Computer Science • 23(1) 2022 https://doi.org/10.7494/csci.2022.23.1.4120

Waad Bouaguel

EFFICIENT MULTI-CLASSIFIER WRAPPER FEATURE-SELECTION MODEL. APPLICATION FOR DIMENSION REDUCTION IN CREDIT SCORING

Abstract *The task of identifying the most relevant features for a credit-scoring application is a challenging task. Reducing the number of redundant and unwanted features is an inevitable task for improving the performance of a credit-scoring model. The wrapper approach is usually used in credit-scoring applications to identify the most relevant features. However, this approach suffers from the issue of subset generation and the use of a single classifier as an evaluation function. The problem here is that each classifier may give different results that can be interpreted differently. Hence, we propose an ensemble wrapper feature-selection model in this study that is based on a multi-classifier combination. In the first stage, we address the problem of subset generation by minimizing the search space through a customized heuristic. Then, a multi-classifier wrapper evaluation is applied using two-classifier-arrangement approaches in order to select a set of mutually approved sets of relevant features. The proposed method was evaluated on four credit datasets and has shown good performance as compared to individual classifier results.*

Keywords multi-classifier, heuristic, dimensionality reduction, credit scoring

Citation Computer Science 23(1) 2022: 133–155

1. Introduction

A key issue that faces financial institutions when building credit-scoring (CS) models is how to define the most appropriate set of features. In fact, the progress of data-storage technologies has provided opportunities to have a large set of features and expand financial analyses. However, high-dimensional data is a challenge to data-mining methods. In general, scoring models use the credit history of previous customers to compute a new applicant's defaulting risk [11]. The collected set of booked loans may come from different sources and may be collected for a general task [16, 18]. Feature selection is a term that is commonly used in machine learning to denote methods for reducing a dataset to a convenient size for processing and investigation. This process involves not only a predefined cutoff on the number of features that can be considered when building a credit-scoring model but also the choices of appropriate features based on their relevance to the study [4].

In order to reduce the effect of unwanted features in a dataset, feature selection is a crucial process that is generally performed before the classification step. Unneeded features include irrelevant and redundant features [15]. The first (irrelevant features) are those that can never contribute to improving the predictive accuracy of a credit model since they have not any correlation with the response variable. Removing such features reduces the dimension of the search space and speeds up the learning algorithm. The second (redundant features) are those that may replace others in a feature subset since they basically bring similar information (i.e., date of birth and age features). Typically, feature redundancy is defined in terms of features correlation: two features are considered to be redundant if they are highly correlated.

The reduction task of features can lead to parsimonious credit models and helps simplify the practice of different visualization techniques, consequently yielding better accuracy and easier interpretations. Two main classes of feature selection have identified in the literature [14]: filter and wrapper feature-selection methods.

Filter methods are generally used as a pre-processing step. A filter method chooses the best features by studying their intrinsic properties (i.e., the relevance or correlation of the features to the target concept) that are measured via uni-variate statistics without considering the properties of the classifier. Hence, the selection of features is independent of any machine-learning algorithms. In contrast to filter methods, wrappers actually take a classifier's proprieties into consideration. The main idea of wrapper feature selection is to remove unwanted features from the data by using the predictive accuracy of a particular classifier as an evaluation function [9].

To be more precise, the main differences between the two classes of feature-selection methods are as follows: first, filter methods measure the relevance of features by their correlation with the target feature, while wrapper methods measure the usefulness of a subset of a feature by training a model on it. Second, the overfitting problem is more frequent when using wrapper methods as compared to using a subset of features from the filter methods. However, it is important to point out

that filter methods might fail to find the best subset of features on many occasions, while wrapper methods are more likely to provide the best subset of features.

It has been shown that wrappers generally outperform filters [9] in terms of accuracy since they are tuned to the specific interactions between a classifier and a dataset. Hence, we propose to apply wrapper feature selection on credit-scoring data in order to obtain a simpler credit-scoring model. However, wrapper methods have practical and theoretical limitations [3]. They typically lack generality since the resulting subset of features is tied to the bias of the classifier that is used in the evaluation function. The optimal feature subset will be specific to the classifier under consideration. Also, finding the optimal feature subset will come with a high computational cost. This cost depends on the number of times the classifier is trained on the evaluation process, the number of subsets to be investigated, and the sizes of these feature subsets. The number of subsets and their sizes depend on the used search strategy. In the case of a complete search, the number of subsets increases along with the time complexity. However, using a heuristic-only reduced number of subsets will be investigated (which may reduce the quality of the selected features).

In this paper, we try to solve these two shortcomings in wrapper feature selection in credit-scoring application: the bias of the classifier, and the subsets' generation process. In order to minimize the number of evaluations that are performed by the classifier while maintaining good accuracy, we design a search algorithm that reduces the number of possible candidates. The proposed algorithm uses a mixture of complete search and heuristic search techniques in order to reduce the search space. Then, an empirical study is conducted that combines multiple classifiers in the process of wrapper evaluation in order to select an optimal and unbiased set of features. We show how the number and type of the classifiers within the combination framework may influence the final results.

This paper is organized as follows. Section 2 briefly reviews the major issues of wrapper feature selection. Section 3 describes the proposed approach for solving the discussed issues. Experimental investigations are given in Section 4, while Section 4.2 gives empirical results and discussions regarding four datasets. Finally, Section 5 provides conclusions.

2. Issues with wrapper approach of feature selection

The main idea of the manuscript is to perform a feature selection on financial data in order to create simple credit scoring models. Wrapper feature selection was chosen to reduce the feature space; this choice was due to the fact that wrappers generally outperform filters in terms of accuracy. However, wrapper methods suffer from two major shortcomings. The first one is a lack of generality. Typically, a single classifier is used to evaluate the features in a wrapper framework; this makes the final result dependent on the classifier (meaning that using a new classifier with another assumption will change the final result). Based on the important limitation of using

a single classifier, we consider using more than one classifier within a wrapper feature-selection framework to improve the general accuracy. The adopted methodology to overcame this first shortcoming is to perform a complete experimental study in which we investigate the appropriate number of classifiers to use in the study as well as their nature. The second shortcoming of the wrapper feature-selection method is related to the search strategy and the way the subsets are generated for further investigations. Two search strategies are used in the literature. The first one is an exhaustive search in which all possible feature candidates are evaluated. The second strategy is heuristic, which helps in getting a valid subset within a reasonable amount of time. An exhaustive search always guarantees the best solution; however, it is unrealistic if the number of features is important. Heuristics give a good approximation, but it is still impossible to look for an optimal subset. The adopted methodology to overcome this second shortcoming is to perform a first reduction of the search space by using the prior knowledge of bank experts and heuristics. This will reduce the search space for the exhaustive search.

2.1. Issue I: using single classifier or combination of classifiers

Using a single classifier in the wrapper process may favor one candidate subset over others [6, 15]. In fact, the difference in the biases and assumptions of each classifier may affect the final result in terms of accuracy and execution time [3]. When changing the classifier, the set of features to be selected may change; this leads to a lack of generality in the produced model. The level of the computational complexity of the classifier is also a fundamental factor to be investigated. Classifiers that have a large computational cost will take much longer to choose the best subset of features than a low-computational-cost classifier. For example, when a support vector machine (SVM) is used as an evaluation function in the process of finding the best feature subset, it may take more time to identify the most relevant features than when using logistic regression (LR) or the k-nearest neighbors algorithm (KNN).

The wrapper approach can also be based on a combination of the results of several classifiers. The number of classifiers that are used in the combination framework affects the evaluation process. If a small number of classifiers is considered, then it is likely that the level of agreement (degree of matching) among them will be high. A high agreement among classifiers may subsequently result in more-relevant features being selected with different levels of accuracy. However, if a large number of classifiers are used, we may end up getting fewer relevant features. Indeed, the level of agreement between the classifiers will probably be low since more classifiers are required to agree on the relevance of a feature.

Based on the important limitation of using a single classifier, we consider using more than one classifier within a wrapper feature-selection framework in order to improve the general accuracy. Hence, we look for a mutually approved set of significant features. Such a set will possibly increase the classification accuracy and reduce the biases of the individual classifiers.

2.2. Issue II: subset generation and search strategy

A theoretical ideal feature-selection approach would be based on an exhaustive search of a full set of features in order to find the optimal subset. However, an exhaustive search becomes rapidly impractical as the number of features (denoted by d) increases (even for a moderate number of features) [1, 10, 20] . If we look at different ways in which feature subsets are generated among many variations, three basic schemes are available in the literature; forward selection, backward elimination, and random scheme [8].

Forward selection and backward elimination are considered to be heuristics. Generally, sequential generation can help in getting a valid subset within a reasonable amount of time, but it is still impossible to look for an optimal subset. This is due to the fact that the generation scheme uses a heuristic to obtain an optimal subset by sequentially selecting the best one (as in the forward case) or removing the worst one (as in the backward case). Using such a generator will certainly speed up the selection process; however, it cannot turn back if the search falls in a local optimum. In fact that the generator has no way of getting out of the local optimum because what has been removed in each step cannot be added in the next steps. This fact is an important shortcoming of sequential schemes.

To overcome this problem, one can use a random-generation scheme to add randomness to the fixed rule of sequential generation and avoid getting stuck at some local optima. Although the random-generation scheme could improve the sequential results, it does not guarantee finding an optimal subset. This can be further elaborated in terms of the search strategies [19].

Hence, we propose reducing the number of features by forward selection and backward elimination in order to minimize the search space so that the exhaustive search method can handle the generation process within a realistic amount of time. In this way, the selected feature set is much better in terms of accuracy than those from forward selection and backward elimination; the feature subsets are also obtained much faster than with the exhaustive method.

3. New approach for wrapper feature selection

In this section, we design a combination approach for wrapper feature selection. We consider building a three-stage wrapper feature-selection model.

- At first, a primary dimensionality reduction step based on a similarity study with the prior knowledge is conducted on the original feature space. This step is used to reduce the search space.
- Second, the subset-generation step is performed by using a mixture of heuristic and exhaustive search methods.
- The final step is an evaluation of the effect of the wrapper feature selection by using multiple classifiers from the same family as well as the effect of combining multiple classifiers from different families in the wrapper process.

3.1. Primary dimensionality reduction step: similarity study

The first step of the designed approach aims specifically at selecting fewer redundant features without a loss of quality. The redundancy is measured by a similarity measure between a pre-selected set of features and the remaining features in the dataset. The objective is to enhance the existing set of pre-selected features by adding additional features as a complement. This enhancement is based on expert knowledge.

Experts from banks typically possess valuable knowledge about which important features to be included in an analysis based on their experience. Thus, the possible improvement of an exhaustive search is to use this prior knowledge and eliminate any redundant features before generating the candidate subsets. Since our goal is to take advantage of any additional information about this feature, we build a complementary set of features to be added to those that were pre-selected by bank experts.

First, we split the features into two sets. The first one groups a set of features that were assumed to be more relevant according to some prior knowledge (selected by experts), while the second set contains those that remain. Once the two sets are obtained, we conduct a similarity study on the exiting correlations between the set of variables by using the mutual information (MI) metric. This metric is used to measure the relevance of features by taking the amount of information that is shared by two features into account. Formally, the MI of two continuous random variables (X^j and $X^{j\prime}$) is defined as follows:

$$MI(X^j, X^{j\prime}) = \int\int p(x^j, x^{j\prime}) log \frac{p(x^j, x^{j\prime})}{p(x^j)p(x^{j\prime})} dx^j dx^{j\prime}, \tag{1}$$

where $p(x^j, x^{j\prime})$ is the joint probability density function, and $p(x^j)$ and $p(x^{j\prime})$ are the marginal probability density functions.

In the case of discrete random variables, the double integral becomes a summation, where $p(x^j, x^{j\prime})$ is the joint probability mass function, and $p(x^j)$ and $p(x^{j\prime})$ are the marginal probability mass functions. Large values of MI indicate a high correlation between two features, and zero indicates that two features are uncorrelated.

Once the pair-based similarity matrix is obtained, we investigate the levels of the similarity of each pair of features from the two different sets. If the similarity is above 80%, the evaluated feature is eliminated; otherwise, this variable is maintained for further examination.

The first part of Figure 1 shows a simplified flow chart of the dimensionality reduction before the second step.

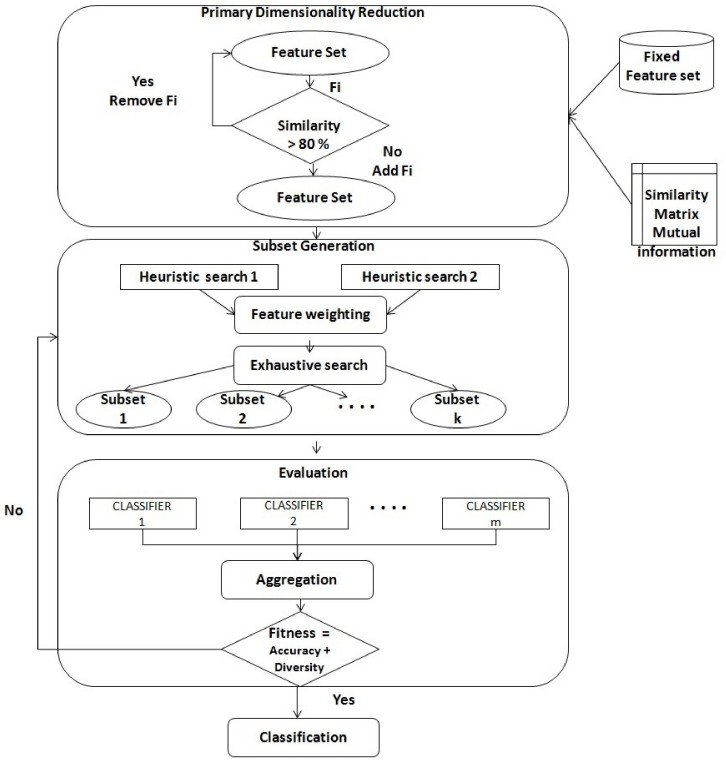

Figure 1. Wrapper approach that combines multiple classifiers for feature selection; F_i represents feature to be evaluated in each step

3.2. Subset-generation step: combination of heuristics for reducing set of features

Although the search space is reduced in the previous step, applying an exhaustive search is still computationally impossible. According to [1], an exhaustive search can be done only when the number of features is less than ten. Using more than ten features would be costly in terms of computational time. In fact, an exhaustive search is an enumeration search method that considers all possible feature combinations. To reduce the search space to a manageable size, specific heuristics can be used. Our objective in this second step is to reduce the search space to fewer than ten features to make an exhaustive search using heuristics possible.

In theory, each search strategy has its particular effects on the selected feature subset as well as on the performance of the induction algorithm. Therefore, we propose the use of ensemble methods to combine the results of several heuristics. We use both sequential forward feature selection and backward feature elimination as part of a combined feature selection. Figure 2 illustrates the proposed combination process for an example of ten features. In the first step, the forward-selection and backward-

elimination methods are simultaneously applied to the reduced feature set, resulting in two different intermediate feature lists. Each list includes a set of complementary variables. In the second step, the two lists are merged into one single list of the most relevant features, while the non-selected features are eliminated. Since some of the selected features may appear in one of the intermediate feature lists and not in the other, these features must be re-weighted in order to take their relevance degree into consideration. A feature that is selected by both forward and backward selection is considered to be more relevant than another feature that is selected only once.

Consequently, the resulting features are then re-weighted according to their numbers of appearances in the intermediate lists. Actually, the weight is equal to 1 if it appears in the two intermediate feature subsets; otherwise, it is 0.5. In the third step, a complete search is used on the weighted features.

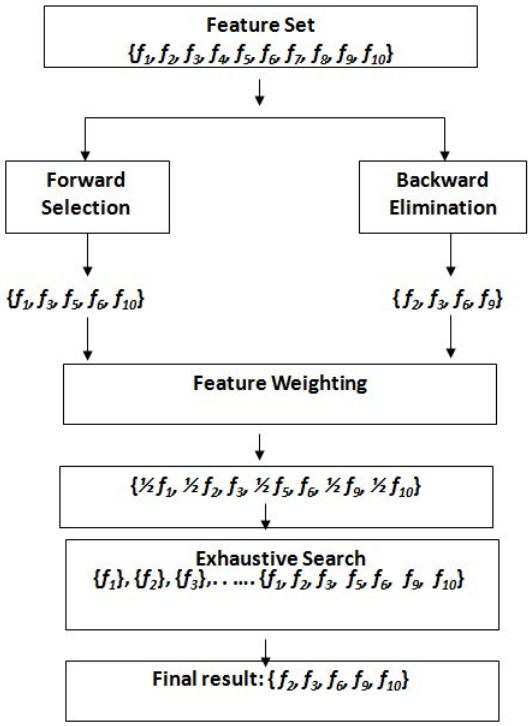

Figure 2. Combined wrapper's feature selection-search strategies

3.3. Evaluation step: effects of using multiple classifiers

Many classification methods have been proposed to deal with the credit-worthiness problem on the basis of information from past applicants. The most common statistical methods for evaluating applicants' solvency are logistic regression (LR) and discriminant analysis (DA) [12]. Unfortunately, this category of methods needs some

fundamental assumptions on the data [17], such as the normality distribution and absence of multi-collinearity. In addition to statistical methods, different machine-learning and artificial-intelligence methods have been used; e.g., Decision Tree (DT), Artificial Neural Networks (ANN), Support Vector Machines (SVM), and many others. Although the majority of these methods are simple and do not need assumptions on the data, those methods need a good mechanism to search for optimal model parameters and feature subsets.

Each of these individual methods produces a single discrimination rule and has some qualities and restrictions that may influence the feature-evaluation process. No one can generalize the superiority of one classifier over another for all applications. Rather than trying to optimize the accuracy of one classifier, it is better to integrate multiple classifiers. This approach has been recognized to be successful, as it achieves better performance and has a higher precision of predictability in the learning process [2,5,16]. Here, the same ensemble concept is adopted. Figure 1 shows how the results of a set of classifiers are merged to form a new evaluation function.

3.3.1. Classifier selection and arrangement approaches

The chosen algorithms in this study are representative of the most popular family of machine-learning classifier models that were selected to form committees of experts in order to test various classifier-combination schemes. We focus only on the general aspect of each family. Among the most popular classifier models, four were selected: DT, SVM, KNN, and ANN.

For the combination of classifiers, two different classifier arrangement approaches are used within the wrapper-evaluation process; namely, the same-type approach, and the mixed-type approach. The same-type approach combines only classifiers from the same family and uses them within the wrapper framework to select the relevant features, while the mixed-type approach combines classifiers from different families.

Table 1
Summary of used classifiers within each family

DT	ANN	KNN	SVM
J48	Multi-layer Perceptron (MP)	K=1 (1NN)	Polynomial (SVMP)
RandomForest (RF)	Voted Perceptron (VP)	K=5 (5NN)	Radial (SVMR)

For the same-type combination approach, we combined two classifiers for each of the four families of algorithms; these were as follows: two classifiers from the DT family, two from the ANN family, one from the KNN family (with two different numbers of neighbors – K=1, and K=5), and one from the SVM family (with two different kernels: polynomial and radial kernel functions). All of the considered classifiers are summarized in Table 1.

Concerning the mixed-type arrangement approach, we investigate how classifiers from different families work together and how their interaction affects the selection

of the features. The classifiers are combined so that each is used with every other one from a different nature. This leads to the construction of a total of 76 mixed-type classifier combinations (described in Tables 2–3), which include 24 two-classifier mixed-type combinations and 52 three-mixed-type combinations.

Table 2
Summary of possible combinations of pairs of selected classifiers

Possible combinations
(J48+ SVMP), (J48+ SVMR), (J48+ MP), (J48+ VP), (J48 +1NN), (J48+5NN), (RF+ SVMP), (RF+ SVMR), (RF+ MP), (RF+ VP), (RF +1NN), (RF+5NN),(SVMP + MP), (SVMP + VP), (SVMP +1NN), (SVMP +5NN), (SVMR + MP), (SVMR + VP), (SVMR +1NN), (SVMR +5NN), (MP +1NN), (MP +5NN), (VP +1NN), (VP +5NN).

Table 3
Summary of all possible combinations of three classifiers

Possible combinations
(J48 +RF + SVMP), (J48 +RF+ SVMR), (J48 +RF + MP), (J48 +RF +VP), (J48 +RF +1NN), (J48 +RF +5NN), (J48+ SVMP+ SVMR), (J48+ MP + VP), (J48 +1NN+5NN), (J48+ SVMP + MP),(J48+ SVMP + VP), (j48+SVMP +1NN), (J48+ SVMP +5NN), (J48+ SVMR + MP), (J48+ SVMR + VP), (J48+ SVMR +1NN), (J48+ SVMR +5NN), (J48+ MP +1NN), (J48+ MP +5NN), (J48+ VP +1NN), (J48+ VP +5NN), (RF + SVMP+ SVMR), (RF + MP + VP), (RF +1NN+5NN), (RF + SVMP + MP), (RF + SVMP + VP), (RF+SVMP +1NN), (RF + SVMP +5NN), (RF + SVMR + MP), (RF + SVMR + VP), (RF+SVMR +1NN), (RF + SVMR +5NN), (RF + MP +1NN), (RF + MP +5NN), (RF + VP +1NN), (RF + VP +5NN), (SVMP+SVMR + MP), (SVMP+SVMR +VP), (SVMP+SVMR +1NN), (SVMP+SVMR +5NN), (SVMP + MP + VP), (SVMP +1NN+5NN), (SVMP + MP +1NN), (SVMP + MP +5NN), (SVMP + VP +1NN), (SVMP + VP +5NN), (SVMR + MP + VP), (SVMR +1NN+5NN), (SVMR + MP +1NN), (SVMR + MP +5NN), (SVMR + VP +1NN), (SVMR + VP +5NN).

3.3.2. Aggregation rules

Traditionally, an experimental study is performed to compare the performance of several classifiers and select the best ones in order to build a multi-classifier system. However, many alternative approaches that are based on combining multiple classifiers have emerged over the recent years [6, 7], which can be basically classified into two classifier-combination scenarios. In the first scenario, all of the classifiers use the same representation of the input example. In this case, each classifier (for a given input example) produces an estimate of the same posterior class probability. In the second scenario, each classifier uses its representation of the input example. For multiple classifiers that use distinct representations, many existing schemes can be considered where all of the representations are used jointly in order to make a decision. Examples of these combination rules are the the average rule, minimum rule, maximum rule, and majority voting rule.

The arithmetic mean (also known as the average) gives an aggregated value that is smaller than the greatest argument and greater than the smallest one. The resulting aggregation is "a middle value." The minimum and maximum are also basic aggregation operators; the minimum gives the smallest value of a set, while the maximum gives the greatest one. Majority voting is also a common classifier combination method; it is used particularly in classifier ensembles when the class labels of the classifiers are crisp. Majority voting does not require any parameter to be trained nor any additional information for the later interpretation of results.

4. Experimental Investigations

4.1. Dataset description and evaluation metrics

To evaluate the performance of the proposed multi-classifier wrapper feature-selection model in credit scoring, we considered four different real-world credit datasets: the Australian, German, and HMEQ credit datasets (which are extracted from the UCI repository of machine-learning databases), and a credit dataset from a Tunisian bank. Table 4 displays the main characteristics of the used datasets.

Table 4
Summary of datasets used in experiments

Names	Australian	German	HMEQ	Tunisian
Total instances	690	1000	5960	2970
Nominal features	6	13	2	11
Numeric features	8	7	10	11
Total features	14	20	12	22
Number of classes	2	2	2	2

- The Australian credit dataset presents an interesting mixture of attributes: continuous, nominal with small numbers of values, and nominal with larger numbers of values (with few missing values). This dataset composed of 690 instances, where 307 were creditworthy and 383 were not. All of the attribute names and values were changed to meaningless symbols for confidentiality.
- The German credit dataset is often used by credit specialists for classification purposes. This dataset covered a sample of 1000 credit consumers, where 700 instances were creditworthy and 300 were not. For each applicant, 21 numeric input variables were available: 7 numeric, 13 categorical, and 1 target attribute.
- The HMEQ credit dataset was composed of 5960 instances that described recent home equity loans, where 4771 instances were creditworthy and 1189 were not. The target was a binary variable that indicated whether or not an applicant eventually defaulted. For each applicant, 12 input variables were recorded (where 10 were continuous features, 1 was binary, and 1 was nominal).
- The Tunisian credit dataset covered a sample of 2970 instances of credit consumers, where 2523 instances were creditworthy and 446 were not. Each credit applicant was described by a binary target variable and a set of 22 input variables, where 11 features were numerical and 11 were categorical.

For each dataset, any missing values were replaced with the mean or mode (the value that appears the most often) of the features depending on the type of variable (numerical or categorical). In addition, we performed a discretization process for all of the continuous variables in each dataset in order to simplify the interpretations of the results.

The performance of the designed model (including the three stages) was evaluated once for all of the stages. Given the difficulty of the first stage of the model (which depended on the expert recommendations and on the specificity of the credit-scoring application), we considered that all of the presented variables in the datasets resulted from the first stage of our model. Concerning the evaluation measures, we used standard information-retrieval performance measures: precision, recall, F-measure, and ROC area.

4.2. Results and discussion

The precision, recall, F-measure, and ROC area of the feature subsets that were selected from different combinations are given in Tables 5, 6, 7, and 8 for the four datasets using a ten-fold cross validation. The best results are shown in bold. Two approaches for wrapper evaluation are presented; namely, the same-type and mixed-type approaches. The results for the first approach are investigated in Section (4.2.1), and those for the second approach are presented in Section (4.2.2).

4.2.1. Results and discussion for same-type approach

When looking at the results that were produced by the DT family in Tables 5, 6, 7, and 8, we notice that the J48 classifier achieves the best individual results in most

cases for the German, HMEQ, and Tunisian datasets; however, the individual results that were produced by SVM were slightly better in the Australian dataset.

Table 5

Performance comparison of new wrapper method and other feature-selection methods for Australian dataset

	Precision	Recall	F-Measure	ROC Area
	Decision Tree			
J48	0.867	0.855	0.855	0.862
RF	0.863	0.851	0.851	0.858
Average	0.782	**0.925**	0.848	0.863
Product	0.864	0.852	0.853	0.859
Maximum	**0.930**	0.794	**0.856**	0.859
Minimum	0.866	0.855	0.855	0.862
Majority Vote	0.782	0.922	0.846	**0.865**
	Support Vector Machine			
SVMP	0.921	0.794	0.853	0.855
SVMR	**0.930**	0.799	**0.860**	0.862
Average	0.787	**0.925**	0.850	**0.864**
Product	0.866	0.855	0.855	0.861
Maximum	0.859	0.848	0.848	0.856
Minimum	0.927	0.794	0.855	0.858
Majority Vote	0.781	0.915	0.848	0.857
	Artificial Neural Network			
MP	0.860	0.849	0.850	0.856
VP	0.859	0.848	0.848	0.855
Average	0.862	0.851	0.851	0.857
Product	0.783	**0.919**	**0.861**	**0.860**
Maximum	0.862	0.851	0.851	0.857
Minimum	0.862	0.851	0.851	0.857
Majority Vote	**0.864**	0.853	0.854	0.858
	K-Nearest Neighbor			
1NN	**0.865**	0.852	**0.852**	0.860
5NN	0.859	0.848	0.848	0.855
Average	0.812	**0.890**	0.849	0.877
Product	0.811	0.866	0.838	**0.883**
Maximum	0.820	0.880	0.849	0.875
Minimum	0.824	0.823	0.822	0.876
Majority Vote	0.853	0.851	0.851	0.882

Table 6
Performance comparison of new wrapper method and other feature-selection methods for German dataset

	Precision	Recall	F-Measure	ROC Area
	Decision Tree			
J48	0.735	0.750	0.723	**0.635**
RF	0.686	0.716	0.665	0.570
Average	0.740	0.930	0.824	0.583
Product	0.732	0.933	0.820	0.568
Maximum	0.741	0.930	0.825	0.585
Minimum	**0.744**	0.929	**0.826**	0.591
Majority Vote	0.740	**0.934**	**0.826**	**0.635**
	Support Vector Machine			
SVMP	0.490	0.700	0.576	0.500
SVMR	**0.708**	**0.728**	**0.709**	**0.627**
Average	0.695	0.722	0.678	0.583
Product	0.682	0.714	0.664	0.568
Maximum	0.697	0.723	0.680	0.585
Minimum	0.702	0.726	0.685	0.591
Majority Vote	0.699	0.724	0.679	0.584
	Artificial Neural Network			
MP	0.719	0.738	0.717	0.634
VP	0.703	0.726	0.701	0.614
Average	**0.769**	0.896	0.827	0.634
Product	**0.769**	0.894	0.825	**0.645**
Maximum	0.758	0.894	0.820	0.643
Minimum	0.717	0.737	0.712	0.625
Majority Vote	0.764	**0.904**	**0.828**	0.625
	K-Nearest Neighbor			
1NN	0.699	0.724	0.677	0.582
5NN	0.691	0.718	0.688	0.598
Average	0.745	0.917	0.822	0.592
Product	0.739	**0.937**	**0.826**	**0.601**
Maximum	**0.749**	0.899	0.817	0.597
Minimum	0.745	0.917	0.822	0.592
Majority Vote	0.742	0.914	0.819	0.587

Table 7
Performance comparison of new wrapper method and other feature-selection methods for HMEQ dataset

	Precision	Recall	F-Measure	ROC Area
	Decision Tree			
J48	0.859	0.864	0.844	0.795
RF	0.857	0.860	0.838	0.785
Average	0.867	0.982	**0.921**	0.793
Product	0.863	**0.983**	0.918	0.787
Maximum	**0.914**	0.899	0.906	**0.809**
Minimum	0.855	0.852	0.853	0.806
Majority Vote	0.868	0.979	0.920	0.797
	Support Vector Machine			
SVMP	0.633	0.796	0.705	0.555
SVMR	**0.843**	0.804	0.724	0.619
Average	0.827	0.977	0.896	0.701
Product	0.809	0.815	0.759	0.662
Maximum	0.816	0.822	0.774	0.683
Minimum	0.800	0.819	0.778	**0.691**
Majority Vote	0.824	**0.987**	**0.898**	0.682
	Artificial Neural Network			
MP	0.693	0.638	0.664	0.677
VP	0.81	0.827	0.789	0.607
Average	0.868	0.871	0.869	0.877
Product	0.835	**0.977**	**0.902**	0.602
Maximum	0.811	0.829	0.793	0.734
Minimum	0.838	0.974	0.901	0.732
Majority Vote	**0.911**	0.930	0.920	**0.879**
	K-Nearest Neighbor			
1NN	0.852	0.837	0.791	0.803
5NN	0.837	0.824	0.766	0.812
Average	0.821	**0.998**	0.901	**0.891**
Product	0.850	0.825	0.766	0.881
Maximum	**0.889**	0.997	**0.940**	0.889
Minimum	0.821	0.996	0.900	0.842
Majority Vote	0.832	0.996	0.907	0.844

Table 8
Performance comparison of new wrapper method and other feature-selection methods for Tunisian dataset

	Precision	Recall	F-Measure	ROC Area
	Decision Tree			
J48	0.722	0.850	0.781	0.597
RF	0.797	0.846	0.801	**0.695**
Average	0.858	0.985	0.917	0.652
Product	0.859	0.985	0.918	0.655
Maximum	**0.866**	0.985	**0.921**	0.653
Minimum	0.861	**0.986**	0.919	0.644
Majority Vote	0.858	0.987	0.917	0.649
	Support Vector Machine			
SVMP	0.722	0.850	0.781	0.500
SVMR	0.797	0.837	0.805	0.566
Average	**0.861**	0.962	0.909	**0.666**
Product	0.710	0.842	0.770	0.500
Maximum	0.860	**0.968**	**0.911**	0.563
Minimum	0.798	0.839	0.803	0.661
Majority Vote	0.859	**0.968**	0.910	0.656
	Artificial Neural Network			
MP	0.802	0.843	0.800	0.577
VP	0.826	0.857	0.816	0.562
Average	0.856	0.979	0.913	0.677
Product	0.865	**0.984**	**0.921**	0.659
Maximum	0.867	0.975	0.918	0.668
Minimum	**0.888**	0.855	0.871	**0.731**
Majority Vote	0.866	0.981	0.920	0.657
	K-Nearest Neighbor			
1NN	0.785	0.843	0.794	0.680
5NN	0.792	0.844	0.800	0.685
Average	0.855	0.977	0.912	**0.775**
Product	0.852	0.993	0.917	0.756
Maximum	0.864	0.925	0.893	0.746
Minimum	0.863	0.932	0.896	0.704
Majority Vote	**0.866**	**0.985**	**0.921**	0.753

The good performance of the wrapper using DT classifiers was guided by the nature of this family, which is well-known for its highly accurate performance on financial data [13]. Another important fact that can easily be seen in Tables 5, 6, 7, and 8 is the improvement of the results when using the combination processes of DT, which gave better results than the individual DT classifiers in all of the datasets. The combination rules for DT featured approximately the same performance. Concerning the results of the SVM family, we notice some differences among the individual results from the polynomial and radial SVMs in Tables 5, 6, 7, and 8. For the four datasets, we noticed that the performance with the radial SVM was slightly better. This result was due to the nature of the two kernels. In general, a polynomial kernel looks for linear characteristics within datasets, while a radial kernel identifies the linear and non-linear aspects of datasets. We also noticed that the same-type combinations with SVM improved the overall performance in the large datasets. This improvement was due to the selected features within the combination process (which were more suitable for the classification task). For example, a combination of majority vote with minimum and average rules gave significantly higher ROC area and F-measure rates in the Tunisian and HMEQ datasets. The good performance of the obtained combinations that used the SVM family was the result of their natural simplicity. Concerning the results of the ANN and KNN families, Tables 5, 6, 7, and 8 show that both the KNN and ANN classifiers always gave better results when the size of the dataset was small (as in the cases of the German and Australian datasets). For these datasets, the KNN and ANN combination rules resulted in higher classification performance.

We investigate the influence of the classifier family on the selected features. It is interesting to know whether the observed results were only due to the types of classifiers or if they were a result of their interactions with the aggregation methods. Hence, we use a two-way ANOVA to analyze whether the mean values of the F-measure significantly changed along with the levels of the two independent variables (the classifier and aggregation methods). The first independent variable classifier presented the first factor in the ANOVA analysis, where DT, SVM, ANN, and KNN presented the levels of this variable. The aggregation method presented the second factor in ANOVA, where $\{Average, Product, Maximum, Minimum, MajorityVote\}$ presented the levels of this second factor. To test the interaction, we use the hypotheses presented below.

For the first factor (Classifier), H_0 and H_1 are given by the following:

$$\begin{cases} H_0 : \mu^1_{DT} = \mu^1_{SVM} = \mu^1_{ANN} = \mu^1_{KNN} \text{ – performances of classifiers are equal} \\ \\ \text{versus} \\ \\ H_1 : \forall t, \mu^1_t \neq \mu^1_i, i, t \in \{DT, SVM, KNN, ANN\}, i \neq t \text{ – at least one classifer's mean performance is different than the others.} \end{cases} \tag{2}$$

H_0 and H_1 for Factor 2 (i.e., the aggregation method) would be as follows:

$$\begin{cases} H_0 : \mu^2_{Aver} = \mu^2_{Prod} = \mu^2_{Max} = \mu^2_{Min} = \mu^2_{MajV} - \text{performances of aggregation methods are equal} \\ \text{versus} \\ H_1 : \forall t, \mu^2_t \neq \mu^2_i, i, t \in \{Aver, Prod, Max, Min, MajV\}, i \neq t - \text{at least one aggregation method's mean performance is different than the others.} \end{cases} \tag{3}$$

The results that were obtained from the two-way ANOVA are summarized in Table 9.

Table 9
Tests of between-subject effects in wrapper framework

Source	Type III Sum of Squares	DF	Mean Square	F	Sig. (p-value)
Aggregation Method	0.013	**4**	0.003	0.768	**0.550**
Classifier	0.063	**3**	0.021	5.081	**0.003**
Aggregation Method * Classifier Error	0.015	**12**	0.001	0.301	**0.987**
	0.247	60	0.004	–	–
Corrected Total	0.338	79	–	–	–

Dependent Variable: F-measure

Table 10
Multiple comparison table for classifier levels in wrapper framework

Classifier (I)	Classifier (J)	Mean difference (I–J)	Sig.
ANN	DT	–0.01405	0.900
	KNN	–0.003	0.999
	SVM	0.05790*	0.030
DT	ANN	0.01405	0.900
	KNN	0.01105	0.948
	SVM	0.07195*	0.004
KNN	ANN	0.003	0.999
	DT	–0.01105	0.948
	SVM	0.06090*	0.020
SVM	ANN	–0.05790*	0.030
	DT	–0.07195*	0.004
	KNN	–0.06090*	0.020

The obtained results of the two-way ANOVA (shown in Table 9) show that we do not have a significant interaction between the two factors; this indicates that the impact on the outcome of any specific level change of the F-measure in one factor is

the same for every fixed setting of the other factors (p-value = 0.003). When ANOVA gave a significant result for one of the classification methods, this indicated that at least one classifier's results differed from the other classifiers. Yet, the ANOVA test did not indicate which classifier's results influenced the rejection of H_0. In order to analyze the pattern of the differences between the means, we follow the ANOVA results by pairwise comparisons. The results of these pairwise comparisons for the classifiers are given in Table 10. This table shows that there was a statistically significant difference between the obtained results from SVM and the others classifications.

4.2.2. Results and discussion for mixed-type approach

Given the large number of combinations, the mixed-type approach was evaluated using only the Australian dataset; these results are summarized in Tables 11 and 12. The first table presents the results for the two-classifier mixed combination, while the second presents those for the three-classifier mixed combination. We investigated the impact of the type of classifier and the combination number (two or three) on the feature-selection results.

Table 11
Total number of evaluated subsets and selected features by two classifiers, mixed-type combinations, and associated F-measure rates for Australian dataset

Lowest F-measure lies between 0.847 and 0.855	Intermediate F-measure lies between 0.856 and 0.859	Highest F-measure lies between 0.860 and 0.874
j48+1NN (79.3)	J48+ SVMP (106.4)	J48+ MP (116.7)
RF+SVMR (82.2)	J48+ SVMR (106.4)	RF+SVMP (79.3)
RF+MP (111.6)	J48 + VP (120.7)	RF+1NN (96.4)
RF+VP (104.5)	j48+5NN (105.4)	SVMP+VP (88.4)
RF+5NN (96.4)	SVMP+MP (112.5)	SVMP+1NN (79.3)
SVMP+5NN (116.6)	SVMR+MP (112.7)	MP+5NN (121.7)
SVMR+1NN (79.3)	SVMR+VP (116.7)	VP+5NN (117.7)
SVMR+5NN (127.9)		
MP+1NN (107.6)		
VP+1NN (127.6)		

From Tables 11 and 12, we notice that a combination with few classifiers can achieve the selection of the feature that gives the best F-measure with a smaller number of evaluated subsets. More specifically, the two-classifiers' combinations produced an F-measure that was within a range of [0.860*to*0.874] with a number of evaluated subsets that did not exceed 121 evaluations. On the other hand, the three-classifiers' combination gives the same rate but with a much higher number of evaluated subsets.

Table 11 shows that combining DT classifiers with ANN or KNN classifiers generally yields the lowest F-measures (RF+MP, RF+VP, RF+5NN, and J48+1NN); this

was due to the difference in the nature between these three types of classifiers. Actually, ANN classifiers identify the relationships between features based on the available prior knowledge about the actual features in a dataset. However, KNN classifiers select the most relevant features with the closest distance to a set of specified features that are called neighbors. For this family, the resulting features depend on the number of chosen neighbors. DT classifiers are theoretically different from ANN and KNN, which use a statistical measurement to evaluate the relevance of the features.

Table 12

Total number of evaluated subsets and selected features by three classifiers, mixed-type combinations, and associated F-measure rates for Australian dataset

Lowest F-measure lies between 0.847 and 0.855	Intermediate F-measure lies between 0.856 and 0.859	Highest F-measure lies between 0.860 and 0.874
J48+RF+MV (136.7)	J48+RF+SVMP (82.2)	J48+RF+1NN (79.3)
J48+RF+5NN (131.9)	J48+RF+SVMR (75.2)	J48+VP+1NN (139.7)
J48+1NN+5NN (114.7)	J48+RF+MP (144.10)	RF+MP+VP (111.6)
J48+SVMR+MP (146.7)	J48+MP+VM (126.7)	RF+SVMP+MP (132.6)
J48+SVMR+1NN (75.2)	J48+SVMP+SVMR (75.2)	RF+SVMP+VP (120.8)
J48+SVMR+5NN (141.10)	J48+SVMP+MP (126.7)	RF+SVMP+1NN (116.7)
J48+MP+5NN (122.7)	J48+SVMP+VP (139.6)	RF+SVMR+MP (126.9)
J48+VP+5NN (112.7)	J48+SVMP+1NN (118.6)	RF+SVMR+VP (135.7)
RF+1NN+5NN (79.3)	J48+SVMP+5NN (139.10)	SVMP+1NN+5NN (108.7)
RF+SVMP+5NN (94.5)	J48+SVMR+VP (189.7)	SVMP+MP+1NN (132.8)
RF+SVMR+1NN (88.3)	J48+MP+1NN (165.10)	SVMR+MP+5NN (132.10)
RF+SVMR+5NN (117.8)	RF+SVMP+SVMR (82.2)	SVMP+VP+1NN (118.10)
RF+MP+1NN (130.7)	MP+SVMP+SVMR (139.6)	SVMR+1NN+5NN (108.7)
RF+MP+5NN (133.10)	VP+SVMP+SVMR (120.5)	SVMR+MP+1NN (132.9)
RF+VP+1NN (130.7)	1NN+SVMP+SVMR (82.2)	SVMR+MP+5NN (149.9)
RF+VP+5NN (123.10)	5NN+SVMP+SVMR (75.2)	SVMR+VP+5NN (149.9)
	SVMP+MP+VP (109.5)	
	SVMP+VP+5NN (153.11)	
	SVMR+MP+VP (109.5)	
	SVMR+VP+1NN (122.8)	

Table 12 shows that the majority of those combinations with SVM classifiers selected sets of features that achieved the best rates of F-measure – especially the case when SVM classifiers were combined with KNN classifiers. The fact that these combinations led to high F-measure values despite the fact that they consider classifiers from different families could be due to the existence of particular similarities between these two families. KNN classifiers use a distance metric to decide which are the most relevant features for a target variable, while SVM classifiers use a distance metric to select the most relevant features by measuring the distance between each feature in accordance with the hyper-plane that separates the best class from the target concept.

5. Conclusion

The results that were obtained in this study imply that using feature selection as a pre-processing task helps credit-scoring models to be simpler to understand and faster to build and to have fewer features and better classification performance.

The main goal of our study was to get a robust and simple credit-scoring model; hence, we based our work on the idea that improving a credit-scoring model that detects applicants with bad credit (even by one percent) could lead to a significant decrease in losses for financial institutions. We addressed this issue by emphasizing that a more accurate credit scoring model could be achieved by using the most relevant features; from this comes the importance of feature selection. These merits might encourage us to carry out the necessary feature-selection processes in financial institutions.

In this study, we developed an ensemble wrapper feature-selection approach for a credit-scoring application. The proposed approach was composed of three stages. In the first stage, we performed a dimensionality reduction by using bank experts' knowledge. In the second stage, a heuristic was used to reduce the search space to fewer than ten features (which makes an exhaustive search easier). In the final stage, the generated subsets were evaluated using a multi-classifiers process that involved two arrangement approaches; namely, the same-type and mixed-type approaches. From the three stages, we showed that the use of prior information on relevant features effectively induced a significant gain in complexity with improved generalization. Also, we showed that the number of classifiers and their nature had an important impact on wrapper feature-selection results.

Future research would have to be done in order to draw more-generalized conclusions. Specifically, increasing the number of used classifiers in our evaluation framework can be used to consolidate the obtained conclusions, and other real datasets could also be included.

References

[1] Chan Y.H., Ng W.W.Y., Yeung D.S., Chan P.P.K.: Empirical comparison of forward and backward search strategies in L-GEM based feature selection with RBFNN. In: *ICMLC*, pp. 1524–1527, 2010.

[2] Chen F.L., Li F.C.: Combination of feature selection approaches with SVM in credit scoring, *Expert Systems with Applications*, vol. 37, pp. 4902–4909, 2010.

[3] Chrysostomou K., Chen S.Y., Liu X.: Combining multiple classifiers for wrapper feature selection, *International Journal of Data Mining, Modelling and Management*, vol. 1(1), pp. 91–102, 2008.

[4] Hayashi Y., Takano N.: One-Dimensional Convolutional Neural Networks with Feature Selection for Highly Concise Rule Extraction from Credit Scoring Datasets with Heterogeneous Attributes, *Electronics*, vol. 9(8), 2020. doi: 10.3390/electronics9081318.

[5] Hsieh N.C., Hung L.P.: A data driven ensemble classifier for credit scoring analysis, *Expert Systems with Applications*, vol. 37, pp. 534–545, 2010.

[6] Kozodoi N., Lessmann S., Papakonstantinou K., Gatsoulis Y., Baesens B.: A multi-objective approach for profit-driven feature selection in credit scoring, *Decision Support Systems*, vol. 120, pp. 106–117, 2019. doi: 10.1016/j.dss.2019.03.011.

[7] Kuncheva L.I., Bezdek J.C., Duin P.W.: Decision templates for multiple classifier fusion: an experimental comparison, *Pattern Recognition*, vol. 34, pp. 299–314, 2001.

[8] Liu H., Yu L.: Toward integrating feature selection algorithms for classification and clustering, *IEEE Transactions on Knowledge and Data Engineering*, vol. 17(4), pp. 491–502, 2005. doi: 10.1109/TKDE.2005.66.

[9] Liu Y., Schumann M.: Data mining feature selection for credit scoring models, *Journal of the Operational Research Society*, vol. 56, pp. 1099–1108, 2005.

[10] López J., Maldonado S.: Profit-based credit scoring based on robust optimization and feature selection, *Information Sciences*, vol. 500, pp. 190–202, 2019. doi: 10.1016/j.ins.2019.05.093.

[11] Nalić J., Martinović G., Žagar D.: New hybrid data mining model for credit scoring based on feature selection algorithm and ensemble classifiers, *Advanced Engineering Informatics*, vol. 45, 2020.

[12] Paleologo G., Elisseeff A., Antonini G.: Subagging for credit scoring models, *European Journal of Operational Research*, vol. 201(2), pp. 490–499, 2010.

[13] Piramuthu S.: Evaluating feature selection methods for learning in data mining applications, *European Journal of Operational Research*, vol. 156(2), pp. 483–494, 2004.

[14] Rodriguez-Lujan I., Huerta R., Elkan C., Cruz C.S.: Quadratic Programming Feature Selection, *Journal of Machine Learning Research*, vol. 11, pp. 1491–1516, 2010.

[15] Tripathi D., Edla D.R., Cheruku R., Kuppili V.: A novel hybrid credit scoring model based on ensemble feature selection and multilayer ensemble classification, *Computational Intelligence*, vol. 35(2), pp. 371–394, 2019. doi: 10.1111/coin.12200.

[16] Trivedi S.K.: A study on credit scoring modeling with different feature selection and machine learning approaches, *Technology in Society*, vol. 63, 2020. doi: 10.1016/j.techsoc.2020.101413.

[17] Šušteršič M., Mramor D., Zupan J.: Consumer credit scoring models with limited data, *Expert Systems with Applications*, vol. 36, pp. 4736–4744, 2009.

[18] Wang D., Zhang Z., Bai R., Mao Y.: A hybrid system with filter approach and multiple population genetic algorithm for feature selection in credit scoring, *Journal of Computational and Applied Mathematics*, vol. 329, pp. 307–321, 2018. doi: 10.1016/j.cam.2017.04.036.

[19] Yun C., Shin D., Jo H., Yang J., Kim S.: An Experimental Study on Feature Subset Selection Methods. In: *Proceedings of the 7th IEEE International Conference on Computer and Information Technology*, pp. 77–82, CIT '07, IEEE Computer Society, Washington, DC, USA, 2007.

[20] Zhang X., Zhou Z.: Credit Scoring Model based on Kernel Density Estimation and Support Vector Machine for Group Feature Selection. In: *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1829–1836, 2018.

Affiliations

Waad Bouaguel

University of Jeddah, College of Business, Jeddah, Saudi Arabia, University of Tunis, LARODEC, ISG, Tunisia, bouaguelwaad@gmail.com, ORCID ID: https://orcid.org/0000-0003-2171-0370

Received: 13.02.2021
Revised: 01.07.2021
Accepted: 28.08.2021