam Viktorin, Roman Senkerik, Michal Pluhacek, Tomas Kadavy, and Ales Zamuda. Distance based parameter adaptation for success-history based differential evolution. *Swarm and Evolutionary Computation*, 2018.
38. Suhang Wang, Jiliang Tang, and Huan Liu. Embedded unsupervised feature selection. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.
39. Yong Wang, Jing-Jing Huang, Neng Zhou, Dong-Sheng Cao, Jie Dong, and Han-Xiong Li. Incorporating PLS model information into particle swarm optimization for descriptor selection in QSAR/QSPR. *Journal of Chemometrics*, 29(12):627–636, 2015.
40. Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding. Data mining with big data. *IEEE transactions on knowledge and data engineering*, 26(1):97–107, 2014.
41. Bing Xue, Mengjie Zhang, Will N Browne, and Xin Yao. A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Transactions on Evolutionary Computation*, 20(4):606–626, 2016.
42. Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 856–863, 2003.