Original Research

Analysis of the performance of different classifiers in solving the credit risk scoring problem with noisy and clean data

Reza, Yousefi Zenouz[1]*. Fatemeh, Atapour Mashhad[1]

Received: 15 Feb 2017 / Accepted: 29 May 2023 / Published online: 29 May 2023

*Corresponding Author, reza.zenouz@gmail.com

1- Information Technology and Operations Management, Faculty of Management, Kharazmi Univerity

Abstract

Credit risk scoring is an important issue in the real business world. There are many different classifiers available to address this issue, but the determinant factor in evaluating a classifier's performance is its accuracy level. This factor becomes even more crucial when dealing with noisy data. In this paper, we aim to utilize a unique approach to construct 11 new classifiers by combining various non-parametric methods. These new classifiers complement each other's shortcomings, leading to improved performance and subsequently, higher accuracy rates. Enhancing classifier performance directly impacts the profits of banks and financial institutions that utilize them. This study also investigates the question of whether the number of classifiers is more effective or if their accuracy levels are more critical in building combined classifiers. The single classifiers used in these combinations include K-nearest neighbor, Support Vector Machine, Multi-layer Perceptron, and Decision Tree. Furthermore, all the combined classifiers are compared with two types of ensembles in terms of accuracy rate and robustness at different noise levels. We utilized a standard dataset from the UCI database for our analysis. All results have been compared using the Wilcoxon signed rank test. The findings indicate that ensembles, especially RobustBoost, outperform both multiples and single classifiers in terms of both accuracy rate and robustness. Additionally, multiple classifiers are generally better than single ones. Moreover, the results highlight that in building multiple classifiers, the accuracy rate and robustness of each building block are more important than the number of components used.

Keywords- Credit Risk Scoring; Noisy Data, Ensembles; Multiple Classifier

INTRODUCTION

One important issue that financial institutions encounter is credit scoring. It refers to the act of classifying customers usually into two groups of 'bad' and 'good' (Martínez Sánchez & Pérez Lechuga, 2016). Bad customers are those who have high potential to show undesirable behavior in repaying debts. Good customers are those who are prone to repay their debts according to the contract. The first group is so likely to cause the losses due to their unpaid debts and also violate the obligations (Wang, 2011). Based on some customer's characteristics such as age, loyalty, gender and job, the algorithms allocate the customers into one of the classes. This classification helps bankers make decisions in situations such whether to grant loans to an applicant or not (Finlay, 2011). Furthermore, bankers can utilize this classification to develop practical strategies for each customer group. In recent years, many different algorithms have been developed for credit scoring problem, each with its own strengths and limitations. Therefore, ongoing research aims to improve the efficiency of these algorithms. Given the high volume of deferred payments and demands, even a slight enhancement in algorithms efficiency would be highly appealing (hand &Henly 1997).

This improvement will enhance the accuracy and performance of the models, leading to proper credit scoring. Proper credit scoring, in turn, results in improved cash flow and a reduction of potential risks.(West, 2000).Generally the techniques that have been applied in credit scoring problem can be divided into two groups: parametric and nonparametric. Some parametric methods are Linear Discriminant Analysis (LDA) (Serrano-Cinca, 2013), Logistic Regression (LR) (Dong, 2010); (Bekhet & Eletter, 2014), Multivariate Adaptive Regression Splines (MARS)(T. S. Lee, & Chen, I. F. , 2005), Bayesian Networks (BN) (Kao, Chiu, & Chiu, 2012).… and non-parametric models include: Support Vector Machine (SVM)(Baesens, 2003a; Huysmans, 2005), Artificial Neural Networks (ANN)(West, 2000), Classification and Regression Tree (CART) (T.-S. Lee, Chiu, Chou, & Lu, 2006) and K-Nearest Neighbor (KNN)(Hand, 1997).

LDA is one of the first and most popular parametric methods.(Durand, 1941; Myers, 1963; Serrano-Cinca et al., 2013) The underlying assumption of this method is linearity, namely it supposes the linear relationship between input and output variables. This method has been widely criticized because the mentioned presumption is difficult to meet and is sensitive to multivariate normality(West, 2000). The parametric methods perform with high accuracy rates if all underlying assumptions are satisfied, but achieving these conditions is very hard. The other types of classification methods are nonparametric models that majority of them have been developed in the field of artificial intelligence. These methods have some advantages over their parametric competitors. They are free from such stringent assumptions and can extract information from a given training data set without any prior knowledge about the problem (C.-l. Huang, Chen, & Wang, 2007; Z. Huang, Chen, Hsu, Chen, & Wu, 2004; P. Martens 2010). So due to the mentioned reasons, several studies indicated that, on average non-parametric methods, outperform the parametric methods.(Crook, 2007). But there is no overall rule for determining which classifier is the best, as the performance of each algorithm is dependent on the specific problem. Therefore, to solve a problem effectively, one needs to try different methods and choose the approach that outperforms the others.(Marqu et al., 2012)

Due to these shortages, there is a hypothesis that suggests the possibility of using compound classifiers to overcome the deficiencies or even improve them. Both ensemble and multiple classifiers have compound algorithms. Some studies have demonstrated that in certain cases multiple and ensemble classifiers outperform single classifiers. (Twala, 2010; Wang, Hao, Ma, & Jiang, 2011; West, Dellana, & Qian, 2005) There is an important difference between ensembles (Random Forest, AdaBoost, Boosting), and multiple classifiers. Ensemble classifiers are based on a specific single classifier and operate on different training data sets, combining their predictions. In contrast, multiple classifiers apply the same training data set to different single classifiers, and the results are then combined in various ways.
The efficiency of multiple classifiers is directly related to the diversity of the single classifiers that constitute it. Diversity leads to complementary behaviour, as different algorithms can compensate for each other's drawbacks. Therefore, when using multiple classifiers, the most important aspect is to choose as many diverse single classifiers as possible to achieve greater accuracy in the results. (Ali & Pazzani, 1996; Bian & Wang, 2007; Sáez, Galar, Luengo, & Herrera, 2013).

To gain a realistic view of the performance of different algorithms, they should be evaluated in real situations. The most important factor in real problems that affects the classifier's performance is the presence of noise specially when the classifier is single. Some recent studies have demonstrated that combining the prediction of classifiers can significantly improve the classifier’s accuracy rate in noisy data situations. (Sáez et al., 2013). Most of the studies on credit risk scoring in noisy data, have applied ensembles like bagging or boosting. (Dietterich, 2000; Maclin & Opitz, 1997) But a few of them focus on multiple algorithms which made up of single classifiers, and compare all the existing ways. (Sáez et al., 2013). Dietterich had a study in this field. The results of this paper indicate that when the noise level is low, boosting is more accurate than bagging and randomization, but in higher levels of noise, bagging outperforms other techniques(Dietterich, 2000). Similar result was obtained by another study(Maclin & Opitz, 1997). Some studies deal with the imbalanced data (T.khoshgoftaar, 2011). The results revealed that in total, bagging approach outperforms the boosting. Saez investigated five multiple classifiers under noisy conditions(Sáez et al., 2013). Although there are so many studies in the field of credit risk scoring, but there are few studies that considered noisy data. In this paper, to conduct a comprehensive study, we aim to test four of the most important non-parametric classifiers (KNN, CART, SVM, MLP), along with all possible combinations of them (multiple classifiers), and two ensembles (AdaBoost, RobustBoost) across five levels of noise. For all these five levels, we investigate and compare the accuracy rate and robustness of all classifiers (multiples, ensembles, single non-parametric) using the Wilcoxon signed rank test. Additionally, we intend to address the question of whether the number of blocks in building multiple classifiers is more important and effective than the block's accuracy rate. The remainder of this paper is organized as follows: In section2, the details of methods are presented. Section3 deals with the details of the experimental design. Section 4 shows the results. The analysis and conclusions are covered respectively in section 5 and 6.

METHODOLOGY

This paper deals with the credit scoring problem under noisy data. Different single, multiple and ensemble classifiers were implemented and their performance and robustness were compared. Among the top 10 algorithms used in the field of data mining (Wu, 2007), four of them, selected for their diverse paradigms and outstanding robustness to noise, are: Support Vector Machine (SVM) (C. Cortes, 1995), K-Nearest Neighbor (KNN) (McLachlan, 2004), Multilayer Perceptron (MLP) (West, 2000), and Classification and Regression Tree (CART) (T.-S. Lee et al., 2006). From the different types of ensembles, Adaboost (Mar et al., 2013) and RobustBoost (Wang et al., 2011) were chosen. Additionally, the single classifiers were combined in pairs, triplets, and all together.

- *Support Vector Machine (SVM)*

Support Vector Machine (SVM), proposed by V. Vapnik in 1995, is arguably the most popular machine learning algorithm used for classification and regression problems in recent years. The primary goal of SVM is to find the hyperplane that maximizes the margin width between the hyperplane and the samples. According to statistical learning theory, achieving this maximization would minimize the model's complexity and general risk of error.
For non-linear data that cannot be separated by a hyperplane, a soft margin is utilized. Since most real-world data exhibits non-linear relationships, training examples are allowed to have some slack on the wrong side of the margin to separate them. This is achieved by considering a given penalty proportion that restricts the distances of the slacks from the hyperplane. While maximizing the margin width, the sum of penalties should be minimized.
A parameter known as 'C' controls the cost associated with the goals in the overall optimization problem.(Harris, 2015; Hens & Tiwari, 2012) (Harris, 2013) If we consider $y_i \in \{-1,1\}$ for all $i = 1, 2, ..., n$,we can model the SVM problem as the following:

$$\max(\sum_{i=1}^{n}\alpha_i + \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j k(\mathrm{x}_i, \mathrm{x}_j)) \quad (1)$$

$s.t:$

$$0 \le \alpha_i \le C \quad i = 1, 2, ..., n \quad (2)$$

$$\sum_{i=1}^{n} y_i \alpha_i = 0 \quad (3)$$

In the above equation, α_i indicates the Lagrange multiplier for training example. k demonstrates the kernel function which is used for non-linear data. The commonly used kernel functions, are(Eletter, 2014):

Linear: $K(\mathrm{x}_i, \mathrm{x}_j) = \mathrm{x}_i \,.\, \mathrm{x}_j$

Polynomial: $K(\mathrm{x}_i, \mathrm{x}_j) = (\mathrm{x}_i \,.\, \mathrm{x}_j + 1)^d$

Radial basis function (RBF): $\mathrm{K}(\mathrm{x}_i, \mathrm{x}_j) = \exp(-\frac{\| \mathrm{x}_i - \mathrm{x}_j \|^2}{2\sigma^2})$

In this paper, we used RBF as a kernel function and also grid search to obtain the optimal values of the parameters of the SVM model.

K-nearest Neighbor (KNN)

The K-nearest neighbor algorithm is a non-parametric method that classifies the data according to the majority vote of its most similar neighbors. We assumed k=20 as the optimal value of neighbors in this study. The Euclidean distance between two points is used as a measure of similarity (Brown, 2012,(Ziegler, 2013)

$$d(\mathrm{x}_i, \mathrm{x}_j) = \| \mathrm{x}_i - \mathrm{x}_j \| = [(\mathrm{x}_i - \mathrm{x}_j)^T (\mathrm{x}_i - \mathrm{x}_j)]^{\frac{1}{2}} \quad (4)$$

- *Multilayer Perceptron (MLP)*

MLP is the most popular type of Artificial Neural Networks (ANNs) with a feed-forward scheme, comprising one input layer, one output layer, and any number of hidden layers. While the number of hidden layers can theoretically be any value, in practice, networks with one or two hidden layers are preferred. The number of hidden layers and their nodes are the two main factors that influence the performance of MLP. Properly determining these factors is crucial for achieving optimal performance.

MLP is particularly well-suited for solving nonlinear problems that involve non-linear relationships between variables. The data is forwarded through the input layer to the output layer. In the output layer, the results are calculated and compared with the target values. The difference between the actual and target values is then sent as a signal back to the input layer. This process continues until the actual and target values converge.

To avoid infinite iterations, a rule must be established to stop the process (Tsai, 2008). The general architecture of MLPs is shown in the figure (1) below:

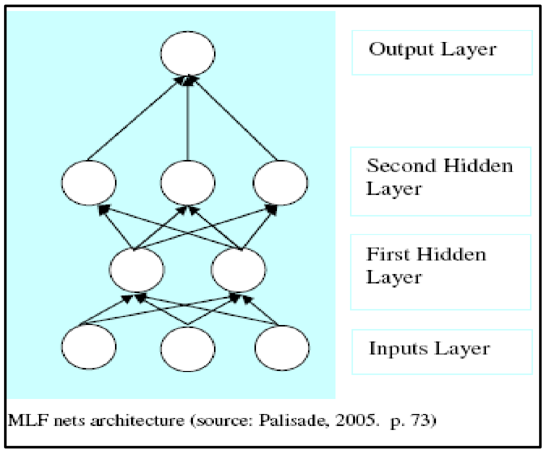

FIGURE 1
MLP ARCHITECTURE

- *Classification and regression Tree (CART)*

CART was developed by Breiman et al. (1984). In this method during a statistical procedure by using historical data a kind of decision tree is constructed. The splitting criteria can be 'gdi', 'twoing', 'deviance'. After the completion of the tree building stages, the pruning process starts. In this paper, gdi (Gini's diversity index) is utilized as splitting criteria. (T.-S. Lee et al., 2006)

- *Adaptive Boosting (AdaBoost)*

AdaBoost algorithm that was proposed by Freund and Schapire (Freund.Y and Schapire.E, pp, 148-156, & 1996.) constitutes the best known member in boosting family. Using successive bootstrap samples, a sequence of base classifiers $C_1, C_2, ..., C_M$ is generated. T_i is obtained by weighting the training instances in M iterations. At the first step, all of the training samples are weighted equally and at each iteration, these weights are adjusted according to the misclassification errors. Thus, samples that are misclassified by C_{i-1}, are more likely to appear in the next bootstrap sample T_i . The final decision is made according to a weighted vote of the classifiers. (W_i , the weight of each classifier C_i is computed according to the performance of the weighted sample T_i that was trained.)(Marqués, 2012). Adaboost and generally Boosting technique improve the performance of weak classifiers but on the other hand these techniques are more sensitive to the noisy data and outliers.

- *RobustBoost*

Like other Boosting algorithms, also in AdaBoost the weights of misclassified observations increase at every boosting step. These weights can grow very fast if there is not an upper bound to them. In this situation, the boosting algorithm sometimes focuses on a few misclassified observations and disregards the majority of training data. As a result, the mean classification's errors increase. In these conditions, using RobustBoost is recommended. When applying this algorithm, the weights of misclassified observations are not increased for the entire data. Hence it can produce a better classification accuracy average. Instead of minimizing a specific loss function, Robust Boost maximizes the number of observations with the classification margin more than a certain threshold. (Freund, 2009) In this paper the Robust Boost algorithm is implemented with two different error goals.

EXPERIMENTAL FRAMEWORK

- *Data sets*

In this study, we applied the widely-used German data set from UCI Machine Learning Database Repository (http://archive.ics.uci.edu/ml/). It consists of 1000 samples that 700 of them are creditworthy and the others are grouped as bad applicants. This data set has 24 attributes or input variables for any of its instances which describe the applicant. This set of attributes includes credit history, account balances, loan amount, personal information, age, job title, housing, loan purpose and so on.

- *Noisy data*

Since real-world data are noisy, we examined different classifiers' performances and compared them in the presence of noise. This noise can corrupt the performance of classifiers used for credit risk assessment This effect intensifies, particularly when the classifier is sensitive to noise, making it very challenging to classify customers accurately. Recent studies have shown that combining classifiers improves their performance (Sáez et al., 2013). Therefore, in this paper, we built 11 different combinations of classifiers. These multiple classifiers were constructed to enhance the accuracy rate and performance of the single classifiers.

As mentioned by Sáez, the worst and most corruptive noise is uniform class noise (Sáez et al., 2013). Hence, we used this type of noise, which occurs when examples are labeled incorrectly. For example, a dataset with 15% noise means that 15% of the samples' labels are changed. To investigate the effect of noise on single, multiple, and ensemble classifiers, we utilized 5 levels of noise: 0%, 5%, 15%, 25%, and 35%. Initially, we assumed that the data set used in this paper is noise-free, and then different levels of noise were introduced into it. For instance, to have a data set with 15% noise, we randomly selected 15% of the samples and changed their labels. As a result, we obtained 5 data sets, and the performances of all 17 classifiers (11 multiple, 2 ensemble, 4 single) were tested under these five levels of noise. It should be noted that before introducing noise into the original data, normalization was performed as a preprocessing phase.

- *Single classifiers vs. multiple classifiers*

For a given problem, finding the best classification algorithm is not easy, as no single classifier dominates all others in all domains. This means that a classifier may excel in one domain, but there is no guarantee that it would perform well in other situations. Hence, there is a good reason to use Multiple Classifier Systems (MCSs). By employing MCSs, there is no need to choose just one algorithm to solve the problem. Instead, several classifiers are used, allowing them to leverage their advantages and compensate for each other's disadvantages, leading to more complete and comprehensive coverage. (Sáez et al., 2013)

- *Classifier's performance*

In order to compare the performance of different classifiers, we used the accuracy rate as a standard comparison index. It can be calculated based on confusion matrix. The confusion matrix is given as follows.

TABLE1
CONFUSION MATRIX FOR EACH CLASSIFIER

The class one data that's classified correctly	The class one data that's classified incorrectly
The class two data that's classified incorrectly	The class two data that's classified correctly

$$\text{Accuracy}= \frac{C(1,1)+C(2,2)}{C(1,1)+C(1,2)+C(2,1)+C(2,2)} \qquad (5)$$

The accuracy rate for each classifier can be calculated as mentioned above. The other criterion is Robustness, which is used when the situation is noisy. The Robustness criterion shows how much the classifier's performance differs in noisy and clean situations (Freund.Y and Schapire.E et al.).

$$RLA_{x\%} = \frac{ACC_{0\%} - ACC_{x\%}}{ACC_{0\%}}$$

- *The multiple classifiers list*

Considering the mentioned single classifiers (SVM, MLP, CART, KNN) the list of MCSs have been built and examined in this paper is written as follow:

First, combine all of them

1) SMKC: that made up of (SVM-MLP-KNN-CART)

Second: Triple combination

2) SMK: (SVM-MLP-KNN)

3) SMC: (SVM-MLP-CART)

4) SKC: (SVM-KNN-CART)

5) MKC: (MLP-KNN-CART)

Third: Double Combination

6) SM: (SVM-MLP)

7) SK: (SVM-KNN)

8) SC: (SVM-CART)

9) MK: (MLP-KNN)

10) MC: (MLP-CART)

11) KC: (KNN-CART)

- *How to combine*

In this paper, we employ a parallel strategy to combine the single classifiers. Under this strategy, all the single classifiers are trained and tested using the same data sets. The accuracy rate of each classifier is normalized, resulting in a vector that represents the weight of each classifier, known as the 'weight vector.' Subsequently, each classifier predicts a portion of test samples randomly, based on its normalized weight. Finally, the actual outputs of each classifier are compared with their target values, and the accuracy rate of each Multiple Classifier System (MCS) is obtained.

When building multiple classifiers, the single classifiers with high performance are given greater participation and have a higher likelihood of being selected to predict the test sample's label. Hence, the classifier with a higher accuracy rate has a greater allocation in building the MCS. Fig (2) shows the procedure of multiple classifiers (MCSs) construction:

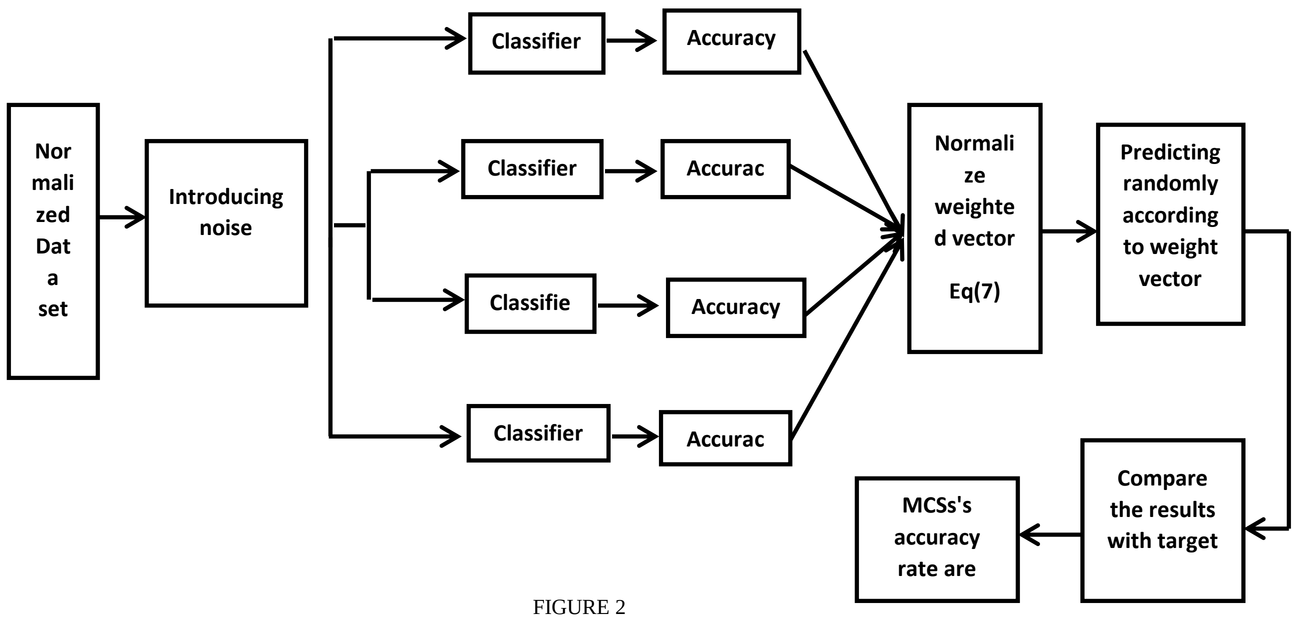

FIGURE 2
BRIEF PROCEDURE OF BUILDING MCSS.

$$W = [a_1, a_2, \ldots\ldots, a_n]; \tag{7}$$

$$a_1 + a_2 + \ldots\ldots + a_n = 1;$$

(Freund.Y and Schapire.E et al.)

The vector (W) consists of the normalized accuracy rates of the classifiers' blocks. Each element in the vector represents the proportion of each classifier in contributing to the final result. In other words, a1% of the test samples are predicted by classifier one, a2% by classifier two, and so on. In this study, we randomly selected 300 instances from our dataset, which is composed of 1000 samples, for the testing phase. These 300 instances are then divided according to the members of the (W) vector, and each classifier predicts samples based on its normalized weight. To illustrate this approach, let's consider an example: Classifier 'SMKC' & W = [0.3, 0.2, 0.3, 0.2], and the number of test samples is 300. So, SVM should predict the label of 90 test samples (0.3 * 300 = 90) that were randomly chosen from the 300 instances. CART, KNN, and MLP should predict 60, 90, and 60 samples, respectively, in a similar manner. Figure (3) provides a brief overview of this example.

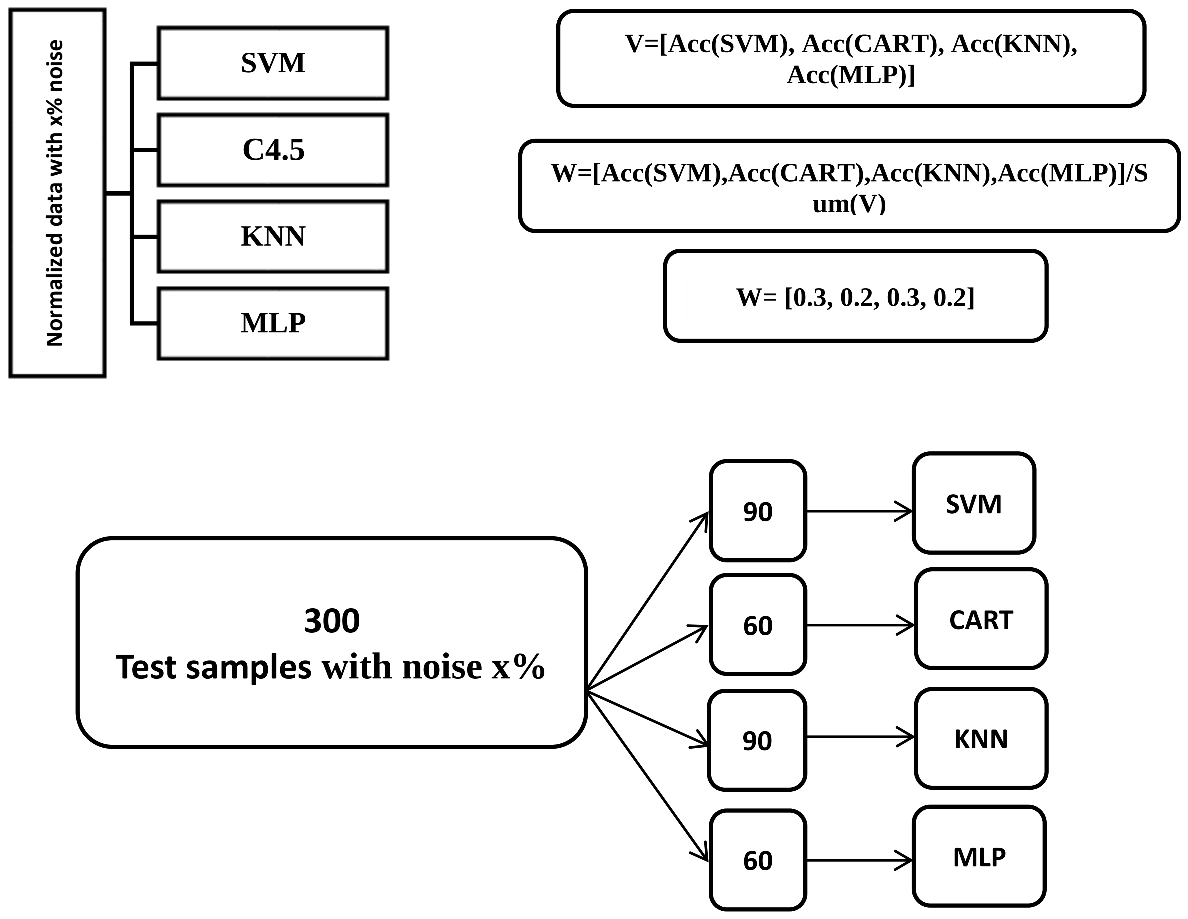
Normalized data with x% noise
SVM
C4.5
KNN
MLP
V=[Acc(SVM), Acc(CART), Acc(KNN), Acc(MLP)]
W=[Acc(SVM),Acc(CART),Acc(KNN),Acc(MLP)]/Sum(V)
W= [0.3, 0.2, 0.3, 0.2]
300
Test samples with noise x%
90
60
90
60
SVM
CART
KNN
MLP

FIGURE 3
SMKC BUILDING PROCEDURE

Now, the predicted classes of the sample tests should be compared with the actual labels, and the accuracy rate of SMKC is obtained. Each classification algorithm has certain parameters that need to be optimized. In this study, the parameters of the algorithms are determined and optimized, and they are presented in Table (2) below.

TABLE 2
CLASSIFICATION ALGORITHM'S PARAMETER

SVM	KNN	MLP	CART
δ, C optimized: Grid search for each data set **Kernel function: RBF** **Data preprocessing: normalization in [0 1]** **Train/test=7/3**	Number of neighbors: 20 Distance function: Euclidian Train/test=7/3	Number of hidden layers: 1 6 neurons in hidden layer 24 neurons in input layer 1 node in output layer Train/test=7/3	Prune after tree building Min leaf=1 Split criterion= 'gdi' Train/test=7/3
Ada Boost		Robust Boost	
Learn rate: 0.1 **Base classifier: Tree** **Train/test=7/3**		Error goal: 0.15, .01 Base classifier: Tree Train/test=7/3	

- *Data Analysis*

Finally, the performances of the classifiers were compared using Wilcoxon's signed rank test, a non-parametric pair-wise test. This test is an effective tool for identifying significant differences between each pair of algorithms. Single, multiple, and ensemble classifiers were compared at each level of noise. The test output provides the p-value, which represents the lowest level of significance of a hypothesis that results in a rejection. It enables us to determine whether two algorithms are significantly different and the degree of this difference. To utilize the Wilcoxon signed rank test, we implemented all the mentioned algorithms at least ten times, each time with randomly chosen training and testing samples in the same proportion (7/3).

In other words, the 1000-sample dataset was divided into two parts, training and testing, with a proportion of (7/3), and this process was repeated ten times. Subsequently, all the classifiers were run ten times using these constructed sets of training and testing data. To avoid extending the study beyond feasibility, we present here the average results of these ten runs for each classifier at each noise level. When dealing with noisy data, the robustness of classifiers, which indicates how much noise affects their performance, becomes more important than their accuracy rate. This is because a classifier may have a good accuracy rate but be sensitive to noise. Therefore, we utilized both the accuracy and robustness of the classifiers to investigate the results.

RESULTS AND FINDINGS

The results of the classifiers' accuracy are presented in Table (3). As mentioned, Table (3) displays the averages of the accuracy rates of the classifiers. Similarly, the averages of the classifiers' robustness are shown in Table (4). The Robustness Loss Analysis (RLA) indicates the difference between a classifier's performance under noisy data and its performance in a clean situation. Consequently, the RLA is zero for the original dataset, which is considered as clean data.

TABLE 3
AVERAGE ACCURACY RATE OF CLASSIFIERS IN 5 LEVEL OF NOISE

Noise level	0%	5%	15%	25%	35%
CART	84.9	84.57	82.26	79.16	74.4
MLP	85	84.06	83.1	78.99	73.23
KNN	84.89	83.4	82.73	75.26	71.53
SVM	85.98	85.17	83.5	79.23	76.57
AdaBoost	86.8	85.5	82.36	78.36	78.27
RobustBoost (error rate=.15)	86.2	85.7	83.99	79.13	77.73
RobustBoost (error rate=.01)	86.11	85.00	84.78	82.79	79.5
SMKC	84.3	84.03	83.04	78.93	76.89
SMK	83.75	82.6	81.83	77.67	75.71
SMC	83.00	82.84	82.72	78.00	76.03
SKC	84.58	84.02	83.25	79.36	76.64
MKC	82.7	82	81.87	78.36	75.8
SM	86.91	81.87	81.22	78.23	74.57
SK	84.9	83.7	82.83	78.30	75.83
SC	85.8	84.64	83.30	79.79	76.04
MK	85.4	83.2	82.99	78.13	75.5
MC	85.71	83.61	83.13	79.2	76.4
KC	85.5	83.94	83.79	80.06	76.77

ANALYSIS OF THE RESULTS

- *Performance results analysis*

Multiples vs. single

In this section our focus is on comparing each multiple classifier with its constituent building blocks..

- In the original data set, the classifiers like SMKC (84.3%), SKC (84.58%), MKC (82.7%), SM (86.91%), SK (84.9%), SC (85.8%), MC (85.71%) outperform their components.

TABLE 4
AVERAGE ROBUSTNESS OF CLASSIFIERS.IN 5 LEVELS OF NOISE

Noise Level	5%	15%	25%	35%
CART	0.0039	0.0311	0.0676	0.123
MLP	0.0111	0.0224	0.0707	0.138
KNN	0.0176	0.0254	0.113	0.157
SVM	0.0094	0.0288	0.0785	0.109
AdaBoost	0.015	0.0512	0.0972	0.0983
RobustBoost (error rate=.15)	0.0058	0.0256	0.0820	0.0983
RobustBoost (error rate=.01)	0.0129	0.0154	0.0386	0.0768
SMKC	0.0032	0.0149	0.0637	0.0879
SMK	0.0137	0.0325	0.0726	0.0960
SMC	0.0019	0.0034	0.0602	0.0840
SKC	0.0066	0.0157	0.0617	0.0939
MKC	0.0085	0.0100	0.0525	0.0834
SM	0.0580	0.0655	0.0999	0.1420
SK	0.0141	0.0244	0.0777	0.1068
SC	0.0135	0.0291	0.0700	0.1138
MK	0.0258	0.0282	0.0851	0.1159
MC	0.0245	0.0301	0.0760	0.1086
KC	0.0182	0.0200	0.0636	0.1021

- At the 5% noise level, SMKC (84.03%), SMC (82.84%), MKC (82%), SM (81.87%), SK (83.7%), SC (84.64%), MC (83.61%), KC (83.94%) are the multiple classifiers that outperform all their constituent components.
- When the noise level increases by 15%, all the classifiers mentioned in 5% retain their excellent performance and superiority over their components.
- At the noise level of 25%, some of the classifiers mentioned in the previous levels of noise, which previously performed better than their components, lose this superiority and do not outperform their members at this level. SMKC (78.93%), MKC (78.36%), SM (78.23%), SK (78.30%), MC (79.2%) are the classifiers that lost their excellence. Additionally, some classifiers not listed as good classifiers in lower levels, become good classifier at this level such as SKC (79.36%).

- Up to the 35% noise level, the SMK classifier which was the weakest classifier in lower levels, becomes the best performer at this level with an accuracy rate of 75.71%.

  As mentioned above in any noise level, some classifiers perform better than their components. However, KC, SKC and SC consistently outperformed their members in all noise levels.

- The best and worst classifiers in each noise level

- In the clean situation (0%), the SM classifier with an 86.91% accuracy rate is the best and the MKC with an 82.7% accuracy rate is the worst.
- In 5% noise level, the best classifier is robust Boost(Error Rate = 0.15) (85.7%) and the worst is SMK (82.6%).
- When the noise increases to 15%, the worst classifier belongs to SMK with an 81.83% accuracy rate. The best one is RobustBoost (Error Rate = 0.01) with an 84.78% accuracy rate.
- In the 25% noise level, RobustBoost (Error Rate = 0.01) with an 82.79% accuracy rate is the best, while KNN with a 75.26% accuracy rate is the worst among all classifiers.
- RobustBoost(Error Rate=0.01) achieves the best performance with a 79.5% accuracy rate in the 35% noise level. In this level, the worst classifier is KNN with a 71.53% accuracy rate .

TABLE5
THE BEST AND WORST CLASSIFIERS IN DIFFERENT LEVELS OF NOISE ACCORDING TO ACCURACY RATE CRITERION

	Worst AR among all	Best AR among all	Worst AR among MCSs	Best AR among MCSs
0%	MKC	SM	MKC	SM
5%	SM	RB1	SM	SC
15%	SMK	RB2	SMK	KC
25%	KNN	RB2	SMK	SC
35%	KNN	RB2	SM	SMKC

As the table shows, ensembles primarily demonstrate the best performance. Additionally, we can observe that the KNN algorithm exhibits the worst performance in the last two levels of noise. This is because the KNN algorithm relies on its neighbors, and as a result, noise can negatively impact this dependency, leading to a decrease in its performance. There is a diagram below that compares the performance of 18 classifiers in each noise level.

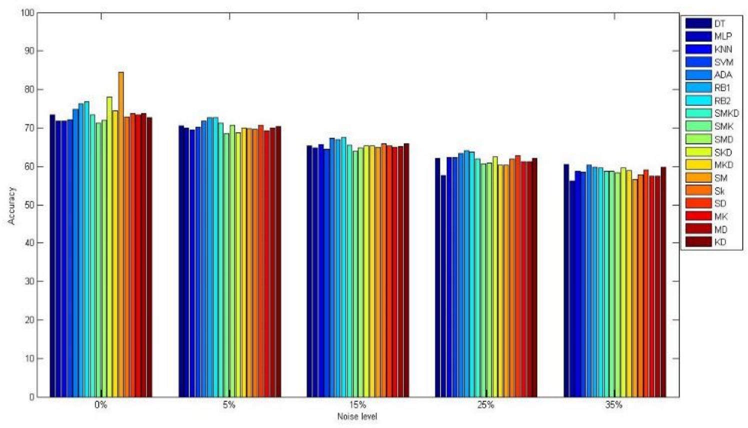

FIGURE 4
CLASSIFIERS' PERFORMANCE IN DIFFERENT LEVELS OF NOISE.

- Robustness results analysis

As mentioned before, in the 5% noise level, RLA indicates the classifier's robustness against noise. In other words, it quantifies the difference in a classifier's performance between noisy and clean data. Therefore, the minimized RLA represents the best performance, indicating little difference between the two scenarios.

- In 5% level noise, the SMC with RLA =0.0019 is the most robust classifier among all.
- In 15% noise, the MKC algorithm that its RLA is equal to 0.01 is the most robust classifier.
- The RobustBoost classifier (Error Rate=0.01) is the most robust classifier in 25% noise level.
- When the noise level changes into 35%, the most robust classifier still belongs to RB2 with RLA=0.0768.

As the results showed, the Rb2 classifier outperforms all others according to both criteria considered in this paper. Not only is it the most robust classifier, but its performance, as measured by the accuracy rate, is also very good.

TABLE 6
THE BEST AND WORST CLASSIFIERS IN DIFFERENT LEVELS OF NOISE ACCORDING TO ROBUSTNESS CRITERION

	Worst RLA among all	Best RLA among all	Worst RLA among MCSs	Best RLA among MCSs
5%	SM	SMC	SM	SMC
15%	SM	MKC	SM	MKC
25%	MK	Rb2	MK	MKC
35%	KNN	Rb2	SM	MKC

In noisy data, the robustness factor is more important than the accuracy rate. Therefore, in this paper, Table (Freund.Y and Schapire.E et al.) holds greater significance than Table (5). The results reveal that despite the SM classifier's accuracy rate being the best in the clean situation, it is highly sensitive to noise and exhibits the worst performance among all classifiers in any noise level. Thus, it is not a robust classifier and is not preferable in noisy situations. On the other hand, RB2 and MKC are among the most robust classifiers in three noise levels. Therefore, these multiple classifiers can be chosen in noisy situations as robust alternatives.

CONCLUSION

The main purpose of this study is to compare most of the classifiers used in the field of credit risk scoring. This comparison includes single classifiers, such as SVM, MLP, CART, and KNN, as well as all the combinations of them as multiple classifiers. Ensemble classifiers are also included in this comparison. The German dataset from the UCI Machine Learning Database Repository was utilized, and as the first step, standard normalization was applied for data preprocessing. We ran all 18 classifiers with 5 datasets that include different levels of noise (0%, 5%, 15%, 25%, 35%). Each tryout was examined ten times to use the Wilcoxon signed-rank test and reduce the probability of chance results. This means that we have ten datasets with a 5% noise level, and all the classifiers are run with these ten 5% noise level datasets. The same process is repeated for other datasets with 15%, 25%, and 35% noise levels. In all these tests. We considered two criteria to evaluate classifiers: 'Accuracy Rate' and 'Robustness'. In the case of noisy data, the robustness of the classifier is more important than its accuracy rate. Finally, after conducting these extensive examinations, we can summarize the results as follows: Multiple classifiers significantly outperform the single ones, but this cannot be generalized for all cases. Among all multiples, SC, KC, SM, and SMKC were the best based on the accuracy rate criterion in this study. All three ensembles performed better than all other classifiers, regardless of the noise level. The SM classifier showed good performance in the original dataset (0% noise level), but in noisy data, it became one of the worst classifiers due to its sensitivity to noise. Therefore, SM is not preferable in noisy data scenarios. Among all multiples, SMC and MKC were the best based on the robustness criterion. Another finding of this study is that, in building multiple classifiers, the quality of members is more important than the number of blocks. For instance, MKC, composed of three single classifiers, is more robust than SMKC, which is constituted of four classifiers.

This superiority endorses the results mentioned later. As you can see in Table (5) and Table (Freund.Y and Schapire.E et al.), the best multiples, whether based on accuracy rate or robustness, have SVM as a building block. However, when considering time as another criterion, ensembles outperform even multiples because they are much faster than any combination involving SVM. The outcome of this investigation is significant for credit risk scoring.

For any other field, depending on the importance of the problem, time should be invested to find the best solution and choose the most accurate classifier. When the problem is critical, the time expended to find the best solution is indeed worthwhile.

REFERENCES

[1] Ali, K. M., & Pazzani, M. J. (1996). Error reduction through learning multiple descriptions. Mach. Learn., 24(3), 173-202. doi: 10.1023/a:1018249309965

[2] Baesens, B., Gestel, T.V., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J. (2003a). Benchmarking state-of-the-art classification algorithms for credit scoring. . Journal of the Operational Research Society, 54, 627–635.

[3] Bekhet, H. A., & Eletter, S. F. K. (2014). Credit risk assessment model for Jordanian commercial banks: Neural scoring approach. Review of Development Finance, 4(1), 20-28. doi: http://dx.doi.org/10.1016/j.rdf.2014.03.002

[4] Bian, S., & Wang, W. (2007). On diversity and accuracy of homogeneous and heterogeneous ensembles. Int. J. Hybrid Intell. Syst., 4(2), 103-128.

[5] C. Cortes, V. V. (1995). Support vector networks. Machine Learning 20 273–297.

[6] Crook, J. N., Edelman, D.B., Thomas, L.C. (2007). Recent developments in consumer credit risk assessment. . European Journal of Operational Research 183, 1447–1465.

[7] Dietterich, T. (2000). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. Machine Learning, 40(2), 139-157. doi: 10.1023/A:1007607513941

[8] Durand, D. (1941). Risk elements in consumer instalment financing. National Bureau of Economic Research, New York.

[9] Finlay, S. (2011). Multiple classifier architectures and their application to credit risk assessment. European Journal of Operational Research, 210(2), 368-378. doi: http://dx.doi.org/10.1016/j.ejor.2010.09.029

[10] Freund, Y. (2009). A more robust boostingalgorithm. . arXiv:0905.2138v1.

[11] Freund.Y and Schapire.E, pages, 148ñ156, & 1996. (1996). Experiments with a new boosting algorithm. . Machine Learning: Proceedings of the Thirteenth International Conference, 148_156.

[12] Hand, D. J., Henley, W.E. (1997). Statistical classification methods in consumer credit scoring: a review. Journal of the Royal Statistical Society, Series A-Statistics in Society 160, 523–541.

[13] Harris, T. (2013). Quantitative credit risk assessment using support vector machines: Broad versus Narrow default definitions. Expert Syst. Appl., 40(11), 4404-4413. doi: 10.1016/j.eswa.2013.01.044

[14] Harris, T. (2015). Credit scoring using the clustered support vector machine. Expert Syst. Appl., 42(2), 741-750. doi: 10.1016/j.eswa.2014.08.029

[15] Hens, A. B., & Tiwari, M. K. (2012). Computational time reduction for credit scoring: An integrated approach based on support vector machine and stratified sampling method. Expert Syst. Appl., 39(8), 6774-6781. doi: 10.1016/j.eswa.2011.12.057

[16] Huang, C.-l., Chen, M.-c., & Wang, C.-j. (2007). Credit scoring with a data mining approach based on support vector machines. Expert Systems with Applications, 33(4), 847-856. doi: 10.1016/j.eswa.2006.07.007

[17] Huang, Z., Chen, H., Hsu, C.-J., Chen, W.-H., & Wu, S. (2004). Credit rating analysis with support vector machines and neural networks: a market comparative study. Decis. Support Syst., 37(4), 543-558. doi: 10.1016/s0167-9236(03)00086-1

[18] Huysmans, J., Baesens, B., Vanthienen, J. (2005). A Comprehensible SOM-Based Scoring System. . Machine Learning and Data Mining in Pattern Recognition, 3587, 80–89.

[19] Kao, L.-J., Chiu, C.-C., & Chiu, F.-Y. (2012). A Bayesian latent variable model with classification and regression tree approach for behavior and credit scoring. Knowledge-Based Systems, 36(0), 245-252. doi: http://dx.doi.org/10.1016/j.knosys.2012.07.004

[20] Lee, T.-S., Chiu, C.-C., Chou, Y.-C., & Lu, C.-J. (2006). Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. Computational Statistics & Data Analysis, 50(4), 1113-1130. doi: http://dx.doi.org/10.1016/j.csda.2004.11.006

[21] Lee, T. S., & Chen, I. F. . (2005). A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. . Expert Systems with Applications, 28, 743-752.

[22] Maclin, R., & Opitz, D. (1997). An empirical evaluation of bagging and boosting. Paper presented at the Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence, Providence, Rhode Island.

[23] Maktabi, H., Tohidi, H. & Seyedaliakbar, S.M. (2011). An Application of Linear Programming for Efficient Resource Allocation Case Study of University Education. Australian Journal of Basic and Applied Sciences, 5(12): 703-706

[24] Mar, #237, Cubiles-De-La-Vega, a.-D., Blanco-Oliver, A., Pino-Mej, R., #237, . . . Lara-Rubio, J. (2013). Improving the management of microfinance institutions by using credit scoring models based on Statistical Learning techniques. Expert Syst. Appl., 40(17), 6910-6917. doi: 10.1016/j.eswa.2013.06.031

[25] Marqu, A. I., #233, Garc, V., #237, S, J. S., #225, & nchez. (2012). Two-level classifier ensembles for credit risk assessment. Expert Syst. Appl., 39(12), 10916-10922. doi: 10.1016/j.eswa.2012.03.033

[26] Martínez Sánchez, J. F., & Pérez Lechuga, G. (2016). Assessment of a credit scoring system for popular bank savings and credit. Contaduría y Administración, 61(2), 391-417. doi: http://dx.doi.org/10.1016/j.cya.2015.11.004

[27] McLachlan, G. J. (2004). , Discriminant Analysis and Statistical Pattern Recognition. John Wiley and Sons.

[28] Myers, J. H., Forgy, E.W. (1963). The development of numerical credit evaluation systems. . Journal of the American Statistical Association 50, 799–806.

[29] P. Martens , S. A., H. Maud and R. Mohsin (2010). Is globalization healthy: a statistical indicator analysis of the impacts of globalization on health. Globalization and Health.

[30] Sáez, J. A., Galar, M., Luengo, J., & Herrera, F. (2013). Tackling the problem of classification with noisy data using Multiple Classifier Systems: Analysis of the performance and robustness. Information Sciences, 247, 1-20.
[31] Serrano-Cinca, C., Bego, #241, Guti, A., #233, & Rrez-Nieto. (2013). Partial Least Square Discriminant Analysis for bankruptcy prediction. Decis. Support Syst., 54(3), 1245-1255. doi: 10.1016/j.dss.2012.11.015
[32] Tohidi, H., Jabbari, M.M., (2012). "Decision role in management to increase effectiveness of an organization". Procedia-social and behavioral sciences, 32: 825-828.
[33] Twala, B. (2010). Multiple classifier application to credit risk assessment. Expert Syst. Appl., 37(4), 3326-3336. doi: 10.1016/j.eswa.2009.10.018
[34] Wang, G., Hao, J., Ma, J., & Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. Expert Syst. Appl., 38(1), 223-230. doi: 10.1016/j.eswa.2010.06.048
[35] West, D. (2000). Neural network credit scoring models. Comput. Oper. Res., 27(11-12), 1131-1152. doi: 10.1016/s0305-0548(99)00149-5
[36] West, D., Dellana, S., & Qian, J. (2005). Neural network ensemble strategies for financial decision applications. Comput. Oper. Res., 32(10), 2543-2559. doi: 10.1016/j.cor.2004.03.017
[37] Wu, Z. (2007). Introduction to Network Analysis of Microwave Circuits Software VNA and Microwave Network Design and Characterisation (pp. 1-37): John Wiley & Sons, Ltd.
[38] Ziegler, J. K. n. A. A. S. n. A. G. A. n. A. A. (2013). Consumer credit risk: Individual probability estimates using machine learning. Expert Systems with Applications.