Vietnam Journal of Computer Science (2018) 5:107–121
https://doi.org/10.1007/s40595-018-0107-y

REGULAR PAPER

Three local search-based methods for feature selection in credit scoring

Dalila Boughaci[1] · Abdullah Ash-shuayree Alkhawaldeh[2]

Received: 30 July 2016 / Accepted: 10 May 2018 / Published online: 28 May 2018

Abstract

Credit scoring is a crucial problem in both finance and banking. In this paper, we tackle credit scoring as a classification problem where three local search-based methods are studied for feature selection. The feature selection is an interesting technique that can be launched before the data classification task. It permits to keep only the relevant variables and eliminate the redundant ones which enhances the classification accuracy. We study the local search method (LS), the stochastic local search method (SLS) and the variable neighborhood search method (VNS) for feature selection. Then, we combine these methods with the support vector machine (SVM) classifier to find the best described model from a dataset with the correct class variable. The proposed methods (LS+SVM, SLS+SVM and VNS+SVM) are evaluated on both German and Australian credit datasets and compared with some well-known classifiers. The numerical results are promising and show a good performance in favor of our methods.

Keywords Credit scoring · feature selection · variable selection · classification · support vector machine · optimization · local search · stochastic local search · variable neighborhood search

1 Introduction

Credit scoring (CS) is an important process for banks as they have to be able to distinguish between good and bad applicants in terms of their creditworthiness. CS is the process of evaluating the creditworthiness of applicants to decide if the credit will be granted or not [33]. The evaluation process is usually based on some variables related to applicants such as historical payments, guarantees, default rates, etc.

Several CS models are proposed in literature [40]. Among them we find the following ones: Linear regression statistical methods [20] that permit to analyze data and verify if the credit can be granted to a given applicant or not. Discriminant analysis and logistic regression, which are one of the most broadly established statistical techniques used to classify applicants as "good" or "bad" [44]. Decision trees [42], CART (Classification and Regression Trees) [4] and Bayesian networks [16,26] are used to classify data in credit scoring models.

More sophisticated methods based on computational intelligence are also studied for developing credit scoring models. As examples, we give: the neural networks [15,38], the k-nearest neighbor classifier [21], the support vector machines (SVM) [3,24], the ensemble classifiers [2], the genetic programming [1] and the evolution strategies [31]. In [27], authors propose an interesting quantification method for credit scoring. They use a categorical canonical correlation analysis to determine the relationship between categorical variables. In [22], authors propose a feature selection method based on quadratic unconstrained binary optimization (QUBO) algorithm. In [6] authors propose a cooperative classification system based on agents for CS.

On the other hand, meta-heuristics are a kind of computational techniques that have been used successfully for solving several optimization problems in several areas. The meta-heuristic approaches can be divided into two main categories: population-based methods and single solution-oriented methods [8]. The population-based methods called

✉ Dalila Boughaci
dalila_info@yahoo.fr; dboughaci@usthb.dz

Abdullah Ash-shuayree Alkhawaldeh
alkwaldhh@yahoo.com

[1] LRIA/Computer Sciences Department, University of Sciences and Technology Houari Boumediene (USTHB), El-Alia, Beb-Ezzoaur, BP 32, 16111 Algiers, Algeria

[2] Department of Accounting, Faculty of Economics and Administrative Sciences, The Hashemite University, Zarqa, Jordan

also evolutionary methods maintain and evolve a population of solutions while the single solution-oriented methods work on a current single solution. Among the evolutionary approaches for optimization problems, we mention the well-known genetic algorithms [17], evolutionary computation [34], and harmony search [36,45]. Among the single solution-oriented methods, we cite stochastic local search (SLS) [25], simulated annealing (SA) [28], tabu search (TS) [18] and variable neighborhood search (VNS) [23,32].

In this paper, we are interested in feature selection for credit scoring (CS). We tackle CS as a classification problem where three single solution-oriented meta-heuristic methods are studied for the feature selection. The feature selection is a technique that permits to eliminate the redundant variables and keep only the relevant ones. This manner can reduce the size of the dataset and simplify the data analysis. The feature selection has been applied in data classification to enhance the classifier performance and to reduce data noise [37,41].

We propose to study local search, stochastic local search and variable neighborhood search for feature selection in credit scoring. The proposed feature selection is then combined with a support vector machine to classify the input data. The three variants of the proposed approach (LS+SVM, SLS+SVM and VNS+SVM) are implemented and evaluated on two well-known datasets which are: Australian and German Credit datasets.

The rest of this paper is organized as follows: Sect. 2 gives a background on some concepts used in this study. Sect. 3 discusses the proposed methods for credit scoring. Sect. 4 gives some experimental results. Finally, Sect. 5 concludes and gives some perspectives.

2 Background

The aim of this section is to explain the credit scoring problem and give an overview of feature selection and some basic concepts on support vector machine used in this study.

2.1 Problem definition and formulation

Credit scoring is an important issue in computational finance. CS is a set of decision models that can help lenders in the granting of applicant credit. Based on such models, lenders can decide whether applicant is eligible for credit or not [33]. To build CS models, we often exploit information about the applicant, such as: the age, the number of previous loans, default rates, etc. This information is called variables, attributes or features. The CS models may allow lenders to distinguish between "good" and "bad" applicants. It can give also an estimation of the probability of default.

More precisely, the CS problem can be stated as follows [22,33]:

Let us consider a set of variables where each variable may be: a numeric or a category. For instance, the variable bank balances is a numerical feature that can be represented as an integer or a numeric. Other examples of numerical variables in CS can be the applicant age, the interest rates and so on. Categorical variables are called qualitative variables. We give as example a credit history or a geographic region code. The qualitative variables may also include "missing" (not specified values).

An instance or a sample is defined as an observation for each variable and an outcome represented the label class.

The classification is the problem of discovering the class of an observation. We have as input a set of independent variables and as output the label class. The objective is then to maximize the classification accuracy rate.

In CS, the variables are the set of features that describe the applicants profile and the financial data. The credit data can be divided into two main classes: "good" applicants (where the label class is Y=1) who paid their loan back, and "bad" applicants (Y=0) who defaulted on their loans.

The credit data is then a set of applicants to be classified into two classes: "bad" (Y=0) or "good" (Y=1).

According to [22], the CS problem can be formulated as follows:

- The credit data can be organized as a matrix D of m rows and n columns where n is the number of features and m is the number of past applicants.

$$\mathbf{D} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

  For example, the first row $a_{11}, a_{12}, \ldots a_{1n}$ represents the specific data values of the first applicant where a_{1i} is the value of feature i of the applicant number 1.
- The creditworthiness of new applicant can be determined by using the data on past applicants recorded in the matrix D.
- The decision is then represented as a vector Y with m elements where each element y_i has two possible values 0 or 1. An element y_i receives the value 1 when the applicant i is accepted, 0 otherwise.

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$$

- The classification then is the problem of determining the decision vector Y that indicates the accepted applicants (y_i=1) or the rejected ones (y_i=0).

Classification plays an important role in CS. However, before launching the classification task, a pre-processing is needed. Data preparation is an interesting step. It permits to prepare properly and accurately the data. This allows getting efficient models that can help the creditor in making a correct decision. In this study, we are interested in feature selection for CS. The aim is to select from the original set of n variables a subset of K variables to be used in the decision-making. The feature selection is a pre-processing that can be launched before classification task. The details about this technique are given in the next subsection.

2.2 Feature selection

Feature selection called also attribute selection or variable selection is the process of removing the redundant attributes that are deemed irrelevant to the data mining task. It is an important step that may be launched before classification to eliminate irrelevant variables. This process can improve the classification performance and accelerate the search process [10,12,35–37,41].

Several methods have been studied for feature selection. These methods can be divided in two main methods: the wrapper methods [29] and the filter methods [30].

- The wrapper methods use data mining algorithm for searching the optimal set of attributes while filter methods eliminate and filter out the undesirable attributes before starting the classification task.
- The filter methods usually use heuristics instead of machine-learning algorithms used by the wrapper methods [29]. The machine learning algorithm selects the optimal set of attributes with high classification accuracy. However, the wrapper methods are time consuming compared to filter methods because the machine-learning algorithm is run iteratively while searching the set of best attributes.

2.3 Support vector machine

In this study, we are interested in supervised learning technique that finds the best described computer model from a dataset with the correct class variable. Support vector machine is one of the most well-known machine-learning techniques. The technique was proposed by Vladimir Vapnik for classification and regression [11,16,24].

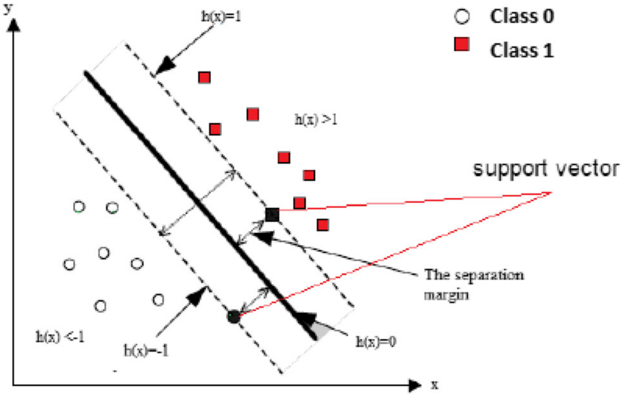

Fig. 1 Linear separation of two classes 0 and 1 in two-dimensional space with SVM classifier [5]

SVM classification method learns from a training dataset and attempts to generalize and make correct predictions on novel data.

Let us consider a test sample or a novel data to be classified. The problem is to predict whether the test data belongs to one of the considered classes. The training data is a set of examples of the form $\{x_i, y_i\}(i = 1, ..., l)$. x_i are called input vector. Each input vector has a number of features. $y_i \in \{0, 1\}$. y_i are the response variables called also labels. These input vectors are paired with corresponding labels to find the correct class variable.

As shown in Fig. 1, in *two-class* supervised learning and when data are linearly separable, SVM can separate the data points into two distinct classes where, in class "good", $y = 1$ and in class "bad", $y = 0$. $h(x)$ is the decision function.

Support vector machine are also called Kernel methods, where the kernel represents similarity measures of examples. We find the following kernel functions: Linear, polynomial, Laplacian, sigmoid and Gaussian called also radial basis function (RBF) [16,24]. An interesting library for support vector machines (LIBSVM) with open source code can be found available online [13,14,43].

3 Proposed approaches for feature selection

In this section, we propose three local search-based methods for feature selection. In the following, we start with the feature vector solution, the accuracy measure and then we give details on the three local search methods for feature selection. The feature selection is combined then with a support vector machine to classify data.

3.1 The feature vector solution representation

The aim of the feature selection is to search for an optimal set of variables or features to be used with the SVM classifier in the classification task.

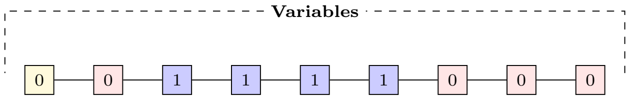

Fig. 2 The feature vector solution representation

A solution can be represented as a binary vector which denote the variables present in the dataset, with the length of the vector equals to n, where n is the number of variables. More precisely, a solution is a set of selected variables. To represent such a solution we use the following assignment: if a variable is selected in a solution, the value 1 is assigned to it, a value 0 is assigned to it otherwise. For example, Fig. 2 represents a vector solution. We have a dataset of nine variables where the third, the fourth, the fifth and the sixth variables are selected (bits with value 1).

3.2 Accuracy measure

We used the classification accuracy to measure the quality of a solution called also *fitness*. We used also the cross-validation standard way to measure the accuracy of a learning scheme on a dataset. The classification accuracy is computed as the ratio of number of correctly classified instances to the total number of instances using the formula (1)

$$\text{Fitness} = \text{Accuracy} = \frac{tp + tn}{tp + fn + fp + tn} \qquad (1)$$

where

– tp is the true positive and tn is the true negative,
– fp is the false positive and fn is the false negative.

3.3 Feature selection step

In this work, we study three local search-based meta-heuristics for feature selection. The first is a local search (LS). The second is a stochastic local search (SLS) and the third is a variable neighborhood search (VNS). The local search-based feature selection searches for the best variables set. Then, the support vector machines (SVM) classifies the input data in the reduced dataset, corresponding to the subset of selected variables represented by the feature vector solution generated by the local search method. As already mentioned, the support vector machine (SVM) is a machine-learning classifier that permits to find an optimal separating hyper-plane. It uses a linear hyper-plane to create a classifier with a maximum margin [26]. In the following, we give details on LS, SLS and VNS for feature selection.

3.3.1 Local search method

The local search method (LS) is a hill-climbing technique [25]. LS starts with a random solution x and tries to find better solutions in the current neighborhood. The neighboring solution x' of the solution x is obtained by modifying one bit. Neighborhood solutions are generated by randomly adding or deleting a feature from the solution vector. For example, if n the number of variable equals to 7 and the current solution vector is x = 1111111, then the possible neighbor solutions can be : {**0**111111, 1**0**11111,11**0**1111,111**0**111,1111**0**11,11111**0**1, 111111**0**}.

Among the neighbors, we select for the next iteration the one with the best accuracy. The process is repeated for a certain maximum number of iterations fixed empirically. LS method is sketched in Algorithm 1.

Algorithm 1: LS with SVM

```
Data: $n$ the number of variables, $max\_iterations$
Result: A set of selected variables $x^{best}$
Start with an initial random solution $x$
// Evaluation through 10-fold cross-validation
Evaluate the fitness of $x$ noted $f(x)$ by calling the SVM algorithm on $x$
for ($I = 1$ to $max\_iterations$) do
    Generate the neighborhood solutions of $x$;
    Evaluate the fitness of each neighbor $x$ by calling the SVM algorithm on $x$
    $x_i^{new}$= pick a best neighbor solution according to the fitness value
    if ( $f(x_i^{new}) > f(x_i)$) then
        $x_i=x_i^{new}$;
Return the set of selected significant variables and the best classification
accuracy.
```

3.3.2 Stochastic local search method

The stochastic local search (SLS) used here is inspired from the one used in [7]. SLS is a local search meta-heuristic which has been already studied for several optimization problem such as satisfiability and optimal winner determination problem (WDP) in combinatorial auctions [8,9]. SLS starts with an initial solution generated randomly. Then, it performs a certain number of local steps that combines diversification and intensification strategies to locate good solutions.

– Step 1: The diversification phase selects a random neighbor solution .
– Step 2: The intensification phase selects a best neighbor solution according to the accuracy measure.

The diversification phase is applied with a fixed probability $wp > 0$ and the intensification phase with a probability $1 - wp$. The wp is a probability fixed empirically. The process is repeated until a certain number of iterations called $max\_iterations$ is reached. The SLS+SVM method for classification is sketched in Algorithm 2. The proposed SLS+SVM for feature selection starts with a randomly initial solution and then tries to find a good solution in the whole neighborhood in an iterative manner. The SVM classifier is built for each candidate solution constructed by SLS method. The solution is evaluated by the cross-validation method. The SLS process permits the selection of potential attributes that lead to good prediction accuracy. The objective is to find the optimal subsets of attributes by finding optimal combinations of variables from the dataset.

Algorithm 2: SLS with SVM

```
Data: $n$ the number of variables, $max\_iterations$, $wp$
Result: A set of selected variables $x^{best}$
Start with an initial random solution $x$
// Evaluation through 10-fold cross-validation
Evaluate the fitness of $x$ noted $f(x)$ by calling the SVM algorithm on $x$
for ($I = 1$ to $max\_iterations$) do
    r ← random number between 0 and1.
    if ( $r < wp$) then
        *Step 1
        $x_i^{new}$= a random neighbor solution
    else
        //*Step 2
        Generate the neighborhood solutions of $x$;
        Evaluate the fitness of each neighbor $x$ by calling the SVM algorithm
        on $x$
        $x_i^{new}$= pick a best neighbor solution according to the fitness value
        if ( $f(x_i^{new}) > f(x_i)$) then
            $x_i$=$x_i^{new}$;
Return the set of selected significant variables and the best classification
accuracy.
```

3.3.3 Variable neighborhood search method

The variable neighborhood search (VNS) is a local search meta-heuristic proposed in 1997 by Mladenovic and Hansen. Various variants of VNS have been proposed since then, but the basic idea is a systematic change of neighborhood combined with a local search [23,32]. In this work, we used four structures of neighborhood which are N1, N2, N3 and N4. At each iteration, we select among the four structures one randomly to create neighbor solutions.

- **N1**: where the neighbor solution x' of the solution x is obtained by modifying one bit as done with local search method. For example, if n the number of feature equals to 12 and x = 111111111111 is a current solution vector, then the possible neighbors in **N1** can be as follows: {**0**11111111111, 1**0**1111111111, 11**0**111111111, 111**0**11111111, 1111**0**1111111, 11111**0**111111, 111111**0**11111, 1111111**0**1111, 11111111**0**111, 111111111**0**11, 1111111111**0**1, 11111111111**0**}.

- **N2**: where the neighbor solution x' of the solution x is obtained by modifying two bits simultaneously. For example, if x = 111111111111 is a current solution vector, then the possible neighbors in **N2** can be as follows: {**00**1111111111, 1**00**111111111, 11**00**11111111, 111**00**1111111, 1111**00**111111, 11111**00**11111, 111111**00**1111, 1111111**00**111, 11111111**00**11, 111111111**00**1, 1111111111**00**} .

- **N3**: where neighboring solution x' of the solution x is obtained by modifying three bits simultaneously. For example, if we take the same x = 111111111111 as a current solution vector, then the possible neighbors in $N3$ can be : {**000**111111111, 1**000**11111111, 11**000**1111111, 111**000**111111, 1111**000**11111, 11111**000**1111, 111111**000**111, 1111111**000**11, 11111111**000**1, 111111111**000**}.

- **N4**: where neighboring solution x' of the solution x is obtained by modifying randomly one bit.

Like SLS, VNS starts with a randomly initial solution and then tries to find a good solution in the whole neighborhood in an iterative manner. The SVM classifier is called for each candidate solution constructed by VNS method to evaluate the accuracy rate. The process is repeated until a certain number of iterations called $max\_iterations$ is reached.

The best solution with a best accuracy rate is selected. The VNS algorithm for feature selection is sketched in Algorithm 3.

Algorithm 3: VNS with SVM

```
Data: $n$ the number of variables, $max\_iterations$
Result: A set of selected variables $x^{best}$
Start with an initial random solution $x$
// Evaluation through 10-fold cross validation
Evaluate the fitness of $x$ noted $f(x)$ by calling the SVM algorithm on $x$
for ($I = 1$ to $max\_iterations$) do
    k ← random number between 1 and 4.
    if ($k = 1$) then
        Apply $N1$ structure
        $x_i^{new}$= the best neighbor solution
    else
        if ($k = 2$) then
            Apply $N2$ structure
            $x_i^{new}$= the best neighbor solution
        else
        if ($k = 3$) then
            Apply $N3$ structure
            $x_i^{new}$= the best neighbor solution
        else
            $x_i^{new}$= a random neighbor solution
    if ( $f(x_i^{new}) > f(x_i)$) then
        $x_i$=$x_i^{new}$;
Return the set of selected significant variables and the best classification
accuracy.
```

Table 1 An instance of the german.data-numeric dataset

Variable number	Value	Scaled value	Variable number	Value	Scaled value
1	1.000000	− 1	13	1.000000	− 1
2	6.000000	− 0.941176	14	2.000000	1
3	4.000000	1	15	1.000000	− 1
4	12.000000	− 0.89011	16	0.000000	− 1
5	5.000000	1	17	0.000000	− 1
6	5.000000	1	18	1.000000	1
7	3.000000	0.333333	19	0.000000	− 1
8	4.000000	1	20	0.000000	− 1
9	1.000000	− 1	21	1.000000	1
10	67.000000	0.714286	22	0.000000	− 1
11	3.000000	1	23	0.000000	− 1
12	2.000000	− 0.333333	24	1.000000	1
Class					0

Table 2 An instance of Australian dataset with scaled values

Variable number	Value	Scaled value
A1	1	1
A2	22.08	− 0.749474
A3	11.46	− 0.181429
A4	2	
A5	4	0.538462
A6	4	− 0.25
A7	1.585	− 0.888772
A8	0	− 1
A9	0	− 1
A10	0	− 1
A11	1	1
A12	2	
A13	100	− 0.9
A14	1213	− 0.97576
Class		0

4 Experiments

All experiments were run on an Intel Core(TM) i5-2217U CPU@1.70 GHz with 6 GB of RAM under Windows 8—64 bits, processor x64.

4.1 The dataset normalization

The dataset normalization called also feature scaling is a mandatory preprocessing step before staring the classification task. This step is used to avoid variables in greater numeric ranges to dominate those in smaller numeric ranges. The feature values are linearly scaled to the range $[-1, +1]$ or $[0, 1]$ using formula (2), where X denotes the original value; X denotes the scaled value. MAX_a is the upper bound of the feature value a, and MIN_a is the lower bound of the feature value a.

In our study, we scaled the different feature values to the range $[-1, +1]$.

$$X^{'} = \left(\frac{X-MIN_a}{MAX_a-MIN_a}\right) \times 2 - 1. \qquad (2)$$

4.2 The dataset description

To evaluate the performance of the proposed methods for credit scoring, we considered both German and Australian credit datasets from UCI (University of California at Irvine) Machine Learning Repository[1]. The descriptions of the two credit datasets are given as follows:

1. The German credit dataset is a credit dataset proposed by the Professor Hans Hofmann from *Universit"at Hamburg*. The dataset consists of 1000 instances. There are two classes: **class 1** (worthy, 700 instances) and **class 0** (unworthy, 300 instances). We find on UCI, two versions of German dataset:
   - The original dataset *german.data* that contains categorical/symbolic variables. The number of variable is equal to 20 where 7 are numerical and 13 categorical.
   - The "german.data-numeric" dataset provided by Strathclyde University to be used with algorithms which cannot cope with categorical variables. The number of attributes is equals to 24 numerical attributes. In our experiments, we worked on the "german.data-numeric" dataset version.

   An example of an instance of "german.data-numeric" before and after the scaling process is given in Table 1.

[1] https://archive.ics.uci.edu/ml/datasets

Table 3 The results of 50 run of LS+SVM on German dataset

Run number	Accuracy %	Number of selected variables	Run number	Accuracy %	Number of selected variables
1	77.200	17	2	77.200	12
3	77.100	5	4	77.400	12
5	77.400	12	6	77.400	10
7	77.400	14	8	77.000	9
9	77.600	11	10	77.600	10
11	77.100	14	12	77.200	10
13	77.100	12	14	77.400	10
15	77.300	10	16	77.300	9
17	77.400	7	18	77.400	14
19	77.100	11	20	77.100	13
21	77.100	14	22	77.300	9
23	77.300	12	24	77.500	12
25	77.500	11	26	77.300	15
27	77.100	12	28	77.200	12
29	77.300	12	30	77.300	15
31	77.500	11	32	77.000	11
33	77.400	13	34	77.500	11
35	77.400	13	36	77.200	16
37	**77.700**	**13**	38	77.200	15
39	77.200	14	40	77.300	10
41	77.400	14	42	77.600	13
43	77.500	16	44	**77.700**	15
45	77.000	9	46	77.300	10
47	77.200	9	48	77.200	12
49	77.200	12	50	77.500	13

	Min.	First Qu.	Median	Mean	Third Qu.	Max.
Summary on accuracy %	77.00	77.20	77.30	77.31	77.40	**77.70**
Number of selected variables :	5.0	10.0	12.0	11.9	14.0	17.0

Bold values represent the best result

We note that an instance describe the profile of a given applicant.

2. The Australian Credit Approval is proposed by Quinlan [39]. It concerns credit card applications. The dataset consists of 690 instances of loan applicants. There are two classes: **class 1** (worthy, 307 instances) and **class 0** (unworthy, 384 instances). The number of variables is equal to 14. There are 6 numerical and 8 categorical variables. An example of an instance of "Australian" is given in Table 2.

4.3 Numerical results

Due to the non-deterministic nature of the proposed methods, 50 runs have been considered for each dataset and for each method. In the following, we give the results obtained with LS+SVM, SLS+SVM and VNS+SVM methods. We give the accuracy rate for each run for each method on each dataset.

We compute some summary statistics on accuracy and the number of selected variables. We give the minimum (Min), the mean, the median, the first quartile (first Qu.), the third quartile (third Qu.) and the maximum (Max). We give also the best solution found with the best accuracy for each dataset. The results are given in Tables 3, 4, 5, 6, 7, 8.

From Tables 3, 4, 5, 6, 7, 8 we observe that the obtained results can have the same number of selected variables but different accuracy on different runs. The local search-based feature selection methods do not lead to the same solution when applied to the same problem. This is due to the non-deterministic nature of these methods. In addition, some variables have a significant effect on the solution quality which leads to improvements in accuracy when such

Table 4 The results of 50 run of LS+SVM on Australian dataset

Run number	Accuracy %	Number of selected variables	Run number	Accuracy %	Number of selected variables
1	**86.376**	**9**	2	86.086	7
3	85.942	13	4	86.086	7
5	86.086	10	6	85.797	9
7	86.086	7	8	85.942	5
9	86.086	11	10	86.086	7
11	86.086	4	12	86.086	9
13	86.086	6	14	86.376	11
15	86.231	7	16	86.086	8
17	86.231	6	18	86.231	7
19	86.231	7	20	86.231	8
21	86.086	9	22	86.231	6
23	86.086	7	24	86.231	6
25	85.942	7	26	86.086	8
27	86.086	7	28	86.086	6
29	86.086	8	30	86.231	9
31	85.942	7	32	86.231	8
33	86.086	8	34	86.231	7
35	86.231	8	36	86.231	6
37	86.231	8	38	86.231	10
39	85.942	6	40	86.231	9
41	86.231	8	42	86.086	3
43	86.086	9	44	86.086	10
45	86.231	8	46	86.086	8
47	86.086	7	48	86.086	8
49	86.231	7	50	86.231	10

	Min.	First Qu.	Median	Mean	Third Qu.	Max.
Summary on accuracy	85.80	86.09	86.09	86.13	86.23	**86.38**
Number of selected variables	3.000	7.000	8.000	7.735	9.000	13.000

Bold values represent the best result

variables are selected. We can conclude that the generated solutions are not unique.

We can obtain solutions with the same number of variables but with different accuracy rate because the selected variables are not always the same. For example: Table 8 shows that the solutions with eight selected variables found in run 1, run 4 and run 39 are not the same in spite of the same number of selected variables. The accuracy rates are 86.811, 86.521 and 86.376, respectively.

For instance, the following two solutions have **8** selected variables. The solution: "0 **1 1 1 1 1 1** 0 0 **1** 0 0 **1** 0 " has an accuracy rate equals to **86.811%**. The selected variables are A2, A3, A4, A5, A6, A7, A10 and A13. But the solution: "**1 1 1** 0 0 **1** 0 **1 1** 0 **1** 0 0 **1**" has an accuracy rate equals to **86.521%**. The selected variables are: A1, A2, A3, A6, A8, A9, A11 and A14. This means that for the Australian dataset, the set of variables {A2, A3, A4, A5, A6, A7, A10 and A13} is more significant than the set of variables {A1, A2, A3, A6, A8, A9, A11 and A14}.

According to the numerical results, we can say that the three methods succeed in finding good results for the two considered datasets. However, we see a slight performance in favor of the variable neighborhood search (VNS). The latter is able to find better solution compared to LS and SLS. Hence, we can conclude that the VNS method with the four different neighbor structures is effective for feature selection and classification.

The superiority of VNS is due to the good combination of intensification and diversification which permits to explore the search space effectively and locate good solutions.

In addition to the numerical results given in the different Tables 3, 4, 5, 6, 7, 8, we draw the boxplots given in Figs.

Table 5 The results of 50 run of SLS+SVM on German dataset

Run number	Accuracy %	Number of selected variables	Run number	Accuracy %	Number of selected variables
1	77.300	15	2	77.400	9
3	77.700	12	4	77.700	12
5	77.500	17	6	77.500	9
7	77.600	13	8	77.600	10
9	77.400	11	10	77.000	16
11	77.200	10	12	77.200	12
13	77.400	15	14	77.300	9
15	77.300	10	16	**77.900**	**12**
17	77.400	11	18	77.300	15
19	77.700	16	20	77.200	9
21	77.200	13	22	77.200	10
23	77.400	12	24	77.300	15
25	77.300	9	26	77.500	13
27	77.200	10	28	77.800	13
29	77.400	10	30	77.100	17
31	77.200	15	32	77.300	11
33	77.200	11	34	77.800	10
35	77.500	14	36	77.600	15
37	77.500	8	38	77.600	10
39	77.200	10	40	77.400	10
41	77.200	13	42	77.600	13
43	77.500	11	44	77.100	10
45	77.200	11	46	77.500	15
47	77.300	16	48	77.600	10
49	77.400	10	50	77.400	12

	Min.	First Qu.	Median	Mean	Third Qu.	Max.
Summary on accuracy	77.0	77.2	77.4	77.4	77.5	**77.9**
Number of selected variables :	8.00	10.00	11.50	12.00	13.75	17.00

Bold values represent the best result

3 and 4 to better visualize the distribution of values of the classification accuracy.

From the box diagram depicted in Figs. 3 and 4,we visualized the distribution of classification accuracy on the 50 runs for each algorithm and for both Australian and German dataset. This diagram shows clearly that in general VNS is able to produce good solutions. The results are promising and demonstrate the benefit of the proposed technique in feature selection. To further demonstrate the effectiveness of the proposed technique in credit scoring, we give further comparisons in the next subsection.

4.4 A comparison with a pure SVM

In this section, we compare the three proposed methods LS+SVM, SLS+SVM and VNS+SVM with a pure SVM on both German and Australian datasets. The aim is to show the impact of the feature selection in the classification task.

Table 9 gives the results obtained with SVM, LS+SVM, SLS+SVM and VNS+SVM methods. We give the best accuracy rate and the number of best variables set (significant) returned by each method.

As we can see from Table 9 that the three methods are better than the pure SVM. The three proposed methods are able to find good results for the two considered datasets. SLS and LS are comparable and succeed in improving the accuracy rate of SVM.

Further, VNS+SVM method is more effective on both Australian and German datasets compared to both LS+SVM and SLS+SVM. We draw Fig. 5 (respectively Fig. 6) to compare a pure SVM with our approach in term of accuracy rate (respectively in term of the number of selected variables)

Table 6 The results of 50 run of SLS+SVM on Australian dataset

Run number	Accuracy %	Number of selected variables	Run number	Accuracy %	Number of selected variables
1	86.086	4	2	86.086	9
3	86.086	7	4	86.086	9
5	86.231	6	6	86.231	9
7	86.231	11	8	86.086	6
9	86.086	11	10	86.086	7
11	**86.376**	**10**	12	86.086	8
13	86.231	9	14	86.376	5
15	86.086	6	16	85.942	9
17	86.231	9	18	86.086	7
19	86.086	5	20	86.086	6
21	86.086	7	22	86.231	7
23	86.231	9	24	86.086	11
25	**86.376**	**9**	26	86.231	10
27	86.086	8	28	86.086	7
29	86.231	5	30	86.086	6
31	85.942	7	32	86.231	7
33	86.376	5	34	86.086	5
35	86.231	6	36	86.231	8
37	86.086	9	38	86.231	7
39	85.942	9	40	86.231	9
41	86.231	4	42	86.231	8
43	**86.376**	**9**	44	86.231	5
45	86.086	9	46	86.086	9
47	86.086	11	48	86.086	9
49	86.086	9	50	86.086	8

	Min.	First Qu.	Median	Mean	Third Qu.	Max.
Summary on accuracy	85.94	86.09	86.09	86.16	86.23	**86.38**
Number of selected variables	4.0	6.0	8.0	7.7	9.0	**11.0**

Bold values represent the best result

point of view. The performance of our approach compared to SVM is shown clearly in Figs. 5 and 6. We note that:

- LS returns 13 significant selected variables for the German dataset which are: A2, A4, A10, A12, A13, A15, A17, A18, A19, A20, A22, A23 and A24. The accuracy rate is equal to 77.70 %. The significant variables returned by LS for the Australian dataset are: A2, A3, A4, A6, A7, A9, A10, A13 and A14. The accuracy rate is equal to 86.38% and the number of selected variables s is 9.
- SLS returns 12 significant selected variables for the German dataset which are: A1, A3, A10, A13, A15, A16, A17, A18, A19, A20, A22, A23 and A24. The accuracy rate is equal to 77.90%. The significant variables found by SLS for the Australian dataset are: A2, A4, A6, A7, A9, A10, A11, A13 and A14. The accuracy rate is equal to 86.38% and the number of selected variables is 9.
- The **16** significant selected variables returned by VNS for the German dataset are: A2, A4, A5, A6, A9, A11, A12, A13, A14, A15, A16, A17, A18, A19, A22 and A24 where the accuracy rate is equals to **78%** and the. For the Australian dataset, VNS returns **8** significant variables which are: A2, A3, A4, A5, A6, A7, A10, A13. The best accuracy rate is equal to **86.81%**.

In this section, we compared the three proposed methods with feature selection to a pure SVM to measure the effectiveness of the additional feature selection method. As shown in Table 9, the proposed methods perform better than the pure SVM on both German and Australian datasets.

Further, we remark that fewer features are selected in the model to be used by SVM compared to the initial feature

Table 7 The results of 50 run of VNS+SVM on German dataset

Run number	Accuracy %	Number of selected variables	Run number	Accuracy %	Number of selected variables
1	77.400	14	2	77.700	14
3	77.300	11	4	77.600	12
5	77.500	13	6	77.300	13
7	77.400	12	8	77.300	12
9	**78.000**	**16**	10	77.600	10
11	77.300	15	12	77.800	9
13	77.200	11	14	77.500	10
15	77.400	11	16	77.400	13
17	77.300	13	18	77.200	15
19	77.500	14	20	77.500	9
21	77.400	13	22	77.300	14
23	77.800	14	24	77.800	14
25	77.200	12	26	77.500	8
27	77.300	12	28	77.300	7
29	77.800	13	30	77.200	11
31	77.200	13	32	77.400	12
33	77.300	9	34	77.600	10
35	77.900	11	36	77.500	11
37	77.400	14	38	77.400	15
39	77.400	9	40	77.400	15
41	77.300	13	42	77.200	14
43	77.500	14	44	77.800	13
45	77.600	10	46	77.700	15
47	77.500	13	48	77.600	12
49	77.300	13	50	77.400	17

	Min.	First Qu.	Median	Mean	Third Qu.	Max.
Summary on accuracy	77.20	77.30	77.40	77.46	77.60	**78.00**
Number of selected variables	7.00	11.00	13.00	12.36	14.00	17.00

Bold values represent the best result

number of the dataset. This implies that some features in the dataset are redundant and should be eliminated to enhance the classification accuracy.

4.5 Further comparison

To show the performance of the proposed approaches in credit scoring, we evaluated them against some well-known classifiers. Several classifiers can be found on the WEKA Data mining software package [43].

We compared our approaches with some popular classifiers which are: the rule-learning scheme (PART), ZeroR, JRip, BayesNet, NaiveBayes, adaBoost, attributeSelectedClassifier, Bagging, RandomForst, RandomTree and J48. These eleven classifiers from WEKA [43] were used in this study by means of their default parameters originally set in WEKA.

We add also a comparison with two well-known filtering methods. We choose the best-first search (CFS) and the ranking filter information gain methods (IGRF). We note that CFS is a correlation based feature selection that can be used to select a set of variables. However, CFS is unable to select all relevant variables when there are strong dependencies between variables. The IGRF ranking filter permits to select a set of variables from the original dataset using score or weights [19,37]. We combined these two feature selection methods (CFS and IGRF) with SVM to classify data.

Table 10 compares the three proposed methods (LS+SVM, SLS+SVM and VNS+SVM), the eleven classifiers from WEKA, CFS+SVM and IGRF+SVM on the two considered

Table 8 The results of 50 run of VNS+SVM on Australian dataset

Run number	Accuracy %	Number of selected variables	Run number	Accuracy %	Number of selected variables
1	**86.811**	**8**	2	86.376	7
3	86.376	9	4	86.521	8
5	86.521	10	6	86.376	5
7	86.521	6	8	86.521	7
9	86.521	10	10	86.667	7
11	86.376	9	12	86.521	8
13	86.521	9	14	86.232	8
15	86.376	5	16	86.521	6
17	86.667	4	18	86.521	4
19	86.521	8	20	86.521	8
21	86.667	7	22	86.521	9
23	86.521	5	24	86.667	8
25	86.376	6	26	**86.811**	**8**
27	86.232	9	28	86.521	12
29	86.376	4	30	86.521	7
31	86.376	7	32	86.521	7
33	86.376	9	34	86.667	8
35	86.376	6	36	86.521	6
37	**86.811**	9	38	86.667	9
39	86.376	8	40	86.521	8
41	86.521	8	42	86.376	7
43	86.376	7	44	86.521	6
45	86.521	9	46	86.376	6
47	86.376	6	48	86.521	11
49	86.376	9	50	86.376	3

	Min.	First Qu.	Median	Mean	Third Qu.	Max.
Summary on accuracy	86.23	86.38	86.52	86.50	86.52	**86.81**
Number of selected variables	3.0	6.0	8.0	7.4	9.0	12.0

Bold values represent the best result

datasets: Australian and German. The comparison is in term of the average classification accuracy rates.

As shown in Table 10, the three proposed approaches (LS+SVM, SLS+SVM and VNS+SVM) are comparable to the well-known classifiers. We can see a slight performance in favor of our VNS+SVM method. The proposed method (VNS+SVM) gives the highest average classification accuracy compared to PART, JRip, BayesNet, NaiveBayes, adaBoost, attributeSelectedClassifier, Bagging, RandomForst, RandomTree and J48 on both Australian and German datasets.

Further, we remark that OneR and VNS+SVM classifiers are comparable on Australian dataset. However, VNS+SVM is better than OneR on German dataset. OneR gives an average accuracy equal to **86.6%** on Australian dataset but it fails on German dataset where the average accuracy rate value given by OneR is equal to **60.8%**. The VNS+SVM method succeeds in finding good results for both Australian and German datasets. For Australian dataset, VNS+SVM gives an average accuracy value equal to **86.50%** when VNS is used as a feature selection method within SVM classifier. For German dataset, VNS+SVM gives the best average accuracy value equals to **77.46%** compared to the all considered classifiers.

When we compare the feature selection methods (CFS, IGRF and our three local search methods), we can see that our approaches provide good results compared to both CFS and IGRF ranking methods. For example, for German dataset, SVM with CFS gives an average accuracy value equals to **72.70%** when the CFS is used as a feature selection method while SVM with IGRF gives an average accuracy equal to **75.6%**. The results are much better when we use our pro-

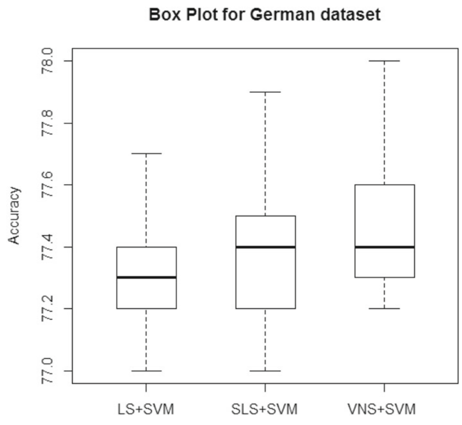

Fig. 3 LS+SVM versus SLS+SVM versus VNS+SVM on German dataset

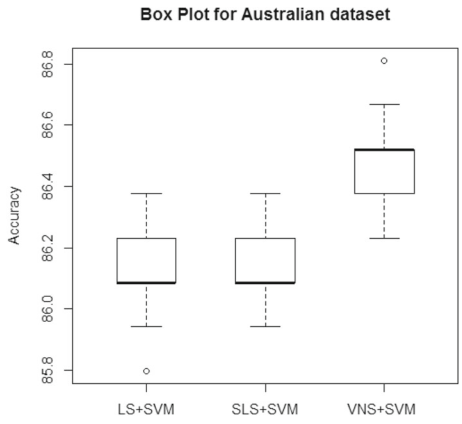

Fig. 4 LS+SVM versus SLS+SVM versus VNS+SVM on Australian dataset

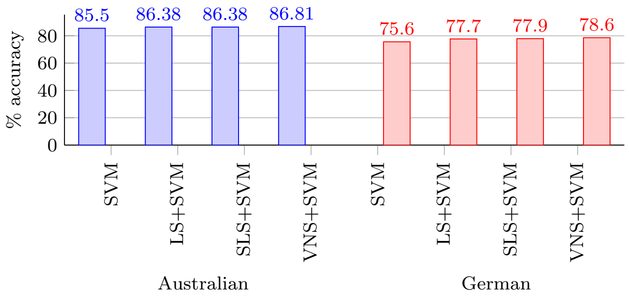

Fig. 5 Accuracy% comparison

posed approaches in particular when we consider VNS with SVM. As already said, the resulting method VNS+SVM gives the best average accuracy value which is equal to **77.46%** for the German dataset. This performance is also confirmed on Australian dataset with an average accuracy value equal to 86.50%.

In conclusion, we can say that the three proposed approaches (LS+SVM, SLS+SVM and VNS+SVM) are comparable. However, promising results are obtained when combining SVM with the VNS-based feature selection method. This improvement can be shown for the two considered datasets which proves the ability of VNS+SVM as a good classifier in credit scoring.

Table 9 SVM .vs. LS+SVM .vs. SLS+SVM .vs. VNS+SVM

Method		Australian	German
SVM	Accuracy %	85.50	75.6
	Number of attributes	14	24
LS+SVM	Max Accuracy %	86.38	77.70
	Number of significant selected attributes	9	13
	(best solution found)		
SLS+SVM	Max Accuracy %	86.38	77.90
	Number of significant selected attributes	9	12
	(best solution found)		
VNS+SVM	Max Accuracy %	**86.81**	**78.00**
	Number of selected significant attributes	**8**	**16**
	(best solution found)		

Bold values represent the best result

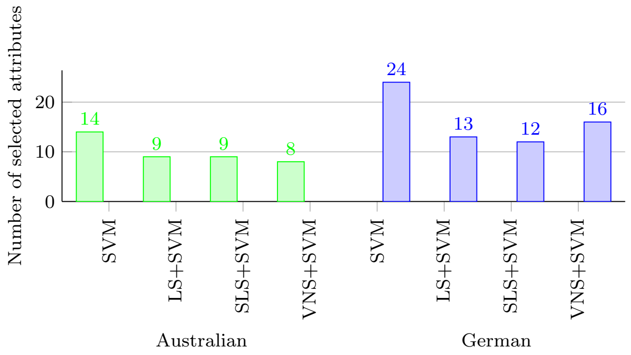

Fig. 6 Comparison in term of the number of selected attributes

5 Conclusion

This paper studied three local search feature-based methods combined with SVM model for credit scoring. The proposed model finds the best set of features (called also significant attributes or variables) by removing irrelevant variables and keeping only appropriate ones. The set of features is used then with SVM classifier to classify data. We studied three variants of local search-based feature selection: the local search hill climbing, the stochastic local search and the variable neighborhood search. The three variants combined with SVM are evaluated on two well-known German and Australian credit scoring datasets. The proposed methods have good accuracy

Table 10 A comparison according to the average classification accuracy rates

Method	Australian	German
PART	84.1	69.6
ZeroR	30.8	49
OneR	**86.6**	60.8
JRip	85.7	69.4
BayesNet	86.1	74.6
NaiveBayes	79.2	74.3
adaBoost	84.4	66.1
attributeSelctedClassifier	83.7	69.3
Bagging	85.5	73.2
RandomForest	86.2	75.1
RandomTree	76.9	66.7
J48	86.1	68.7
CFS+SVM	73.19	72,70
IGRF +SVM	85.5	75.6
LS+SVM	86.13	77.31
SLS+SVM	86.16	77.40
VNS+SVM	**86.50**	**77.46**

Bold values represent the best result

performance with fewer features. The proposed VNS+SVM method performs better on both German and Australian datasets compared to SVM, LS+SVM, SLS+SVM and other well-known classifiers. We plan to improve our work by optimizing the SVM parameters. Further, it would be nice to study the impact of feature selection-based method on other machine-learning techniques.

Acknowledgements The authors would like to thank the developers of the Library for support vector machines (LIBSVM) for the provision of the open source code. The authors would like to thank also the developers of Waikato Environment for Knowledge Analysis (WEKA).

References

1. Abdou, H.A.: Genetic programming for credit scoring: the case of Egyptian public sector banks. Expert Syst. Appl. **36**, 11402–11417 (2009)
2. Abelln, J., Mantas, C.J.: Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring. Expert Syst. Appl. **41**, 3825–3830 (2014)
3. Bellotti, T., Crook, J.: Support vector machines for credit scoring and discovery of significant features. Expert Syst. Appl. **2009**(36), 3302–3308 (2009)
4. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees 1984. Wadsworth, Belmont (1984)
5. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. **2**(1998), 121–167 (1998)
6. Boughaci, D., Alkhawaldeh, A.A.K.: A cooperative classification system for credit scoring. In: Proceedings of AUEIRC 2017. Springer (2017) **(to appear)**
7. Boughaci, D., Benhamou, B., Drias, H.: A memetic algorithm for the optimal winner determination problem. Soft Comput. **13**, 905–917 (2009)
8. Boughaci, D.: Meta-heuristic approaches for the winner determination problem in combinatorial auction. In: Yang XS. (ed.) Artificial Intelligence, Evolutionary Computing and Metaheuristics, Studies in Computational Intelligence, vol. 427, pp. 775–791. Springer, Berlin, Heidelberg (2013)
9. Boughaci, D., Benhamou, B., Drias, H.: Local Search Methods for the optimal winner determination problem. J. Math. Model. Algorithms (Springer) **9**(2), 165–180 (2010) . http://www.springerlink.com/content/hv637861870mx8j4/
10. Caruana, R., Freitag, D.: Greedy attribute selection. In: Proceedings of the eleventh international conference on machine learning. (ICML 1994, New Brunswick, New Jersey). Morgan Kauffmann, pp. 28–36 (1994)
11. Campbell, C., Ying, Y.: Learning with Support Vector Machines. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool, San Rafael (2011)
12. Chakraborty, B.: Genetic algorithm with fuzzy fitness function for feature selection. In: Proceedings of the IEEE international symposium on industrial electronics vol. 1, pp. 315–319 (2002)
13. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) http://www.csie.ntu.edu.tw/cjlin/libsvm/oldfiles/index-1.0.html
14. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) http://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/**Data sets**
15. Desay, V., Crook, J.N., Overstreet, G.A.: A comparison of neural networks and linear scoring models in the credit union environment. Eur. J. Oper. Res. **95**(1996), 24–37 (1996)
16. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian Network Classifiers. Mach. Learn. **29**(1997), 131–163 (1997)
17. Goldberg, D.E., Korb, B., Deb, K.: Messy genetic algorithms: motivation, analysis, and first results. Complex Syst. **5**(3), 493–530 (1989)
18. Glover, F.: Tabu search—part 1. ORSA J. Comput. **1**(2), 190–206 (1989). https://doi.org/10.1287/ijoc.1.3.190
19. Hall, M.a: Correlation-based feature selection for machine learning. Methodology **21i195i20**, 15 (1999). April
20. Hand, D.J., Henley, W.E.: Statistical classification methods in consumer credit scoring. J. R. Stat. Soc. Ser. A (Stat. Soc.) **160**, 523–541 (1997)
21. Henley, W.E., Hand, D.J.: A k-nearest neighbour classifier for assessing consumer credit risk. Statistician **45**(1996), 77–95 (1996)
22. Milne, A., Rounds, M., Goddard, P.: Optimal feature selection in credit scoring and classification using a quantum annealer (2017)https://1qbit.com/whitepaper/optimal-feature-selection-in-credit-scoring-classification-using-quantum-annealer/
23. Hansen, P., Mladenovic, N.: Variable neighbourhood search: principles and applications. Eur. J. Oper. Res. **130**, 449–467 (2001)
24. Hertz, J.A., Krogh, A., Palmer, R.G.: Introduction to the Theory of Neural Computation. Addison-Wesley Publishing Company Inc, Redwood City (1991)
25. Hoos, H., Stutzle, T.: Stochastic Local Search: Foundations and Applications. Morgan Kaufmann, San Francisco (2005)
26. John, G.H., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the eleventh conference on

uncertainty in artificial intelligence. San Mateo: Morgan Kaufman, pp. 338-345 (1995)

27. Ju, Y., Sohn, S.Y.: Technology credit scoring based on a quantification method. Sustainability **9**(6), 1057 (2017). (Multidisciplinary Digital Publishing Institute)
28. Kirkpatrick Jr., S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science **220**(4598), 671680 (1983). Bibcode:1983Sci...220..671K. https://doi.org/10.1126/science.220.4598.671. **(JSTOR 1690046)**
29. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial intelligence, Special issue on relevance 273–324. (1996)
30. Lanzi, P.L.: Fast feature selection with genetic algorithms: a filter approach. In: IEEE international conference on evolutionary computation, vol. 25, pp 537–540 (1997)
31. Li, J., Wei, L., Li, G., Xu, W.: An evolution strategy-based multiple kernels multi-criteria programming approach: the case of credit decision making. Decis. Support Syst. **51**, 292–298 (2011)
32. Mladenovic, N., Hansen, P.: Variable neighbourhood decomposition search. Comput. Oper. Res. **24**, 1097–1110 (1997)
33. Miller, M.: Research confirms value of credit scoring. Natl. Underwrit. **107**(42), 30 (2003)
34. Moscato, P.: On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. In: Caltech concurrent computation program, C3P Report 826 (1989)
35. Nekkaa, M., Boughaci, D.: Memetic Algorithm with Support Vector Machine for Feature Selection and Classification, Memetic Computing (2015), vol. 7, 5973, Springer (2015). https://doi.org/10.1007/s12293-015-0153-2, http://link.springer.com/journal/12293
36. Nekkaa, M., Boughaci, D.: Hybrid Harmony Search Combined with Stochastic Local Search for Feature Selection. Neural Process Lett (2015). Springer (2015). https://doi.org/10.1007/s11063-015-9450-5, http://link.springer.com/journal/11063
37. Phyu, T.N.: Survey of Classification Techniques in Data Mining. In: Proceedings of the international multi conference of engineers and computer scientists, vol I IMECS 2009, March 18–20, 2009, Hong Kong (2009)
38. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1992)
39. Quinlan, J.R.: Simplifying decision trees. Int. J. Man Mach. Stud. **27**, 221–234 (1987)
40. Sousaa, M.R., Gamaa, J., Brando, E.: A new dynamic modeling framework for credit risk assessment. Expert Syst. Appl. **45**, 341–351 (2016)
41. Tan, K.C., Teoh, E.J., Goh, K.C., Yua, Qb: A hybrid evolutionary algorithm for attribute selection in data mining. Expert Syst. Appl. **36**, 8616–8630 (2009)
42. Vapnik, V.: Statistical Learning Theory. John Wiley and Sons, New York (1998)
43. Waikato Environment for Knowledge Analysis (WEKA), Version 3.8. The University of Waikato, Hamilton, New Zealand. http://www.cs.waikato.ac.nz/ml/weka/ (2018). Accessed February 2018
44. Wiginton, J.C.: A note on the comparison of logistic and discriminant models of consumer credit behavior. J. Financ. Quant. Anal. **15**, 757–770 (1980)
45. Yang, X.-S.: Harmony search as a metaheuristic algorithm. In: Editor, Z., Geem, W. (eds.) Music-Inspired Harmony Search Algorithm: Theory and Applications, Studies in Computational Intelligence. Springer, Berlin (2009)