A novel self-adaptive differential evolution for feature selection using threshold mechanism

1st Dušan Fister
Faculty of Economics and Business
University of Maribor
SI-2000 Maribor, Slovenia
dusan.fister1@um.si

2nd Iztok Fister
Faculty of Electrical Engineering and
Computer Science, University of Maribor
SI-2000 Maribor, Slovenia
iztok.fister@um.si

3rd Timotej Jagrič
Faculty of Economics and Business
University of Maribor
SI-2000 Maribor, Slovenia
timotej.jagric@um.si

4th Iztok Fister Jr.
Faculty of Electrical Engineering and
Computer Science, University of Maribor
SI-2000 Maribor, Slovenia
iztok.fister1@um.si

5th Janez Brest
Faculty of Electrical Engineering and
Computer Science, University of Maribor
SI-2000 Maribor, Slovenia
janez.brest@um.si

***Abstract*—Nowadays, most of databases for classification or regression consists of numerous features that describe the domain of interest. Therefore, they may have a huge influence on the results of classification/regression. A lot of research has shown that some features can be eliminated before the classification/regression in order to obtain better results. In this paper, we propose a novel solution that is based on self-adaptive differential evolution for feature selection on a econometric database. A new solution is systematically presented in this paper. Results of the proposed feature selection method, according to the ROC-AUC score, overcome results, obtained without using it.**

***Index Terms*—classification, differential evolution, linear regression, feature selection**

I. Introduction

Recently, the amount of data has been drastically increasing in almost all areas [12]. Let us mention only some examples: The rise of industry 4.0 caused producing enormous amount of data over the Internet of Things, with which processes and human communicate between each other in real-time [15]. Especially, medical imaging is a source of producing big data in medicine [17]. Athletes in different sports monitor their training sessions with smart watches equipped with various sensors for data acquisition [9]. However, traditional economics also rely on dozens of data for their operations [7]. Indeed, we can say that success of each company mostly bases on data.

On the other hand, researchers in many scientific domains have also been faced with finding new ways how to process such big amount of data. As a result, a bunch of methods for analyzing these data has been emerged during the decades in the data mining domain [2]. Data mining process offer researchers and practitioners to gain new insights into data. Actually, the new knowledge discovered from these data can influence the future decision-making processes either of companies or individuals. However, data mining is very complex process consisting of many steps.

Beside an enormous number of records, the big data consist also of many features [10], where each feature can have many attributes. Typically, these features suffer a classifying process, in which they are recognized, differentiates, and understood. The major problem accompanying the process is a dimension of data. Fortunately, it turns on that some features in data are redundant and therefore irrelevant in the case of information loss. As a result, the major task before classification is how to reduce the number of features such that the classification accuracy remains the same or even better and the training training time is decreased. Herewith, the Feature Selection (FS) is usually used [22].

In general, the FS is an optimization problem of huge time-complexity. This means that the exhaustive search exploring all potential solutions in the search space is not appropriate for solving these problems. Hence, researchers have been developed algorithms that are capable of finding pseudo-optimal solutions in real-time. Nowadays, the stochastic population-based nature-inspired algorithms [18], [20]–[22] become the general, very powerful tool for solving the hardest (so called NP-hard) problems, to which the FS is also counted.

This paper proposes a novel self-adaptive Differential Evolution (DE) [19] for FS using threshold mechanism. It is a continuation of already published work [8], where no FS mechanism was used. This study has shown that the logistic regression successfully satisfied its demands and has therefore been used here as well.

The DE is a member of Evolutionary Algorithms (EAs) appropriate for global optimization. The self-adaptive version of this algorithm was proposed by Brest et al. [3] known under the name jDE that improves the results of its original counterpart primarily in solving the continuous optimization problems. Although no particular reason for selecting the jDE algorithm can be stated, jDE algorithm was applied for solving the FS problem in this study. Nevertheless, it brings secure and trackable adaptation and on the other hand does not suffer from

premature convergence and stagnation that are weaknesses of SHADE family based algorithms. This algorithm introduces the so-called threshold mechanism, which presents a hybridization of the original jDE with a local search heuristic. The heuristic improves each solution by searching for the optimal threshold, which determines the presence/absence of the particular feature in the solution.

The algorithm was applied as a pre-processing method of a logistic regression, whose task was to predict the potential bank depositor, according to publicly available database of bank deposits [16], based on phone call. The deposit database includes records of more than 40,000 bank clients. The results of experiments revealed a huge potential of the proposed algorithm. Let us mention that the results do not present comparisons between state-of-the-art FS methods, but should be observed as an analysis of the original and reduced database. Indeed, such kind of data analysis is devoted to a special research area in economics, i.e., econometrics [13] that represents a combination of more scientific domains, like economy, mathematics and statistics.

The structure of this paper is as follows: Section II introduces classification in economics and FS problem briefly. Section III focuses on DE algorithm and its self-adaptive variant jDE. Section IV proposes an FS solution, while chapter V its results. Original database and its modification are presented as well. The paper concludes with the evaluation of the proposed solution and future outlines.

II. Classification in economics

Classification in economics is one of the fundamental research areas of econometrics and is generally performed using the logistic regression. It is used broadly for predicting potential buyers/clients, that may be interested for buying or taking the benefits of a product. Typically, most precise predictions are desired. Corporations and organizations actually invest a lot of assets for promoting new products by taking campaign phone calls and building a database about them simultaneously. Since success of selling a product over the phone may highly depend, corporations would likely to firstly contact most interested buyers/clients, whose probability of buying a new product is, based on past experience, high. In that way, corporation will increase sales and will not waste time by contacting low or not interested buyers. It is therefore very important, that used classification and pre-processing techniques maximize prediction performance. Since classification with logistic regression is deterministic, an emphasis is placed to the FS pre-processing method.

The FS problem can formally be defined as follows. Let us assume that a set of features $\mathcal{D} = \{f_1, \ldots, f_N\}$ is given, where N denotes the number of features. The goal is to select a subset of features $\mathcal{F} = \{f_{\pi_1}, \ldots, f_{\pi_M}\}$, where π_i for $i = 1, \ldots, M$ denotes permutation of features, and M id the number of elements, such that $M < N$ and $\mathcal{F} \subset \mathcal{D}$, where M is the number all of features in $\mathcal{D}$.

A. Outline of the deposit database

In our experimental work, an online deposit database, called "Bank Marketing Data Set" [16], was employed. The data in database was collected from direct marketing campaign of a Portuguese banking institution. Each datapoint presents a transaction denoting the phone call to client. During the phone call to a client by Bank employee, some attributes are inquired and saved into the database. The decision of a client, whether to subscribe ("yes") or reject ("no") the bank term deposit, is filled into the database as an outcome. The latter acts as a dependent variable to deliver a classification problem.

The original deposit database comes in two parts: (1) a full database containing 41,188 transactions, and (2) 10 % samples containing 4,119 transactions. From the former, we extract the rest 90 % of the transactions, i.e. 37,069 transactions, to create a training sample. The latter is used for validation. There are 11.3 % subscriptions of the deposit and 88.7 % rejections.

Since the deposit database comes in pre-specified training and validation samples, proposal of this article is to obtain a set of input explanatory variables that best suit with the model and give maximal prediction performance. By holding the original article [16] as an example, a general statistical indicator Area Under Curve (AUC) [11] is used as a classification performance standard. Due to the promptness, logistic regression is employed as a classification algorithm, together with the optimization algorithm, which searches for best set of variables. In line with this, no cross-validation tests are performed, which act as a standard verification tool in classification theory [14]. Therefore, our proposal might come attractive for a narrow group of applicants, e.g. corporations, which appreciate a tiny increase of prediction performance on a known, pre-specified variables.

III. Background information

In this section, we briefly introduce readers the basic differential evolution as well as self-adaptive differential evolution.

A. Basic differential evolution

Differential evolution or simply DE is evolutionary algorithm proposed in 1995 [19] for solving various optimization problems. Main part of DE consists of three operators, i.e. mutation, crossover and selection. Individuals in DE are represented as real-valued vectors. All three basic operators in DE are explained in details in the next subsections:

1) Mutation in differential evolution: In DE mutation, two solutions are selected randomly and their scaled difference is added to the third solution, as follows:

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

where $F \in (0.0, 1.0]$ denotes the scaling factor that scales the rate of modification, while NP represents the population size and $r0,\ r1,\ r2$ are randomly selected values in the interval $1 \ldots NP$.

2) Crossover in differential evolution: DE employs a binomial (denoted as 'bin') or exponential (denoted as 'exp') crossover. The trial vector is built from parameter values copied from either the mutant vector generated by Eq. (1) or parent at the same index position laid i-th vector. Mathematically, this crossover can be expressed as follows [5], [6]:

$$w_{i,j}^{(t)} = \begin{cases} u_{i,j}^{(t)}, & \text{if } \text{rand}_j(0,1) \leq CR \vee j = j_{rand}, \\ x_{i,j}^{(t)}, & \text{otherwise}, \end{cases} \quad (2)$$

where $CR \in [0.0, 1.0]$ controls the fraction of parameters that are copied to the trial solution. The condition $j = j_{rand}$ ensures that the trial vector differs from the original solution $\mathbf{x}_i^{(t)}$ in at least one element.

3) Selection in differential evolution: Mathematically, the selection can be expressed as follows:

$$\mathbf{x}_i^{(t+1)} = \begin{cases} \mathbf{w}_i^{(t)}, & \text{if } f(\mathbf{w}_i^{(t)}) \leq f(\mathbf{x}_i^{(t)}), \\ \mathbf{x}_i^{(t)}, & \text{otherwise}. \end{cases} \quad (3)$$

The selection is usually called 'one-to-one', because trial and corresponding vector laid on i-th position in the population compete for surviving in the next generation. However, the better according to the fitness function will survive.

B. jDE algorithm

During the evolution of DE algorithms, many researchers faced the problem of proper parameter settings in DE. Many studies required a lot of effort in parameter tuning that is really long-lasting and tedious task. However, in 2006, Brest et al. [3] proposed an effective DE variant (jDE), where control parameters are self-adapted during the run. In this case, two parameters namely, scale factor F and crossover rate CR are added to the representation of every individual and undergo acting the variation operators. As a result, the individual in jDE is represented as follows:

$$\mathbf{x}_i^{(t)} = (x_{i,1}^{(t)}, x_{i,2}^{(t)}, ..., x_{i,M}^{(t)}, F_i^{(t)}, CR_i^{(t)}). \quad (4)$$

The jDE modifies parameters F and CR according to the following equations:

$$F_i^{(t+1)} = \begin{cases} F_l + \text{rand}_1 * (F_u - F_l) & \text{if } \text{rand}_2 < \tau_1, \\ F_i^{(t)} & \text{otherwise}, \end{cases} \quad (5)$$

and

$$CR_i^{(t+1)} = \begin{cases} \text{rand}_3 & \text{if } \text{rand}_4 < \tau_2, \\ CR_i^{(t)} & \text{otherwise}, \end{cases} \quad (6)$$

where $\text{rand}_{i=1...4} \in [0, 1]$ are randomly generated values drawn from uniform distribution in interval $[0, 1]$, τ_1 and τ_2 are learning rates, F_l and F_u lower and upper bound for parameter F, respectively.

IV. Proposed data mining method

The proposed data mining method consists of three steps:

- feature extraction,
- feature selection, and
- logistic regression.

In first step, the feature are extracted from the observed database together with particular attributes and corresponding domains of values. The feature selection is implementation of the DE for FS using threshold mechanism. Finally, the logistic regression was used for classification of features. In summary, the pseudo-code of the proposed data mining method is presented in Algorithm 1.

Algorithm 1 Proposed data mining method

Input: DE population $\mathbf{x_i} = (x_{i1}, \ldots, x_{iM}, F, CR, TH)^T$ for $i = 1 \ldots Np$, MAX_FE. and $\mathbf{Listoffeatures}$
Output: The $best$ model with selected features based on best solution

```
DE.init();
while termination_condition_not_meet do
    solution = DE.get_best_solution();
    fitness = eval_logistic_regression();
    DE.generate_new_solution(fitness);
end while
best = create_model(DE.get_best_solution());
```

In the remainder of the paper, the particular steps of the proposed data mining method are described in details.

A. Feature extraction

Domain analysis of the deposit database reveals that there are 20 explanatory variables and one dependent variable that are treated as features. Table I lists the explanatory variables.

The following variables (i.e., features) can be extracted from data in the mentioned table. First, basic client data is entered, e.g. age, job, marital status and type of education [16]. Next, three peculiar personal financial data are questioned, i.e. if a client owns a credit default, or has a housing/personal loan. Following attributes refer to last communication, i.e. what is the preferred type of communication with client, whether the cellular or standard telephone and which month and day has the client last time been contacted and long the call has been. Additionally, total number of contacts during ("campaign") and before ("previous") the campaign, number of days past since the last communication ("pdays") and client's decision from the last contact ("poutcome") are recorded as well. Finally, social and economic attributes, i.e. employment variation rate monthly ("emp.var.rate"), consumer price index monthly ("cons.price.idx"), euribor 3 month rate daily ("euribor3m") and number of employees quarterly ("nr.employed") are added.

Listed explanatory variables fall into one of the two types: numerical and categorical. The former are quantitative and can therefore be easily pre-processed, or transformed. Numerous operations may be applied to them, e.g. arithmetics, ordering, rationing, normalizing and scaling. In this article, we use the normalization operation. Categorical variables are qualitative and offer less chances of pre-processing. Moreover, they can only be discretized by creating and assigning dummy variables (dummification). Consequently, only few arithmetic operations can be applied to them, e.g. frequency count and histogram plot.

Normalization and dummification operations are processed as follows: by normalization, minimum and maximum val-

TABLE I: List of explanatory and dependent variables in original database.

No.	Explanatory variable	Type of explanatory variable	Range of the explanatory variable
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

ues of each explanatory variable are determined. Those are then used to transform original values between interval [0,1]. Minimum value is therefore represented by 0, while the maximum by 1. Dummification on the other hand begins by counting the number of unique instances n of the original explanatory variable, e.g. "yes", "no", "unknown" (in this case $n = 3$). Next, new variables, called dummy variables, are created for each explanatory variable. There have been created n dummy variables for each explanatory variable, e.g. "dummy_yes", "dummy_no" and "dummy_unknown". Last, each datapoint of the original variable is presented as 0 or 1, e.g. "yes" is presented as "dummy_yes=1", "dummy_no=0" and "dummy_unknown=0". Dummification by transforming original qualitative variable includes additional qualitative knowledge to the model, but enlarges the number of variables. In line with this, dummification of original variables improves predictive performance. However, care and understanding must be taken when manually including them into the model. Employing the normalization and dummification techniques enables the database to be fitted. However, special care are needed for two numerical variables, e.g."age" and "pdays". Although "age" is numerical variable, it is dummified as follows: 8 age groups are created with a range of 8 years, starting by 17 years. Actual age variable is then ordered into one of the age group by assigning 1 to that dummy variable and 0 to the rest. Dummification of age variable is known as a general remediation in econometrics community. Variable "pdays" represents the number of days from the last contact. In our case, this variable has been simplified by taking 0 as client had not been and 1 as client had been previously contacted.

B. Feature selection

Feature selection is implemented using the jDE algorithm that needs two modifications, as follows:

- representation of individuals,
- fitness function, and
- local search heuristic.

In the remainder of the paper, the modifications are discussed in details.

1) Representation of individuals: Individuals in jDE are presented as real-valued vectors:

$$\mathbf{x}_i^{(t)} = (x_{i,0}^{(t)}, \ldots, x_{i,M}^{(t)}, F_i^{(t)}, CR_i^{(t)}, TH_i^{(t)}), \tag{7}$$

for $i = 0, \ldots, Np$, where each feature $x_{i,0}^{(t)}$ for $i = 0, \ldots, M$ is drawn from the interval $[0, 1]$, $F^{(t)}$ and $CR^{(t)}$ are jDE control parameters, and $TH^{(t)}$ determines if the corresponding feature is present or absent in the solution. This mapping (also genotype-phenotype mapping in EA) can mathematically be expressed as follows:

$$a_{i,j}^{(t)} = \begin{cases} 0, & \text{if } x_{i,j}^{(t)} \leq TH^{(t)} \\ 1, & \text{otherwise,} \end{cases} \tag{8}$$

where vector $\mathbf{a}_i$ presents so-called attendance matrix determining the presence/absence of the observed j-th feature in the i-th solution. However, the 1 value means that the feature is present and 0 that it is absent in the solution.

A reason for introducing the threshold is founding that the majority of the transaction outcomes in deposit database, where each transaction has only two outcomes (true or false), are not distributed uniformly. This means that the majority of true/false outcomes are not distributes fifty fifty, but indicates bias toward some values. Actually, this value is unknown in advance, but needs to be found experimentally. In our study, this value expressed by the threshold, while looking for its best value is left to jDE.

2) Fitness function: The quality of the solution proposed by jDE for FS using threshold method is estimated using the complex calculation consisting of the following steps:

- logistic regression (modelling),
- validation (simulation),
- prediction coefficient mapping,
- building of the confusion matrix.

In the first step, the attendance vector $\mathbf{a}_i$ is applied to logistic regression (Logit model) in order to build the regression model. The result of modelling is the regression coefficient vector $\mathbf{b}_i$, which is validated on validation sample in order to obtain the probability vector $\hat{\mathbf{y}}_i$. The elements of this vector are drawn from the interval $y_{i,j} \in [0, 1]$. Then, the probability vector needs to be mapped to binary prediction vector $\mathbf{z}_i$.

The motivation for transforming the analogue probability vector $\hat{\mathbf{y}}_i$ into binary prediction vector $\mathbf{z}_i$ lies in the ability to compare the latter with true validation outcomes vector $\mathbf{y}$. For example, transformed predicted success (subscription), denoted by 1, or failure (reject), denoted by 0, can be compared to known binary outcome.

Transformation (discretization) of probability vector $\hat{\mathbf{y}}_i$ is done using the level $L_i \in [0, 1]$. Probabilities lower or equal than set level are assigned as 0, while probabilities higher than level, as 1. Level L_i is not set determinedly, but automatically adapts to maximize prediction performance during the local search heuristic, found in Section IV-D.

In the last step, prediction vector $\mathbf{z}_i$ is compared to true validation outcomes $\mathbf{y}$ to form a confusion matrix, i.e. matrix that separates correct predictions from wrong in a tabular way. Following four elements are included in the confusion matrix: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). All four elements are exhibited in Tab. II.

TABLE II: Exhibition confusion matrix.

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

True positive and true negative are correct predictions of subscripted or rejected deposits. False positive presents the type I error, which means that model predicts subscription, although it was not subscripted in real life. Type II error is its opposite, i.e. false negative, which means that model does not predict subscription, although it was subscripted in real life. Using the four basic elements, classification performance indicator AUC and five basic ratios: sensitivity, specificity, positive predicted value, negative predicted value and accuracy can be derived [1]. Using the AUC indicator, a minimization-wanted fitness function is expressed as:

$$f(trial\_solution) = 1 - \text{AUC}_{best} \tag{9}$$

As seen in Eq. 9, $trial\_solution$ is tested on the Logit model to obtain maximally optimized AUC_{best} and determine the fitness function value $1 - \text{AUC}_{best}$.

C. Logistic regression

Logistic regression (Logit) is a deterministic classification method, invented by David Cox in [4]. It is a special variant of ordinary regression, where binary (0 or 1), or limited, dependent variable appear, rather than usual data. Binary values denote success in case of 1 and failure in case of 0. Regression or fit is executed using the Maximum Likelihood Estimation (MLE) approach. However, treatment of regression coefficients and forecasting are a bit different. Obtained regression coefficients impact the probability of success, which is calculated as natural logarithm of odds ratio, shown in Eq. 10:

$$L_i = ln\left(\frac{P_i}{1 - P_i}\right) \tag{10}$$

where P_i is the probability of success and $1 - P_i$ is probability of failure. Its division gives the odds ratio.

D. Local search heuristic

Variation of the level L_i heavily affects prediction performance. Even well-derived logit model and correct regression coefficients, may cause prediction outcomes to be very biased with wrong settings of the level L_i. It is however our goal to maximize prediction performance. In line with this, we employ a local search heuristic to identify the optimal level $L_i^{(opt)}$ by varying the level L_i and checking the AUC score sequentially. The level which maximizes the AUC score, is then used in a further classification process. The outline of the local search heuristics is outlined in Alg. 2.

Algorithm 2 Local search heuristic

Input: probability vector $\hat{\mathbf{y}}$, validation vector $\mathbf{y}$
Output: optimal level $L^{(opt)}$

```
L = 0;
AUC_best = 0;
while L <= 1 do
    z = discretize(ŷ, L); // discretize ŷ using L
    AUC = calculate_AUC(z, y); // two discrete vectors
    if AUC_best < AUC then
        L^(opt) = L
    end if
    L += 0.005;
end while
return L^(opt);
```

For an i-th solution, initialization of level L and AUC_{best} is first required. Process then continues to iterative while loop, where discretization of probability vector $\hat{\mathbf{y}}$ using the initial level L happens. Binary prediction vector $\mathbf{z}$ is obtained as a result, which can be directly compared to the validation vector $\mathbf{y}$. Using the two, current AUC score can be calculated. It can then be checked, whether the latter improves the best solution AUC_{best} and if it does, the best solution AUC_{best} is updated. Lastly, current level L gets incremented and the loop repeats. When the termination condition is met, the optimal level $L^{(opt)}$ is returned as an output argument. The process can be depicted by a schematic diagram, shown in Fig. 1.

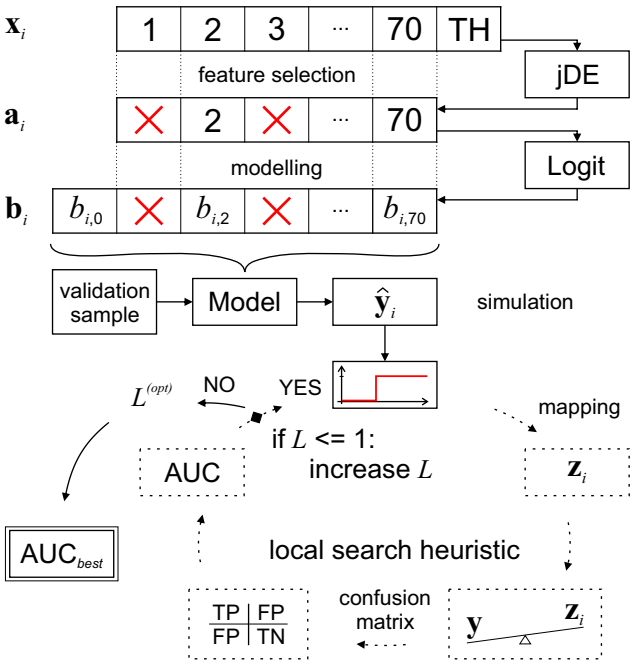

Fig. 1: Schematic diagram of the cost function.

Although local search heuristic, together with the optimization, affect overall prediction performance heavily, local search heuristic does not affect performance of jDE directly. Practically, jDE comes in a first stage and local search heuristic in a second. The purpose of the first stage is to find the set of input explanatory variables which suit to the model at most and the purpose of the second to find the optimal level $L^{(opt)}$ which will maximize prediction performance. The interaction of testing and evaluating trial solutions on the model subsequently creates a feedback system.

V. Experiments and Results

The purpose of our experimental work was to validate the results of the jDE for FS using threshold method applied on: (1) original and (2) reduced deposit database. In line with this, we focused on analysis of the results obtained by the jDE for FS applied on original and reduced database. In the first test, original database with all included features has been tested for prediction using the Logit model. Thus, the correlation analysis was performed, confusion matrix was built and overall prediction results were preserved. The second test was run on the reduced database, where confusion matrix was built as well and prediction results were compared to the results obtained on the original database.

Interestingly, the proposed data mining method is self-adapted. This means that all parameters of applied jDE algorithm are part of individual representation and undergo acting the variation operators, while the searching for the optimal level by local search heuristic is also performed automatically. Therefore, the initial setup of jDE parameters as used during experimental work is presented in Table III.

TABLE III: Parameter settings of jDE.

Parameter	Value
Initial scaling factor F	0.5
Initial crossover ratio CR	0.9
Self-adaptive learning rate τ	0.1
Population size NP	50

The quality of solutions were estimated according to following measures: number of features included, AUC score, level L_i, sensitivity, specificity, positive predictive value, negative predictive value, type I error, type II error and accuracy. In the remainder of the paper, the mentioned experiments are discussed in details.

A. Results on the original database

This test was conducted on the original database with all 70 features. At first, correlation analysis was performed to obtain a basic outline of the database. The results of the analysis are displayed graphically in Fig. 2, where the higher the correlation between the variables, the more red colored are the cells. As can be seen from the figure, significant correlation is found among variables "pdays", "previous", "emp.var.rate", "Euribor", "nr.employed", between the dummy variables of age, especially at younger years, months of years and days of the week. Weaker correlation is found for variables "campaign", between jobs, especially "entrepreneur", "housemaid" and "unemployed", as well as "illiterate education". Therefore, we expect that these variables would be omitted during the FS.

Then, the Logit model has been fitted using the entire training sample of original database, and prediction was executed using the validation sample. As a result, confusion matrix is obtained as presented in Tab. IV.

TABLE IV: Confusion matrix of original database.

	True YES	True NO
Predicted YES	424	652
Predicted NO	27	3016

When commenting results of the prediction presented in the confusion matrix, we are mainly interested in relative indicators. For example, type I and type II errors can be treated as follows: in absolute way, the former totals to 652 and the latter totals to 27. However, these two values do not speak for themselves, but may become to do so, when they are observed relatively: 21.62 % of type I error and 6.38 % of type II error. The former error may seem a bit high, therefore we are interested to decrease this.

B. Results on the reduced database

Next, same experiments were run on the reduced database. The variables selected after running the proposed jDE for FS algorithm are illustrated in Tab. V. As supposed, "pdays", "previous", "emp.var.rate", "cons.price.idx", "cons.conf.idx", "euribor3m" variables are found in the reduced database. Variables "age", "month", "day_of_week" have been found in reduced database too, but not in their entirety. As supposed, "housemaid" and "entrepreneur" job instances were omitted,

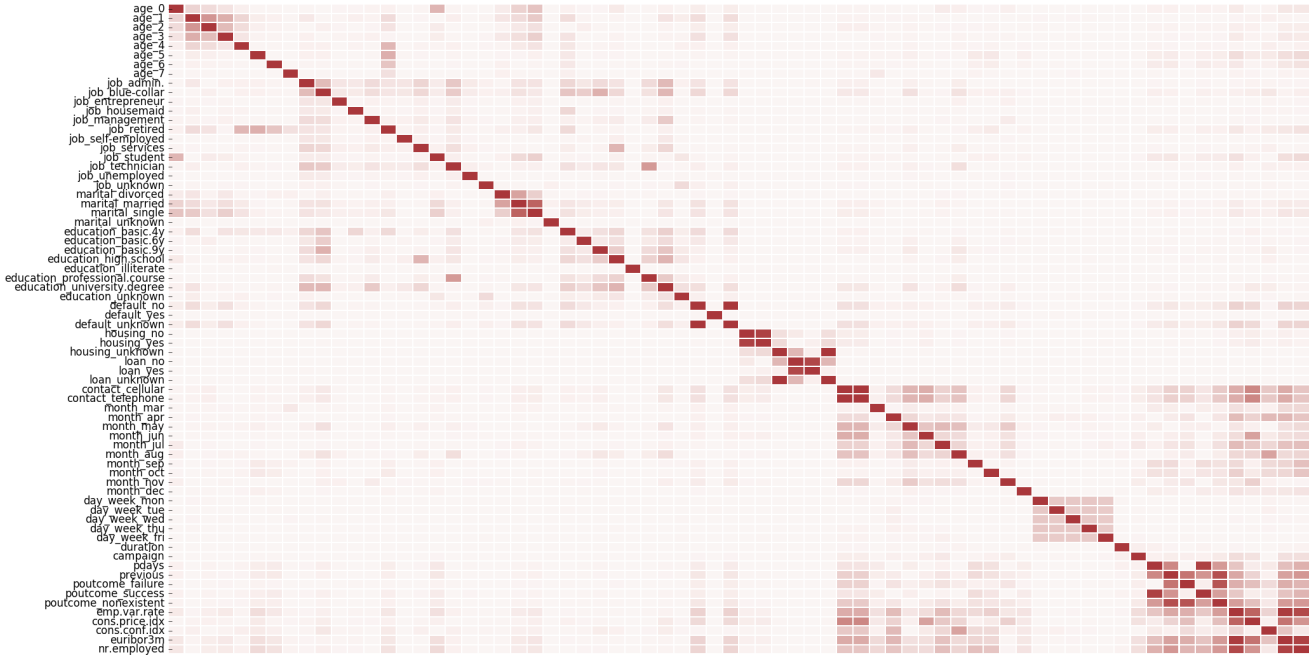

Fig. 2: Correlation analysis of the original database.

while "unemployed" remained in the database. Variables "illiterate education" and "nr.employed" were omitted as well.

TABLE V: List of explanatory variables in reduced database.

No.	Selected variables	Omitted variables
1.	age (26-34, 35-43, 62-70, 80+)	age (17-25, 44-52, 53-61, 71-79)
2.	job ("management", "self-employed", "services", "unemployed")	job ("administrator", "blue-collar", "entrepreneur", "housemaid", "retired", "student", "technician", "unknown")
3.	marital ("married")	marital("divorced", "single", "unknown")
4.	education ("basic.4y", "basic.6y", "high.school", "professional.course", "university.degree")	education ("unknown", "basic.9y", "illiterate")
5.	housing ("no", "unknown")	housing ("yes")
6.	contact ("cellular")	contact ("telephone")
7.	month ("March", "April", "June", "August", "October", "November")	month ("May", "July", "September", "December")
8.	day_of_week ("Tuesday", "Wednesday", "Thursday","Friday")	day_of_week ("Monday")
9.	duration	default
10.	campaign	loan
11.	pdays	nr.employed
12.	previous	
13.	poutcome	
14.	emp.var.rate	
15.	cons.price.idx	
16.	cons.conf.idx	
17.	euribor3m	

Table VI presents the results of reduced database prediction, where by watching absolute results, "predicted NO" error has

TABLE VI: Confusion matrix of reduced database.

	True YES	True NO
Predicted YES	423	576
Predicted NO	28	3092

increased from 27 to 28, thereby increasing the type II error. However, "true NO" error has significantly been decreased from 652 to 576. Therefore, type I error was heavily decreased. In relative terms, type I error decreased to 18.63 % and type II error increased to 6.62 %.

In summary, the overall prediction results are applicable in Table VII, from which it can be seen that these consist

TABLE VII: Prediction results compared for both databases.

	Original database	Reduced database
No. of features	70	38
AUC score	0.88119	0.89044
Level L_i	0.14	0.16
Sensitivity	0.94013	0.93792
Specificity	0.82225	0.84297
Positive predictive value	0.39405	0.42342
Negative predictive value	0.99113	0.99103
Type I error	0.21618	0.18629
Type II error	0.06378	0.06620
Accuracy	0.83515	0.85336

of two prediction results: predictions on original and reduced database. In the remainder of the section, those results are discussed more detailed.

C. Discussion

The most relevant information about the quality of jDE for FS is the reduction of number of features. The proposed

solution has lowered the number of features from 70 to 38, thus almost half of the original features were omitted in the reduced database. The AUC score, as a measure of optimization, has successfully improved after the feature selection. Reduction of dimensions also positively affected the positive predictive value and specificity scores. Both of them were improved significantly, for more than 2 %. The accuracy as a general measure has been improved for a bit less than 2 %, which is another beneficent fact. On the other hand, sensitivity and negative predictive value have decreased after the optimization, but a minimal change is observed. We can therefore conclude, that benefits of the optimization and feature selection outweigh costs.

All listed facts indicate that the self-adaptive differential evolution can successfully be applied to FS problem. Furthermore, optimization in general minimizes/maximizes the fitness function and accordingly, any statistics can be optimized, e.g. the type I error to be minimized or sensitivity to be maximized. The proposed jDE for FS using threshold method is therefore universal. Additionally, let us mention that the results strongly confirmed the results of the study performed in [8]. Additionally, we proofed that dummification of variable "age" helps to increase predictive performance.

VI. Conclusion

A novel self-adaptive jDE for FS using the threshold mechanism has been proposed in this paper. The proposed algorithm was applied to the deposit database, which has additionally been expanded using the dummy variables and used as an origin for FS optimization process. We have been optimizing one of the classification statistics, i.e. AUC score, by sampling entire database and comparing its results to the origin. It was found out, that the results of the classification process on database using reduced for almost half of the original features overcomes the results of the same classification on the original database.

On the other hand, the proposed novel jDE for FS using threshold method is also designed as an universal tool, which allow users the opportunity to optimize the classification process according to more classification statistics. Finally, the users therefore decide, which statistic is the most appropriate for their needs.

The proposed algorithm importantly differs from the existed FS methods due to application of a feedback loop. With constant interaction of modelling and simulation phases this means that database trial solutions are directly evaluated on the model and quality of trial solution is used in the further evolutionary process. In the end, optimal variables for specific model are obtained.

Typical feature selection algorithms nowadays actually perform statistical tests and decide how to substitute the original database by using the fewer variables. Additionally, those methods can control the number of features, which comes as a leak of our FS method. Nevertheless, constraint optimization could be implemented to satisfy this criteria in the future. Moreover, we would like to test and compare jDE algorithm with the other evolutionary algorithms by solving the same problem.

References

[1] Anthony K Akobeng. Understanding diagnostic tests 1: sensitivity, specificity and predictive values. *Acta paediatrica*, 96(3):338–341, 2007.

[2] Michael J Berry and Gordon Linoff. *Data mining techniques: for marketing, sales, and customer support*. John Wiley & Sons, Inc., 1997.

[3] 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.

[4] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 215–242, 1958.

[5] S. Das and P. N. Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31, Feb 2011.

[6] Swagatam Das, Sankha Subhra Mullick, and P.N. Suganthan. Recent advances in differential evolution an updated survey. *Swarm and Evolutionary Computation*, 27:1–30, 2016.

[7] Liran Einav and Jonathan Levin. Economics in the age of big data. *Science*, 346(6210):1243089, 2014.

[8] Dušan Fister, Iztok Fister, and Timotej Jagrič. Artificial intelligence in banking - a universal tool? *Bančni vestnik : The journal of money and banking*, 67(6):12–21, 2018.

[9] Iztok Fister, Dušan Fister, and Simon Fong. Data mining in sporting activities created by sports trackers. In *Computational and Business Intelligence (ISCBI), 2013 International Symposium on*, pages 88–91. IEEE, 2013.

[10] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[11] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.

[12] Saint John Walker. Big data: A revolution that will transform how we live, work, and think, 2014.

[13] George G Judge, Rufus Carter Hill, William Griffiths, Helmut Lutkepohl, and Tsoung Chao Lee. Introduction to the theory and practice of econometrics. 1982.

[14] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[15] Jay Lee, Hung-An Kao, and Shanhu Yang. Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia Cirp*, 16:3–8, 2014.

[16] 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.

[17] Wullianallur Raghupathi and Viju Raghupathi. Big data analytics in healthcare: promise and potential. *Health information science and systems*, 2(1):3, 2014.

[18] Douglas Rodrigues, Luis AM Pereira, TNS Almeida, João Paulo Papa, AN Souza, Caio CO Ramos, and Xin-She Yang. Bcs: A binary cuckoo search algorithm for feature selection. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 465–468. IEEE, 2013.

[19] 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.

[20] Xiangyang Wang, Jie Yang, Xiaolong Teng, Weijun Xia, and Richard Jensen. Feature selection based on rough sets and particle swarm optimization. *Pattern recognition letters*, 28(4):459–471, 2007.

[21] Bing Xue, Mengjie Zhang, and Will N Browne. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE transactions on cybernetics*, 43(6):1656–1671, 2013.

[22] 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.