Credit Risk Assessment Using Learning Algorithms for Feature Selection

Zeinab Hassani a, Mohsen Alambardar Meybodi b and Vahid Hajihashemi c

a Department of Computer Science, Koshar University of Bojnord, Bojnord, Iran; b Department of Applied Mathematics and Computer Science, University of Isfahan, Isfahan, Iran; c Student Member, IEEE, Tehran, Iran

ABSTRACT

Firefly algorithm is one of the latest outstanding bio-inspired algorithms, which could be manipulated in solving continuous or discrete optimisation problems. In this context, we have utilised the firefly algorithm accompanied by five well-known models of feature selection classifiers to have an accurate estimation of risk, and further to improve the interpret-ability of credit card prediction. One of the significant challenges in the real-world datasets is how to select features. As most of the datasets are unbalanced, the selection of features turns to the maximum class of data that is not fair. To overcome this issue, we have balanced the data using the SMOTE method. Our experimental results on four datasets show that balancing data has increased accuracy. In addition, using a hybrid firefly algorithm, the optimal combination of features that predicts the target class label is achieved. The selected features by the proposed method besides being reduced can represent both majority and minority classes.

1. Introduction

Over the last several decades, increasing financial transaction leads to a tendency towards investigation on financial risk, particularly studies that evaluate the risk of customer credit transactions in commercial banks. Studies have shown that the majority of a bank’s loss risk originates from the customer credit risk, which is one of the most rapidly expanding issues in the banking industry. Hence, the banks need to employ an evaluating system to measure customers’ credit risk. Finding new ways of doing a less risky business with an entirely efficient and profitable feature is a permanent challenge in the banks. In retail, risks are often taken with no exact estimate of their degree and conceivable results.

There exist fundamental decisions in the process of loan to a consumer in financial institutions that essentially needs to rely on different algorithmic results rather than human decisions. For instance, in the process of loan approval in many banks, artificial intelligence models play the primary role in decision-making. Thereby, a low credit risk assessment model, which contains error misclassification costs, could lead to non-optimal capital allocation [1,2].
Machine learning techniques have been broadly utilised to improve precision in lots of real-world problems. Evaluation of Credit risk in the financial industry is one of these problems that has been investigated by these techniques in recent years. Most of them have been focused on the methods that incorporate the results of various classifier algorithms to obtain the optimal results [2]. Two different entities, both dataset, and algorithm, have a vital role in improving performance. The utilised databases for credit risk assessments, commonly have high dimensional data with the anomaly. The irrelevant features lead to less precision in the training process’s classification data. Feature selection, the process of selecting the essential features, is required to provide outperformed predictive accuracy and increase speed and scalability. Feature selection techniques are roughly applied in the preprocessing phase of classification [1,3]. Classifying objects is the heart of many kinds of research in pattern recognition, artificial intelligence, vision analysis, and medicine. In this regards, many studies have been investigated on the most important feature selection methods such as Support vector machine (SVM) and random forest [4], K nearest neighbour (KNN) [5,6], fuzzy KNN algorithm [7,8], decision Tree [4,5]. The performance of classifiers is usually estimated according to the predictive accuracy of constructed models. However, numerous real-world data, including financial data, is unbalanced. The machine learning algorithms with unbalanced data are can not well classify, so sometimes submits minority class data to the majority class by mistake. In other words, standard classification algorithms on unbalanced datasets suffer a loss of performance.

Over-sampling and under-sampling are two well-known methods for classifying unbalanced datasets by changing the distribution of data. Over-sampling by omitting the majority class instances and under-sampling by increasing the minority class, alter data distribution. However, the danger of eliminating some potentially useful data, or in case of over-sampling, incrementing the probability of over-fitting may occur in these methods. Synthetic Minority Oversampling Technique (SMOTE), is a well-known technique that adds new samples to the minority class [9].

In other side, Selecting a suitable set of features is a challenge in credit risk assessment algorithms. To measure selected features’ performance, various benchmark classifiers are used [1]. Piramuthu, in [10], bring up decision support applications for evaluating credit risk by a machine learning outlook. A hybrid algorithm composed of genetic algorithm and neural networks (HGA-NN) is proposed in [1]. It recognises a suitable subset of features and leads to improved accuracy classification and scalability in the evaluation of credit risk. The efficiency of the proposed method is measured by applying two real-known credit datasets, the Croatian bank dataset and the German credit dataset, which are chosen of the UCI databases. The results show 78.9% accuracy for German credit dataset, and average prediction precision of the HGA-NN method for the Croatian dataset is 82.88%. In 2007, Huang et al. considered a hybrid algorithm, combined SVM and GA, for feature selection of German Credit Card and tested its efficiency against the neural network, genetic programming, and C4.5. They showed that SVM is an outperformed rather than the other methods [11]. A compression between SVM ensembles with German Credit Data and neural networks was studied in [12]. Also, it was shown the performance of the SVM bagged ensemble is better than other methods. Khashman et al. [13] introduced a new evaluation system for estimating credit risk that employs supervised neural network methods. It obtained performance best with 74.46 accuracies. A study of a new classification technique based on genetic algorithm and neural network on credit risk assessment has been done in [14]. The
proposed algorithm performed on the Croatian and the German retail credit datasets. Xue et al. in [3] based on the RIPPER algorithm, proposed a hybrid model. In their algorithm, a new strategy for data pretreatment is proposed, and next, for feature selection, eliminate the redundant features. Then, to overcome the obstacle of the unbalanced distribution of credit card datasets, a sampling algorithm for minority class is used to normalise the samples. Finally, taking benefits of the rules produced by the RIPPER algorithm, default credit card users are predicted by.

Our contribution: A comparative analysis of different feature selection models with the motivation to improve scalability and precision of the algorithms in evaluating retail credit risks has been investigated. The efficiency of the proposed algorithms is evaluated using a real-world German credit dataset downloaded from the UCI database. As this financial data is unbalanced, firstly, we implement the SMOTE method to balance the data. Then, the striking features (features selection) of the credit card have been investigated using the firefly binary algorithm and various classifiers KNN, Fuzzy KNN, Random Forest, Decision Tree, and SVM. Our results are differentiated from [11–13] in significant ways and indicate that the proposed method attains acceptable results.

The current article is organised into five sections. In Section 2, a unified description of the used tools is given. Afterwards, the proposed method in detail discusses in Section 3. The utilised parameters and criteria in our model accompanied by results are given in Section 4. We consider them to analyse the performance of proposed algorithms in providing a loan to a company. We also analyse in detail the impact of feature selection retained by each classifier on the final results. In Section 5, Experimental results and improvements achieved by each classifier are discussed, and at last, it reveals the study’s conclusion.

2. Preliminaries

In this section, we briefly review fundamental symbols for more description of the methods KNN, Fuzzy KNN, Random Forest, Decision Tree, SVM, and Firefly algorithm that is implemented in other sections.

2.1. K-Nearest Neighbours

K-Nearest Neighbours (KNN) method is a simple, most popular and highly efficient algorithm based on a similarity measure, for instance, the distance function. In the learning machine perspective, KNN is a non-parametric and lazy classification method [15]. It is known as a non-parametric technique which means it does not make any presumption on the underlying data distribution. It is known as a lazy algorithm since it does not utilise the training data points to perform any generalisation. Mainly the purpose of KNN is separating the data points into several classes and then predict to which class a new sample point belongs. It is wholly based on feature similarity of the majority vote of neighbours. A given data point is assigned to that class, which has most common among its k nearest neighbours [16].

The KNN algorithm can be summarised as:

1. Input is a new sample along with a positive integer k,
2. For the given sample, select the k closest entry in the database,
(3) Determine the most common classification of these entries,
(4) The output is to which class the new sample belongs [17].

In other words, consider \( X = \{x_1, x_2, \ldots, x_n\} \) be a training set, contain \( N \) instances which belong to the \( C \) classes. Let \( Q \) be the set of test samples. For a test instance \( y \in Q \), KNN assigns classifier to \( y \) with the weighted Euclidean distance of \( y \) and \( X \) as follows:

\[
c(v) = \text{Argmax}_{r=1,2,\ldots,C} \{ |\{ x_i : x_i \in \Omega_k(X, y) \cap c(x_i) = r \}| \}
\]

(1)

Where \( \Omega_k(X, y) \) denote the \( k \)-neighbourhoods of a point \( y \) in the set \( X \). Hence, the KNN classifier assigns a class label to the instance \( y \), the class in which the majority of the \( k \) neighbours of \( y \) in training set belongs to. However, the importance of increasing the distance of a neighbour and the test point is not considered by KNN. Besides, KNN provides only one single class for each training point and, assigns a unique class label for each test point. Fuzzy KNN will handle these issues.

### 2.2. Fuzzy K Nearest Neighbours

Fuzzy K-Nearest Neighbour algorithm (FKNN), presented in [8], preserves the fundamental decision rule of the KNN method and attempts to overcome both of KNN defects that mentioned in the preceding section. The comparison between FKNN and other classifiers such as crisp KNN, neural networks, Bayesian, and linear discriminant functions illustrate this method’s superiority [18]. Consider \( X = \{x_0, x_1, \ldots, x_N\} \) be a training set consist of \( N \) instances which distributed into \( C \) classes and \( Q \) be the set of test patterns. Shortly, FKNN assigns a probability, called membership \( \mu(\cdot) \), to each of the training points \( y \), in other words, \( \mu_i(y) \) is the membership of the data point \( y \) to class \( i \). The classifier assigns membership to a data point \( y \) according to the following equation.

\[
\mu_i(y) = \begin{cases} 
    \frac{nn_c}{k_{in}} \times 0.49 + 0.51 & \text{if } c = w \\
    0.49 \times \frac{nn_c}{k_{in}} & \text{Otherwise} 
\end{cases}
\]

(2)

Where \( k_{in} \) is the \( i \)th nearest neighbour and \( m \) greater than 1 that specified the strength of the fuzzy distance function. \( k \)-th nearest neighbour vector \( x_j \), denoted as \( K_j \) and \( ||Q - k_j|| \) is the L-norm distance between the set \( Q \) and \( K_j \) [18].

### 2.3. Decision Tree

One of the most widely used classification techniques is decision tree. A decision tree uses a recursive partition method in training instances. It constructs a tree-like graph with a top-down manner in every iteration to partition the instances according the best features [19].
Decision trees are usually used in operational research, particularly in decision analysis, in order to help in the process of identifying a structure to achieving a special goal and also as a well-known tool in machine learning. A decision tree is a flowchart structure in which each inner node that tried to represent a ‘test’ on a feature. Each branch results from the experiment and in the class label for each node (decision after calculating all attributes). The rules of classification are derived from paths of root to leaf. In all sections of decision analysis, a decision tree and the related influence diagram are used as to appear to be a logical decision and intuitive support tool, where the all of the expected values (or expected instruments) of competing alternatives are computed [20].

2.4. Random Forest

Random Forest is a machine learning algorithm that produces reliable and outstanding results even without setting a specific parameter, customarily and objectivity manner that can be utilised for classification and regression functions. The random decision is a totality learning technique for classification, regression, and other functions manipulate by constructing for plenty of decision trees. This procedure applies to individual trees of the classes (classification) or means prediction (regression) [21]. Although known as the unrivalled algorithm in terms of accuracy, this method efficiently used in large data sets with thousands of entrance without over-fitting and any requirement for data pruning is one of the perfect properties of random Forest [22]. Tin Kam Ho [23] consumes the first idea of implementing the random method to achieve for the fulfilment of the ‘accidental difference’ towards propositional classification proposed by Eugene Kleinberg [24–26]. Other extensions of this idea have been explained by Leo Breiman [27]. The deployment Breiman’s ‘bagging’ concept and combined with a random selection of based on features, introduced first by Ho [23] and later completely independent by Amit and Geman [28] to build a set of decision trees with controlled variance. Technically, the idea behind Random forest is to build an ensemble of learners, derived from the original dataset, after training on a bootstrap sample, denoted by Db, using the following sampling method: Let D be the original dataset with N examples, and the Db be a set created by randomly selecting k samples from set D with replacement. After eliminating duplicates, if N is large enough and k = N, the expectedly set Db involves approximately two-thirds of samples from D. The ensemble’s prediction is emerged from the separate decisions according to majority voting for classification process or averaging(regression). The variance in the final model can be reduced by comparing it to the base models, which can lead to avoiding over-fitting [29].

2.5. Support Vector Machine

A Support Vector Machine (SVM) identifies as a discriminative classifier, regression analysis, and supervised learning models with associated learning algorithms in a large area of real-world problems [30]. Assume X be a training set, consisted of N instances which belong to two classes denoted by \( I_1 \) and \( I_2 \). For \( i = 1, \ldots, N \) define \((x_i, y_i)\) where \( x_i \) is an object with d features and \( y_i \) is its label defines as follows: if instance \( x_i \) belongs to \( I_1 \), the value of \( y_i \) equals 1 and \( y_i \) equals \(-1\) if \( x_i \) belongs to class \( I_2 \). Method SVM implements a linear separating
hyperplane to classify the object $x_i$ correctly as follows:

$$z(x) = \text{sgn} \left( \sum_{i=1}^{N} a_i y_i x_i^T x + b \right)$$

(4)

Where the parameter $(a_i \geq 0)$ is Lagrange multiplier and according to the input vector $x_i$, the constant $b$ determines the optimal separating hyperplane. For the non-linearly disjoint case, the classifier is modified as:

$$z(x) = \text{sgn} \left( \sum_{i=1}^{N} a_i y_i k(x_i, x) + b \right)$$

(5)

Where $K(x_i, x)$ is the kernel function, which is a non-linear mapping between input space and one (high-dimensional) of the property space. There are well-known kernel functions that create the decision rules such as radial basis function (RBF) kernel, the linear kernel, the polynomial kernel, and the multilayer perceptron (MLP) kernel. Choosing suitable values for parameters plays a vital role in constructing the classification model with prediction stability and precision. Although there are no general rules that guarantee the best value of the parameters for the program problems [31].

2.6. Feature Selection

In machine learning, algorithms can perform particular classification based on a set of features. Moreover, the scope of features involves hundreds of variables or features used in machine learning applications or pattern recognition. Several methods are extended to address the problem of reducing redundant and irrelevant redundant variables on challenging tasks [32]. Variable Selection (Feature Selection) leads to improvements in data comprehensibility, data visualisation, and prediction performance. It also reduces the training time of learning algorithm, computation requirement, and the effect of the curse of dimensionality [17,32]. Filter, wrapper, and embedded are three main categories that feature selection is broadly classified into them. Filter methods are among the eldest methods of feature selection that work without implementing any learning algorithm. Filter methods as a preprocessing tool utilise in rating the features. In this algorithm, features will be ranked based on a particular evaluation parameter and selected based on four criteria distance, dependency, consistency, and information. The wrapper method finds the subset of features based on the classifier that the feature selection criterion is the performance of the predictor. The subset of features with the highest predictor performance is selected. Embedded methods combine the benefits of both Wrapper models and filter methods using taking different rating parameters in various steps of the search. This model includes feature selection during the training process without splitting the data into training, and testing sets [32,33]. This Embedded idea is similar to wrappers but less prone to overfitting and less computationally expensive. Moreover, the significant restriction with this method is that it takes decisions based on the classifiers [17,33].
2.7. Synthetic Minority Over-sampling Technique

SMOTE, Synthetic Minority Over-sampling Technique, was proposed by Chawla et al., performs a screening method to balance the original unbalanced training set. The simplest algorithm applies the repetition of the minority class, but the main idea of SMOTE is to introduce artificial prototypes. SMOTE mentions that points between two points close to the negative class are assigned to the negative class and tries to synthesise new negative instances among negative grade examples [9]. The new synthetic negative grade combination in SMOTE method is created as follows: Let m be the sample rate. For each negative class instance $x_i$, it obtains $k$ nearest neighbours. Then randomly selects $m$ nearest neighbour $y_{ij}$, where $j = 1, 2, \ldots, m$ and eventually, new point $p_j$ where $j = 1, 2, \ldots, m$ obtains as:

$$p_j = x_i + r \times (y_{ij} - x_i)$$ (6)

Where $r$ and $(0, 1)$ is a random number in interval $(0, 1)$ [33,34]. The technique re-sampled training set, synthesising the new synthetic negative class instances with the original dataset leads to significantly improvement in the degree of imbalance data in the original dataset [3].

2.8. Firefly Algorithm

The firefly algorithm is classified as swarming intelligent algorithm with the most effective performance that has increasingly applied in solving optimisation problems based on flashing phenomena of fireflies [34]. The firefly algorithm (FA), proposed by Yang [35], is inspired by the ability to produce flashing light by fireflies. The main tasks of them can be expressed briefly in two steps: In the first step, they attract mating partners and in the second for a predominant predator of reminded the bitter taste of firefly [36]. To summarise the FA steps, we can introduce it through three phases [37]:

Initialise: A lightning signal is assigned to each firefly according to the two fireflies’ distance. The coefficient of atmospheric absorption obtains as below:

$$I(r) = I_0 e^{-\gamma r^2}$$ (7)

Where $I$ is the strength of the light source, $r$ is the distance between the two considered fireflies, $\gamma$ is the absorption coefficient, and $I_0$ is the strength of the light source when $r = 0$.

Attractiveness: it can be expressed by:

$$B(r) = B_0 e^{-\gamma r^2}$$ (8)

Where $B_0$ is the attractiveness of the firefly when $r = 0$.

The moving step: In each pair of fireflies for the whole population, the less fit Firefly is going to the cost-efficient ones, using the following model:

$$x_{i}^{t+1} = x_{i}^{t} + B_0 e^{-\gamma r^2} (x_{i}^{t} - x_{j}^{t}) + a \times r \text{ and } n(-0.5, 0.5)$$ (9)

Where $\alpha$ is the mutation coefficient often makes a self-adaptive parameter that can decrease through each iteration and $r$ and $n(-0.5, 0.5)$ is a normal randomised number range $[-0.5, 0.5]$. A pseudo-code of the FA is illustrated in Algorithm 1, from which it can be seen that the algorithm consists of the above steps:
**Algorithm 1** Firefly algorithm.

**Input:** Population of $N$ fireflies and objective function

**Output:** The best solution and value of objective function of best solution. generate an initial population of $N$ fireflies, $t = 0$, Number of iterations for optimisation $MaxIteration$. Define light absorption coefficient $\gamma$, $\beta_0$ and $I_0$

1. while ($t < MaxIteration$) do
2.   for $i = 1$ to $N$ do
3.     for $j = 1$ to $N$ do
4.       if fitness($x_i$) > fitness($x_j$) then
5.         move firefly towards firefly according Equations (9)
6.       Vary attractiveness with distance $r$ using Equations (8)
7.       Evaluate new solutions and update light intensity.
8.       end if
9.     end for
10.   end for
11. ranks the fireflies and finds the current global best $g$.
12.   $t = t + 1$;
13. end while
14. Post Process results and visualisation

In the next section, we consider methods KNN, FKNN, Random forest, Decision tree, and SVM as the different objective functions.

### 3. The Proposed Method

In this section, our primary goal is to compare the classification results with the hybrid algorithm in the retail credit risk evaluation domain. The hybrid algorithm is considered a firefly algorithm with five classification methods: KNN, FKNN, Random forest, Decision tree, and SVM. Classification performances are measured by different efficiency measurement tools and hybrid algorithms focusing on the accuracy of classification. First, the dataset is preprocessed. Then the dataset of preprocessing is implemented the proposed algorithm. The procedures for in the proposed method for unbalanced data are as follows:

**Step 1:** Dataset is preprocessed that includes a normalise and balanced dataset. Some values of features are in a different range, and better results depend on the appropriate data in the algorithm. Therefore, dataset is normalised in interval $[0, 1]$. The normalisation formula is as follows:

$$X = \frac{x - x_{\text{MIN}}}{x_{\text{MAX}} - x_{\text{MIN}}}$$  \hspace{1cm} (10)

Where $x_{\text{MIN}}$ and $x_{\text{MAX}}$ are the minimum and maximum values of each feature. Then data balanced by the Smote method. Also, the performance of the classifier algorithm is improved by the normalised dataset.
Table 1. Descriptive information for all datasets.

| Num | Name                        | Instance | Features |
|-----|-----------------------------|----------|----------|
| 1   | German                      | 1000     | 24       |
| 2   | South German                | 1000     | 21       |
| 3   | credit card econometrics    | 1320     | 12       |
| 4   | Thomas                      | 1225     | 15       |

Table 2. Features of German and South German datasets.

| Num | German                                      | South German                   |
|-----|---------------------------------------------|--------------------------------|
| 1   | Status of existing checking account         | status of the debtor’s checking account with the bank |
| 2   | Duration in month                           | credit duration in months      |
| 3   | Credit history                              | history of compliance with previous or concurrent credit contracts |
| 4   | Purpose                                     | purpose for which the credit is needed |
| 5   | Credit amount                              | credit amount in DM            |
| 6   | Savings account/bonds                       | debtor’s savings               |
| 7   | Present employment since                    | duration of debtor’s employment with current employer |
| 8   | Installment rate in percentage of disposable income | credit installments as a percentage of debtor’s disposable income |
| 9   | Personal status and sex                     | combined information on sex and marital status |
| 10  | Other debtors / guarantors                  | Is there another debtor or a guarantor for the credit? |
| 11  | Present residence since                     | length of time (in years) the debtor lives in the present residence |
| 12  | Property                                    | the debtor’s most valuable property |
| 13  | Age in years                                | age in years                   |
| 14  | Other Installment plans                     | installment plans from providers other than the credit-giving bank |
| 15  | Housing                                     | type of housing the debtor lives in |
| 16  | Number of existing credits at this bank     | number of credits including the current one the debtor has (or had) at this bank |
| 17  | Job                                         | quality of debtor’s job        |
| 18  | Number of people being liable to provide maintenance for | number of persons who financially depend on the debtor |
| 19  | Telephone                                   | Is there a telephone landline registered on the debtor’s name? |
| 20  | foreign worker                              | Is the debtor a foreign worker? |
| 21  |                                             | Has the credit contract been complied with (good) or not (bad)? |

Step 2: Parameters algorithm is Initialised. Let $N$ be the number of Fireflies, Max-iteration be the number of iterations for optimisation, $\gamma$ be absorption coefficient parameter, $\alpha$ be attractiveness of the firefly, $\alpha$ be mutation coefficient and $f(x_i)$ be objective function.

Step 3: Initialise a population of $n$ fireflies’ positions at random and $t = 0$.

Step 4: The value of the fitness function of fireflies is evaluated. This step implements five classifier algorithm. The performance of the classifier algorithm is selected for the value fitness function. The performance of a classifier is accurate.

Step 5: Update the position of fireflies by comparing their fitness and then evaluating new solutions and updating the light intensity of fireflies as follows: Let $(x_i, x_j)$ be a pair of fireflies. Then if fitness$(x_i)$ is greater than fitness$(x_j)$, firefly moves towards firefly according to Equation (9). Otherwise, attractiveness with distance $r$ is calculated using Equation (8) and evaluate new solutions and update light intensity by objective function.

Step 6: rank the fireflies, find the current global best and $t = t + 1$.

Step 7: If Satisfy the termination $t < \text{Max – iteration}$ continue step 5 until 6 else return optimal firefly position and its value fitness return optimal firefly position and its value fitness.
Table 3. Features of Thomas and credit card econometrics.

| Num | Thomas | Credit card econometrics |
|-----|--------|--------------------------|
| 1   | Year of birth | card |
| 2   | Number of other dependents | reports |
| 3   | Is there a home phone | age |
| 4   | Spouse's income | income |
| 5   | Applicant's employment status | share |
| 6   | Applicant's income | expenditure |
| 7   | Residential status | owner |
| 8   | Value of Home | self employ |
| 9   | Mortgage balance outstanding | dependents |
| 10  | Outgoings on mortgage or rent | months |
| 11  | Outgoings on Loans | major cards |
| 12  | Outgoings on Hire Purchase | active |
| 13  | Outgoings on credit cards | |
| 14  | Good/bad indicator | |

Table 4. Parameter setting for experiments.

| Parameter | Value | Meaning |
|-----------|-------|---------|
| Max_iteration | 50,100 | The Number of iterations for optimisation |
| n | 15,25 | Number of used search agents in the optimisation |
| $\alpha$ | 0.02 | mutation coefficient |
| $\gamma$ | 1 | Absorption coefficient |
| $\beta$ | 2 | attractiveness of the firefly |

4. Experimental Results

In this section, we explain the dataset and assessment method used in our experiments. Then the classification results and assessments of the proposed algorithm will be illustrated.

4.1. Description of the Datasets

The experiments are applied over four real benchmark datasets such as German Credit Data, South German Credit Dataset from UCI Repository of Machine Learning [38], credit card econometrics from Kaggle [39], and Thomas [40]. Table 1 shows description of datasets. For instance, the German Credit data sets involve 700 instances of worthy credit costumers and 300 instances of unreliable credit costumers. It consists of 24 regular features for each applicant that define some vital properties like personal information, credit background, account balances, loan aims, loan amounts, employment status, etc. The features of used data sets are shown in Tables 2 and 3.

4.2. Evaluation Method

In the interest of evaluating the efficacy of the proposed model that is computed for comparisons, we use accuracy, precision, recall, and F-score. In the flowing, we define evaluation
Table 5. The result of five classifiers on unbalanced data sets.

| Functions        | KNN                | FKNN               | Random forest     | Dtree             | SVM               |
|------------------|-------------------|--------------------|-------------------|------------------|------------------|
| Dataset          | Accuracy          | Precision          | Recall            | Fscore           | Accuracy          | Precision          | Recall            | Fscore           | Accuracy          | Precision          | Recall            | Fscore           |
| German           | 70.80             | 84.40              | 76.42             | 80.17            | 71.61            | 82.74             | 76.67             | 91.46            | 78.72             | 84.58             | 74.42             | 79.55             | 78.52             | 78.93             | 75.50             | 92.80             | 77.02             | 84.13             |
| South German     | 73.30             | 87.03              | 77.61             | 82.01            | 73.12            | 83.21             | 76.15             | 90.10            | 78.87             | 84.10             | 70.78             | 79.49             | 78.96             | 79.07             | 76.03             | 91.93             | 77.89             | 84.29             |
| Thomas credit-card-econometrics | 68.96             | 85.92              | 75.37             | 80.27            | 74.10            | 84.71             | 73.28             | 92.47            | 76.29             | 83.59             | 65.89             | 77.99             | 76.26             | 77.01             | 73.74             | 96.59             | 74.98             | 84.41             |
| Thomas           | 85.66             | 94.04              | 88.28             | 91.06            | 80.61            | 99.45             | 80.26             | 88.81            | 98.02             | 98.71             | 99.00             | 99.03             | 98.11             | 98.06             | 87.26             | 96.66             | 88.15             | 92.18             |

Table 6. Result of hybrid firefly algorithm and different classification on German data set.

| Max iteration | Number of fireflies | KNN                | FKNN               | Random forest     | Dtree             | SVM               |
|---------------|---------------------|--------------------|--------------------|-------------------|------------------|------------------|
| Number of fireflies | Accuracy          | Precision          | Recall            | Fscore           | Accuracy          | Precision          | Recall            | Fscore           | Accuracy          | Precision          | Recall            | Fscore           |
| 50            | 15                  | 81                 | 83.43              | 79.75             | 81.47            | 84.64             | 91.91             | 79.62             | 85.332            | 85.36             | 82.14             | 87.98             | 84.86             | 81.21             | 80.50             | 81.71             | 81.14             | 88.14             | 84.9             | 83.97             |
| 25            | 80.21               | 83.86              | 77.18             | 81.24             | 85.36            | 89.44             | 83.01             | 86.10             | 85.21             | 82.86             | 86.96             | 84.84             | 81.29             | 76.76             | 84.31             | 80.21             | 85.87             | 82.74             | 85.51             |
| 100           | 15                  | 80.64              | 86.29              | 77.61             | 81.65            | 85.71             | 92.57             | 82.53             | 87.26             | 84.71             | 82.43             | 86.37             | 84.35             | 80.93             | 78.33             | 82.71             | 80.25             | 82.36             | 82.29             | 84.24             |
| 25            | 83.21               | 90.71              | 79.02             | 84.40             | 87.14            | 89.18             | 88.05             | 88.61             | 85.64             | 82.86             | 87.77             | 85.24             | 80.93             | 79.78             | 81.70             | 80.60             | 85.36             | 86.57             | 84.72             | 85.55             |
| Features selected | 13 features: 1, 2, 3, 4, 5, 6 | 7 features: 1, 2 | 15 features: 1, 2, 3, 4, 6, 9, 10 | 8 features: 1, 2, 3, 6, 7 | 13 features: 1, 2, 3, 4, 5, 6, 8, 11, 12, 16, 19, 21, 23 |

Table 7. Result of hybrid firefly algorithm and different classification on South German data set.

| Max iteration | Number of fireflies | KNN                | FKNN               | Random forest     | Dtree             | SVM               |
|---------------|---------------------|--------------------|--------------------|-------------------|------------------|------------------|
| Number of fireflies | Accuracy          | Precision          | Recall            | Fscore           | Accuracy          | Precision          | Recall            | Fscore           | Accuracy          | Precision          | Recall            | Fscore           |
| 50            | 15                  | 80.79              | 71.29              | 88.19             | 78.71            | 84.64             | 77.70             | 92                | 84.25             | 84.86             | 83.71             | 85.75             | 84.63             | 80.29             | 82.54             | 78.99             | 80.65             | 83.43             | 83.57             | 83.40             | 83.42             |
| 25            | 81.36               | 72.71              | 88.02              | 79.56             | 85.71            | 79.23             | 88.79             | 83.73             | 86.21             | 87.00             | 86.77             | 85.31             | 79.79             | 83.01             | 78.15             | 80.23             | 84.78             | 82.14             | 86.82             | 84.38             |
| 100           | 15                  | 81.14              | 72.71              | 87.83             | 79.35             | 83.93            | 77.14             | 89.26             | 82.76             | 84.14             | 84.86             | 85.42             | 85.07             | 78.50             | 79.89             | 77.79             | 78.66             | 84.14             | 82.29             | 85.65             | 83.80             |
| 25            | 79.86               | 81.94              | 78.55              | 80.03             | 85.36            | 81.29             | 91.30             | 86.01             | 84.21             | 84.86             | 83.90             | 84.33             | 79.86             | 81.94             | 78.55             | 80.03             | 85.21             | 84.57             | 86.13             | 85.17             |
| Features selected | 7 features: 1, 3, 4, 10, 11, 12, 14 | 7 features: 1, 2, 3, 9, 10, 12, 14 | 6 features: 1, 2, 3, 5, 6, 7 | 6 features: 1, 4, 7, 11, 12, 14 | 7 features: 1, 3, 4, 5, 6, 8, 10, 11, 12, 14 |

FUZZY INFORMATION AND ENGINEERING 539
Table 8. Result of hybrid firefly algorithm and different classification on Thomas data set.

| Max iteration | number of fireflies | KNN | FKNN | Random forest | Dtree | SVM |
|---------------|---------------------|-----|------|---------------|-------|-----|
|               |                     | Accuracy | Precision | Recall | Fscore | Accuracy | Precision | Recall | Fscore | Accuracy | Precision | Recall | Fscore | Accuracy | Precision | Recall | Fscore | Accuracy | Precision | Recall | Fscore | Accuracy | Precision | Recall | Fscore | Accuracy | Precision | Recall | Fscore |
| 50            | 15                  | 75.33 | 68.63 | 79.43 | 73.53 | 78.06 | 74.87 | 78 | 82.48 | 83.38 | 79.43 | 73.53 | 78.06 | 74.87 | 78 | 82.48 | 83.38 | 79.43 | 73.53 | 78.06 | 74.87 | 78 | 82.48 | 83.38 | 79.43 | 73.53 | 78.06 | 74.87 | 78 |
|               | 25                  | 74.33 | 66.19 | 79.05 | 71.94 | 79.44 | 73.63 | 83.75 | 78.37 | 82.04 | 81.49 | 82.46 | 81.92 | 74.67 | 76.25 | 74.06 | 74.99 | 79.94 | 80.27 | 79.86 | 80.03 |
|               | 100                 | 75.22 | 65.86 | 81.13 | 72.60 | 79.17 | 75.15 | 80.25 | 77.61 | 83.04 | 83.71 | 82.69 | 83.14 | 73.67 | 75.49 | 72.96 | 74.08 | 80.22 | 80.49 | 80.19 | 80.27 |
|               | 25                  | 77.05 | 69.19 | 85.45 | 75.09 | 79.44 | 73.84 | 81.41 | 77.44 | 82.76 | 83.49 | 82.33 | 82.88 | 75.22 | 77.59 | 74.48 | 75.82 | 80.66 | 76.61 | 83.38 | 79.81 |

Feature selected
4 features: 1, 4, 7, 9
4 feature: 1, 3, 7, 9
8 features: 1, 2, 6, 7, 8, 9, 10, 13
3 features: 1, 2, 6
5 features: 2, 4, 7, 8, 9

Table 9. Result of hybrid firefly algorithm and different classification on credit card econometrics data set.

| Max iteration | number of fireflies | KNN | FKNN | Random forest | Dtree | SVM |
|---------------|---------------------|-----|------|---------------|-------|-----|
|               |                     | Accuracy | Precision | Recall | Fscore | Accuracy | Precision | Recall | Fscore | Accuracy | Precision | Recall | Fscore | Accuracy | Precision | Recall | Fscore | Accuracy | Precision | Recall | Fscore | Accuracy | Precision | Recall | Fscore |
| 50            | 15                  | 98.88 | 97.76 | 100 | 98.86 | 94.13 | 90.91 | 97.44 | 94.06 | 99.02 | 98.04 | 100 | 99.01 | 98.92 | 99.70 | 98.92 | 98.92 | 98.5 | 98.91 | 77.37 | 76.87 |
|               | 25                  | 98.58 | 97.26 | 99.90 | 98.56 | 97.56 | 98.56 | 96.70 | 97.62 | 98.93 | 97.86 | 100 | 98.91 | 98.93 | 100 | 98.90 | 98.88 | 97.85 | 98.86 | 79.86 | 80.03 |
|               | 100                 | 98.49 | 97.17 | 99.80 | 98.45 | 95.11 | 95.98 | 94.09 | 95.02 | 99.02 | 98.05 | 100 | 99.01 | 98.93 | 100 | 98.89 | 98.97 | 97.95 | 98.96 | 80.19 | 80.27 |
|               | 25                  | 98.58 | 97.36 | 99.80 | 98.56 | 95.60 | 95.81 | 94.81 | 95.32 | 98.93 | 97.86 | 100 | 98.90 | 98.93 | 100 | 98.87 | 98.97 | 97.95 | 98.95 | 83.38 | 79.81 |

Features selected
2 features: 4, 6
4 features: 1, 4, 6, 7
4 features: 1, 5, 8, 9
3 features: 5, 6, 10
2 features: 4, 5
metrics:

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}
\]

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{12}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{13}
\]

\[
F\text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{14}
\]

Where TN is True Negatives, TP is True Positives, FN is False Negatives, and FP is False Positives [2,15].

### 4.3. Results and Discussion

The experiments were done on a desktop computer which has Intel Core i7-6700 @ 3.40 GHz processor, 32 GB memory, Windows10 OS configuration, and experiment software, all algorithms were coded by Matlab R2018a software. The global and optimiser-specific parameter setting is outlined in Table 4.

At first, for unbalanced data sets, results were obtained from 10 independent runs for five classifier algorithms. The results are depicted in Table 5.

Tables 6–9 outline the performance of the hybrid firefly algorithm and different classification on our datasets. In our approach, the binary firefly algorithm is used to select features of the credit datasets, and the classifier is performed with the diverse classifiers of machine learning. In each epoch, the maximum number of function evaluations was used from all data sets. It was 7500 for 15 dimensional problems (15D) with 50 iterations and 15,000 for 25 dimensional problems (25D) with 100 iterations.

The proposed method was considered using various classifiers to obtain the higher performance of the proposed method (feature selection with the optimised hybrid algorithm).

Table 6 shows the accuracy of the hybrid firefly algorithm and different classification on the German credit dataset. HFA-FKNN (hybrid firefly algorithm and FKNN classifier) leads to better accuracy than other classifiers. The best accuracy of all datasets are specified in

| Reference | Method       | Selected features | Accuracy |
|-----------|--------------|-------------------|----------|
| [11]      | SVM+GA       | Average (13.30)   | 77.92    |
|           | GP           | Average (13.20)   | 78.10    |
| [12]      | SVM          | All               | 76.59    |
| [13]      | NN           | All               | 74.67    |
| [1]       | HGA-NN       | Average (12.36)   | 78.90    |
| [14]      | HGA-NN       | All               | 78.6     |
| Proposed  | HFA_KNN      | 13                | 83.21    |
| algorithm | HFA_FKNN     | 7                 | 87.14    |
|           | HFA_RF       | 15                | 85.71    |
|           | HFA_Dtree    | 8                 | 81.29    |
|           | HFA_SVM      | 13                | 85       |

Table 10. The comparison between our results and other methods on German credit dataset.
Figure 1. The comparative bar graph of accuracy in the implementation of Firefly Algorithm on original datasets (light colours) and the proposed method (dark colours).

Bold in the tables. Accuracy of HFA-FKNN has resulted in 87.14 % and seven features are selected.

Comprehensive studies have been done on the German credit dataset with machine learning algorithms. Table 10 shows the comparison of the proposed approach with other studies in this area. In this study, the combined firefly algorithm has a higher accuracy machine learning algorithm, and the combined HFA—FKNN algorithm has a higher accuracy with fewer attributes. Compared to previous studies, the proposed method is superior.

5. Conclusion

In financial risk management, credit cards’ evaluation is a vital role for banks. Banks often need to evaluate customer credit to decide against risk and competition. Identify influential factors of the credit card can help them. Moreover, feature selection using the machine learning algorithms can evaluate them by higher accuracy while traditional techniques require essential assumptions. Real-world data such as financial data are unbalanced, and machine learning algorithms are not well classified. Therefore, at first in this paper, the data is balanced by the SMOTE method, and then the firefly binary algorithm, is used for identifies the effective features of the credit cards. Beside, this study investigates various machine algorithms such as KNN, Fuzzy KNN, Random Forest, Decision Tree, and SVM for the effective classifier of the credit card (Figure 1).
Disclosure statement
No potential conflict of interest was reported by the author(s).

Notes on contributors
Zeinab Hassani received the M.Sc. degree in computer science from the Institute for Advanced Stud-
yed in Basic Science (IASBS), Zanjan, Iran in 2011. She works as instructor at department basic science,
Kosar university of Bojnord. Her current research interests include the Artificial Intelligence, Neural
Network, Computational Geometry.

Mohsen Alambardar Meybodi received the MS and PhD degrees in computer science from Yazd Uni-
versity, Iran, in 2012 and 2017, respectively. He is an associate professor in the Department of Applied
Mathematics and Computer Science, University of Isfahan, Isfahan, Iran. His current research interests
include parameterized algorithms, graph algorithms, and machine learning.

Vahid Hajihashemi is currently a Ph.D. student at FEUP - Faculdade de Engenharia da Universidade do
Porto, Portugal. He has expertise and various publications in artificial intelligence, signal processing,
image processing and pattern recognition. He is a professional computer programmer with expertise
in computer science, stochastic process, mathematical programming and soft computing.

ORCID
Zeinab Hassani http://orcid.org/0000-0001-5516-4377
Mohsen Alambardar Meybodi http://orcid.org/0000-0003-2089-0250
Vahid Hajihashemi http://orcid.org/0000-0002-0842-8250

References
[1] Oreski S, Oreski G. Genetic algorithm-based heuristic for feature selection in credit risk assess-
ment. Expert Syst Appl. 2014.41(4):2052–2064.
[2] Xu P, Ding Z, Pan M. A hybrid interpretable credit card users default prediction model based on
ripper. Concurrency Comput Pract Exp. 2018;30(23):e4445.
[3] Xue W, Zhang J. Dealing with imbalanced dataset: a re-sampling method based on the improved
smote algorithm. Commun Statist Simul Comput. 2016;45(4):1160–1172.
[4] Butaru F, Chen Q, Clark B, et al. Risk and risk management in the credit card industry. J Bank
Finance. 2016.72:218–239.
[5] Galindo J, Tamayo P. Credit risk assessment using statistical and machine learning: basic methodo-
logy and risk modeling applications. Comput Econ. 2000.15(1-2):107–143.
[6] Huang Z, Chen H, Hsu CJ, et al. Credit rating analysis with support vector machines and neural
networks: a market comparative study. Decision Support Syst. 2004.37(4):543–558.
[7] Roh SB, Ajn TC, Pedrycz W. The design methodology of radial basis function neural networks
based on fuzzy k-nearest neighbors approach. Fuzzy Sets Syst. 2010;161(13):1803–1822.
[8] Keller JM, Gray MR, Givens JA, et al. A fuzzy k-nearest neighbor algorithm. IEEE Trans Syst Man
Cybern. 1985.4:580–585.
[9] Jedrzejowicz J, Kostrzewski R, Neumann J, et al. Imbalanced data classification using mapreduce
and relief. J Inf Telecomm. 2018.2(2):217–230.
[10] Piramuthu S. On preprocessing data for financial credit risk evaluation. Expert Syst Appl.
2006;30(3):489–497.
[11] Huang CL, Chen MC, Wang CJ. Credit scoring with a data mining approach based on support
vector machines. Expert Syst Appl. 2007.33(4):847–856.
[12] Wang SJ, Mathew A, Chen Y, et al. Empirical analysis of support vector machine ensemble
classifiers. Exp Syst Appl. 2009;36(3):6466–6476.
[13] Khashman A, Gray MR, Givens JA, et al. Neural networks for credit risk evaluation: investigation of different neural models and learning schemes. Expert Syst Appl. 2010.37(9):6233–6239.

[14] Oreški S, Oreški G. Cost-sensitive learning from imbalanced datasets for retail credit risk assessment. TEM J-Technol Educ Manage Inform. 2018;7(1):59–73.

[15] Deekshitulu B, Chandra P, et al. Classification of heart disease using k-nearest neighbor and genetic algorithm. Procedia Technol. 2013;10:85–94.

[16] Bikku T, Nandam SR, Akepogu AR. A contemporary feature selection and classification framework for imbalanced biomedical datasets. Egypt Inform J. 2018;19(3):191–198.

[17] Jain D, Singh V. Feature selection and classification systems for chronic disease prediction: A review. Egypt Inform J. 2018;19(3):79–189.

[18] Mohamed TM. Pulsar selection using fuzzy knn classifier. Future Comput Info J. 2018;3(1):1–6.

[19] Cherfi A, Nouira K, Ferchichi A. A very fast c4. 5 decision tree algorithm. Appl Artif Intell. 2018;32(2):119–137.

[20] Kamiński B, Jakubczyk M, Szufel P, et al. A framework for sensitivity analysis of decision trees. Cent Eur J Oper Res. 2018.26(1):135–159.

[21] Friedman J, Hastie T, Tibshirani R, et al. The elements of statistical learning, Vol. 1. New York: Springer; 2001.

[22] Jaiswal JK, Samikannu R. Application of random forest algorithm on feature subset selection and classification and regression. In 2017 World Congress on Computing and Communication Technologies (WCCCT) IEEE; 2017. p. 65–68.

[23] Ho TK. Random decision forests. In Document analysis and recognition, 1995., proceedings of the third international conference on 1995; vol. 1, IEEE. p. 278–282.

[24] Kleinberg E. Stochastic discrimination. Ann Math Artif Intell. 1990.1(1):207–239.

[25] Kleinberg E, et al. An overtraining-resistant stochastic modeling method for pattern recognition. Ann Stat. 1996.24(6):2319–2349.

[26] Kleinberg EM. On the algorithmic implementation of stochastic discrimination. IEEE Trans Pattern Anal Mach Intell. 2000.5:473–490.

[27] Breiman L. Random forests. Mach Learn. 2001.45(7):5–32.

[28] Amit Y, Geman D. Shape quantization and recognition with randomized trees. Neural Comput. 1997;9(7):1545–1588.

[29] Čeh M, Kilibarda M, Lisec A, et al. Estimating the performance of random forest versus multiple regression for predicting prices of the apartments. ISPRS Int J Geo-Inf. 2018.7(5):168.

[30] Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995.20(3):273–297.

[31] Cha KJ, Yoo KH, Lee CU, et al. Classification of acoustic noise signals using wavelet spectrum based support vector machine. J Mech Sci Technol. 2018.32(6):2453–2462.

[32] Chandrashekar G, Sahin F. A survey on feature selection methods. Comput Electr Eng. 2014.40(1):16–28.

[33] Lamba M, Munjal G, Gigras Y. Feature selection of micro-array expression data (fsm)-a review. Procedia Comput Sci. 2018.132:1619–1625.

[34] Yelghi A, Köse C. A modified firefly algorithm for global minimum optimization. Appl Soft Comput. 2018;62:29–44.

[35] Yang XS. Nature-inspired metaheuristic algorithms. United Kingdom: Luniver Prees, University of cambridge; 2010.

[36] Elkhechafi M, Benmamoun Z, Hachimi H, et al. Firefly algorithm for supply chain optimization. Lobachevskii J Math. 2018.39(3):355–367.

[37] Tighzert L, Fonlupt C, Mendil B. A set of new compact firefly algorithms. Swarm Evol Comput. 2018;40:92–115.

[38] Murphy P, Aha D. UCI repository of machine learning databases; 1994.

[39] Available from: https://www.kaggle.com/dansbecker/aer-credit-card-data

[40] Thomas LC, Edelman DB, Crook JN. Credit scoring and its applications. Philadelphia: SIAM; 2002;