



University of
Salford
MANCHESTER

Credit card fraud detection using asexual reproduction optimization

Farhang, A, Mansouri, T, Sadeghi Moghaddam, MR, Bahrambeik, N, Yavari, R and Fani Sani, MR

<http://dx.doi.org/10.1108/K-04-2021-0324>

Title	Credit card fraud detection using asexual reproduction optimization
Authors	Farhang, A, Mansouri, T, Sadeghi Moghaddam, MR, Bahrambeik, N, Yavari, R and Fani Sani, MR
Publication title	Kybernetes
Publisher	Emerald Publishing
Type	Article
USIR URL	This version is available at: http://usir.salford.ac.uk/id/eprint/61055/
Published Date	2021

USIR is a digital collection of the research output of the University of Salford. Where copyright permits, full text material held in the repository is made freely available online and can be read, downloaded and copied for non-commercial private study or research purposes. Please check the manuscript for any further copyright restrictions.

For more information, including our policy and submission procedure, please contact the Repository Team at: library-research@salford.ac.uk.



Credit Card Fraud Detection Using Asexual Reproduction Optimization

Journal:	<i>Kybernetes</i>
Manuscript ID	K-04-2021-0324.R1
Manuscript Type:	Research Paper
Keywords:	Artificial intelligence, Information

SCHOLARONE™
Manuscripts

Credit Card Fraud Detection Using Asexual Reproduction Optimization

No Author Given

No Institute Given

Abstract As the number of credit card users has increased, detecting fraud in this domain has become a vital issue. Previous literature has applied various supervised and unsupervised machine learning methods to find an effective fraud detection system. However, some of these methods require an enormous amount of time to achieve reasonable accuracy. In this paper, an Asexual Reproduction Optimization (ARO) approach was employed, which is a supervised method to detect credit card frauds. ARO refers to a kind of production in which one parent produces some offspring. By applying this method and sampling just from the majority class, the classification’s effectiveness is increased. A comparison to Artificial Immune Systems (AIS), which is one of the best methods implemented on current datasets, has shown that the proposed method is able to remarkably reduce the required training time and at the same time increase the recall that is important in fraud detection problems. The obtained results show that ARO achieves the best cost in a short time, and consequently, it can be considered as a real-time fraud detection system.

Keywords: Machine Learning · Asexual Reproduction Optimization · Credit Card Fraud Detection · Fraud Detection · Artificial Immune Systems.

1 Introduction

Credit card fraud inflicts plenty of costs on banks and card issuers and threatens their reputation [1]. A huge amount of money disappears annually from legitimate accounts by fraudulent transactions [2]. In fact, E-business has become one of the most important global markets which demands strong fraud detection systems [3, 4]. In 2017, Online Fraud Report of Cyber Source distinguishes average annual fraud loss among different order channels¹. 0.9% of the annual e-commerce revenues is lost due to payment frauds through Web store channel in North America. This value is 0.8% for Mobile channels and 0.3% for phone/mail order channel. Different definitions of fraud have been presented by different organizations. Based on The World Bank Group’s definition of fraud, the fraudulent practice covers solicitation, offering or taking bribes, or the manipulation of loans in the form of misrepresentation [5]. According to the division of the Association of Certified Fraud Examiners, there are two types of fraud, i.e., internal frauds and external frauds. Internal fraud occurs when an employee deliberately misuses an organization’s properties [6]. External frauds include a more comprehensive

¹ <http://www.cybersource.com>

2 No Author Given

variety in comparison with internal frauds. Dishonest vendors who take bribes are a desirable example to mention. Untruthful customers might alter account information to mislead payments. Besides, third parties may use intellectual properties [7].

The credit card fraud techniques have changed over time, from physically stealing the cards to online frauds [4]. Credit card frauds are categorized into two categories, i.e., application frauds and behavioral frauds. An application defrauder is a person who gets a new credit card from issuing companies by utilizing the wrong information. A behavioral defrauder is a person who has attained the information of a legitimate card fraudulently and makes purchases when the cardholder is not present [8]. As the number of frauds increases, the fighting techniques against fraud become more significant [9]. Protection techniques against fraud include prevention and detection systems. The first layer to protect the system against fraud is prevention. Fraud prevention stops the fraud from occurring at the initial level. Fraud detection is the next protection step. It identifies fraudulent activities when they penetrate the system [10]. People use credit card-based online payments more and more these days, forcing the banks to deploy fraud detection systems [11]. Expert-driven, data-driven, and the combination of both are the three kinds of fraud detection systems. Expert-driven systems are based on fraud scenarios. If the data-stream matches the scenario from the FDS viewpoint, the fraud has happened. Data-driven methods learn the fraud patterns and find them in data streams [12].

Credit card fraud happens when a transaction on someone's credit card is done by another person [13]. If the fraud becomes a prevalent issue in a competitive environment without any preventive systems, it will threaten businesses and organizations seriously [6]. On the other hand, the number of credit card transactions is increasing rapidly, which results in the growth of fraudulent activities [14]. It is pretty expensive to analyze the transaction is done by the client or not [15]. The fraud detection system is aimed to stop it as soon as possible. Whether the fraud detection system is manual or automatic, it has to be effective. The system should identify a high percentage of fraudulent transactions while keeping the false alarm rate low. Otherwise, the users will become apathetic to alarms [16]. To reduce the cost of detection, many machine learning techniques have been implemented. Supervised methods are more common than unsupervised techniques [8].

Nowadays, acknowledging the development of data mining methods, efficient ways have been found to detect fraud [17]. However, many of these methods need a time-consuming training phase. This limitation decreases the applicability of these methods. To address this problem, we propose to use Asexual Reproduction Optimization (ARO). In this paper, we implemented and applied this method on a publicly available dataset. The experimental results show that using the proposed method enables us to achieve reasonable accuracy faster, compared to one of the state-of-the-art fraud detection methods, i.e., Artificial Immune Systems (AIS).

The remaining part of the paper has been organized as follows. Section 2 provides a literature review on credit card fraud detection methods. Following, Section 3 describes the ARO and AIS models. Afterwards, experimental results are presented in Section 4 and analyzed in Section 5. Finally, Section 6 concludes the paper and provides some new directions to continue this research.

Table 1: Approaches for credit card fraud detection.

Machine learning algorithm	Method	references
Supervised classifying	k-Nearest Neighbors (KNN)	[18], [19], [1], [20], [21], [22], [23], [24], [25]
	Bayesian Networks (BN)	[26], [27], [28], [15], [15], [29], [30], [31], [32]
	Decision Trees (DT)	[33], [19], [28], [15], [34], [35], [20], [36] [37] [38] [39]; [40], [41], [42], [43], [44], [25]
	Artificial Immune Systems (AIS)	[45], [28], [15], [1], [46], [47]
	Naive Bayes (NB)	[48], [49], [50], [18], [19], [27], [15], [28], [20], [51], [52], [38], [32], [24], [24], [22], [53], [43], [23], [41], [25]
	Support Vector Machines (SVM)	[4], [17], [54], [55], [56], [57], [33], [19], [20], [58], [38], [39], [40], [59], [23], [60], [61], [62], [25]
	Logistic Regression	[18], [4], [17], [19] [38] [24], [22], [61], [53], [39], [63]
	Random Forest	[50], [4], [17], [33], [19], [12], [20] [51], [38], [64], [63], [44], [65], [66], [43], [53], [67], [60]; [68]; [69]
	Genetic Algorithm (GA)	[70], [71], [72], [73], [74], [75], [76]
	Neural Networks (NN)	[77], [78], [79], [80], [2], [19], [81], [11], [82], [28], [15], [83], [84], [85], [86], [24], [87], [88], [61], [89], [90], [91], [92], [93], [12], [94], [95], [20], [29], [96], [72], [97], [38], [98], [39], [64], [99], [63], [100], [101], [102], [62]
	Scatter Search	[70]
	APATE	[63]
	Fisher Discriminant	[103]
	Self-Organizing Maps	[104], [105], [106]
Unsupervised clustering	Fuzzy	[78], [107], [108], [109]
	Principal Component Analysis	[110]
	Hidden Markov Model (HMM)	[111], [112], [113], [114]; [115], [116], [69]
	Simple K-Means	[110]

2 Credit Card Fraud Detection Methods

Fraud detection merges anomaly-based detection and misuse-based detection by applying data mining techniques. Anomaly-based detection consists of supervised, unsupervised, and semi-supervised algorithms [117]. Supervised algorithms require all existing transactions, which are labeled as fraudulent and non-fraudulent transactions. These algorithms assign a score to a new transaction, which determines the transaction’s label [6, 8]. Unsupervised methods work with unlabeled test dataset and try to find the unusual transactions. These algorithms represent a baseline distribution for the normal behavior. Transactions with a great distance from it are considered unusual ones [8, 118]. Semi-supervised methods contain both labeled and unlabeled instances. Semi-supervised learning aims to design the algorithms, which can use these combined instances [118]. In general, the concept of anomaly/outlier is problem-dependent and it is challenging to capture all aspects of behavior in one single metric [119]. In Table 1, we presented some of the data mining based approaches which are used for credit card fraud detection, carried out in the literature [117, 120–122].

In [77], the authors presented a neural network-based system with a user-friendly interface for fraud detection, implemented on synthetic datasets [77]. In credit card fraud detection, datasets have skewed distributions. Chen et al. employed Binary Support Vector System (BSVS), which could handle this problem better than oversampling techniques. For support vector selection, the genetic algorithm is used. Based on these vectors, they proposed BSVS [54]. Gadi et al. employed BN, NB, AIS, and DT techniques on the Brazilian bank dataset that we used in this paper. They showed that generally applying cost-sensitive and robust optimization leads to better results [28]. Because of optimizing the parameters, AIS is the best technique [15]. Sánchez et al. applied the association rules in the credit card fraud detection system. The system determined patterns for legitimate transactions. The transactions that do not match with the patterns are recognized as fraudulent [108]. Instead of looking individually at data, authors in [62] consider them sequentially. They applied SVM and Long Short-Term Memory Recurrent Neural Network (LSTM) for modeling time series in fraud detection records. LSTM was a more suitable classifier [62]. Rani et al. suggested a method using HMM which could conserve user’s data effectively and bring back the information with ease [115]. Modi et al. examined a single-layer feed-forward neural network

4 No Author Given

for fraud detection. The fraud categorization was divided into four groups of low to high risk. If a transaction is recognized as a fraudulent one, it will belong to one of these groups [96].

Using negative selection in addition to clonal selection, Halvaiee and Akbari improved AIS. They suggested a new method AIS-based Fraud Detection Model (AFDM) for calculating the samples' fitness. Furthermore, in their proposed model, they used cloud computing for training, which reduced the processing time [1]. Zareapoor and Shamsolmoali examined bagged ensemble decision tree on a real dataset and compared it with SVM, KNN, and NB. It achieved the highest detection rate. The time was reduced significantly, and the ensemble technique could solve the imbalanced dataset problem [25]. Carneiro et al. aim the development and implementation of a fraud detection system at an e-tail merchant. They showed that choosing the right variables in the dataset is a key factor. Random forests, logistic regression, and support vector machines were tested. A random forest can be an appropriate practical model [17]. Fiore et al. used Generative Adversarial Networks (GAN) to detect credit card fraud. GAN is a multiple-layer neural network consisted of a generator and a discriminator. They employed GAN for solving imbalanced dataset problem. GAN generates an augmented dataset that has more fraudulent transactions than the initial dataset [11]. Behera and Panigrahi proposed a two-stage system. The first stage tries to match the patterns. It consists of a fuzzy module which computes a score. Given this score, one can envisage three categories: legitimate, fraudulent, and suspicious. The next stage concludes a neural network, which determines whether the suspicious one belongs to a fraudulent or legitimate group [78].

De Sá et al. implemented a customized Bayesian Network Classifier (BNC) on the dataset of a Brazilian payment service. They used a Hyper-Heuristic Evolutionary Algorithm for generating BNC. The proposed method increased economic efficiency remarkably [26]. Gómez et al. used an end-to-end neural network for credit card fraud detection. They focused on solving imbalanced dataset and cost evaluation problems and obtained valuable results [84]. Lucas et al. modelled a sequence of credit card transactions from three different perspectives. Each of these sequences with HMM and the likelihood associated with HMM is used as additional features in the Random Forest classifier for fraud detection [69]. Gianini et al. used a game theory-based approach for detecting credit card fraud by managing a pool of rules [123].

Monirzadeh et al. increased the efficiency of the neural network by using the genetic algorithm. Their research showed that the most effective criterion is the information related to the transaction. Age, gender, and such factors do not affect the detection [72]. In any fraud detection system, the chief problem is always to increase the accuracy of approving a legal transaction, whether in the shortest possible time or at the lowest cost for financial institutions [15]. Therefore, the principal purpose of all the models presented for this issue is to reduce the detection time, increase the accuracy, reduce the costs, and present a model that can improve these factors with better performance. According to the description, the algorithms' performance has been compared through the three aspects of fraud detection speed, accuracy, and cost presented in Table 2 [14].

As shown in Table 2, most of the algorithms have some disadvantages in the mentioned indicators, and among them, AIS performs the best. This confirms the results

Table 2: Comparison of algorithms.

Algorithm	NN	BN	SVM	KNN	DT	fuzzy	AIS	GA	HMM
Fraud detection speed	Fast	Very fast	Low	Good	Fast	Very low	Very fast	Good	Fast
Accuracy	Medium	High	Medium	Medium	Medium	Very high	Good	Medium	Low
Cost	Expensive	Expensive	Expensive	Expensive	Expensive	High expensive	Inexpensive	Inexpensive	High expensive

presented in [28], where different techniques are compared with each other and AIS is the best technique based on their costs [15]. For this reason, it is chosen for comparison with the ARO algorithm. We employed ARO, which is a supervised method for credit card fraud detection. ARO is an asexual reproduction optimization algorithm. Like Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Ant Colony Optimization (ACO), ARO is also an Evolutionary Single-Objective Optimization technique [124]. ARO has some advantages that make it completely different from other algorithms. First, it is an individually based technique which reaches the global optimal point, astonishingly faster than other algorithms. Thus, unlike population-based algorithms that require a large number of computational resources to convert, ARO consumes much fewer resources and converges faster. The second case is about mathematical convergence. It has good exploration and exploitation rates. Third, ARO does not require parameter settings, so it is unlikely to have trouble in setting parameters, which is a common meta-cognitive problem of Genetic Algorithms (GA), Annealing Simulation (SA), Taboo Search (TS), and Particle Swarm Optimization (PSO). Besides, ARO does not use any selective mechanism such as a roulette wheel. Inappropriate adoption of selection mechanisms may lead to problems such as premature convergence due to excessive selection pressure. Fourth, ARO is a free model algorithm that can be applied to various types of optimization [125, 126]. For all of the above reasons, ARO can be selected for comparison with the AIS in fraud detection problems. ARO has not been used in fraud detection up to this point. In this paper, a comparison is made between ARO and AIS. We run ARO on the same dataset on which the AIS has been implemented [1, 28].

3 Using ARO for Fraud Detection

In this section, we explain ARO in more details and how we implement that to detect fraud. Moreover, the AIS algorithm that has the highest performance is briefly explained [1]. As the ARO method is a supervised method, we need to separate the data into train and validation parts. Therefore, like any other supervised method, we use the train part of data for the training and the validation part for the testing phase.

ARO algorithm ARO is taken from asexual reproduction. Asexual reproduction refers to a kind of production in which one parent produces offspring identical to herself [127]. In populations like bacteria, asexual reproduction is prevalent [128]. There are several kinds of asexual reproduction like budding [129], asexual spore production [130], and binary fission [131]. ARO is inspired by the budding method. In the budding method, the parent produces a smaller copy of itself called a bud. The bud separates itself from the parent to become an independent one [124].

6 No Author Given

Here, we explain how we use ARO for detecting frauds. According to the label of transactions in the training data, two separate matrices were created for fraud and legal transactions. For each feature in the legal matrix, the maximum and the minimum values are determined and placed in the maximum and the minimum legal matrices, and a parent is created randomly between the values of the maximum and the minimum matrices. Note that the value of each bit in this parent is a random value between its corresponding bit in the maximum and the minimum legal matrices. The value of the parent fitting was calculated using the fitting function given in Equation 2 and named "parent-fitting".

$$distance_{record-normal-transitions} = \frac{\sum_{i=1}^M \sum_{j=1}^N \left(\frac{|r_i - nt_{ji}|}{max_i - min_i} \right)}{kN} \quad (1)$$

Where M is the number of features in our dataset.

The cut point in a dataset is the best fitness achieved in that dataset.

$$distance_{record-normal-transitions} = \frac{\sum_{i=1}^M \sum_{j=1}^N \left(\frac{|r_i - nt_{ji}|}{max_i - min_i} \right)}{kN} \quad (2)$$

Afterwards, we repeat the following process until the parent fit is smaller than the cut point of the data set.

- Select the starting bit (S) as a random number within the range of the number of features. Select the end (E) between the starting bit and the last number of features. Calculate the probability of mutation through Equation 3:

$$P = \frac{1}{1 + Ln(E - S + 1)} \quad (3)$$

- Put the bud equal to the parent.
- For the bits between the starting and ending bit selected randomly as above, if the probability P calculated in Equation 3 is greater than or equal to an arbitrary random number between zero and one in MATLAB, the value of the bit will be mutated. In this way, the bud will be mutated.
- The value of the mutated bud in Equation 4 is calculated (using the fitting function) and named as "bud fit".

$$distance_{record-final-transitions} = distance_{record-fraud-transitions} - distance_{record-normal-transitions} \quad (4)$$

- If the bud is fitted, it is more than the parent, and the bud replaces the parent. Fitting the bud replaces the parent fitting. The bud is added to the identifier matrix, and one unit is added to the count of the matrix rows of the identifier.
- In each fitting calculation, separate bud fits are calculated for the fraudulent matrix and the legal matrix. Fit the bud for the fraudulent matrix added to fraudulent matrix Equation 5. One is added to the counter of the fraudulent fitting matrix. The fitting of the bud for the legal matrix is added to the legal matrix Equation 2, and one is added to the count of the legal matrix rows. The loop termination condition reaches to a value more than or equal to the parent fit compared to the cut point.

$$distance_{record-fraud-transitions} = \frac{\sum_{i=1}^M \sum_{j=1}^F \left(\frac{|r_i - ft_{ji}|}{max_i - min_i} \right)}{kF} \quad (5)$$

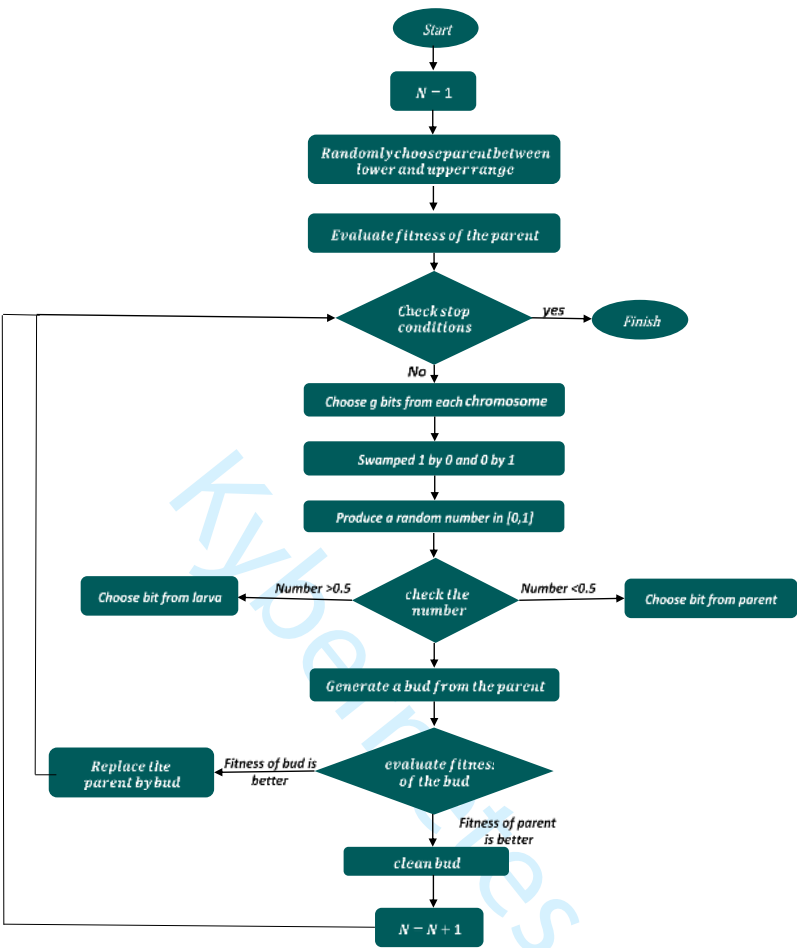


Figure 1: Flowchart of ARO.

The schematic view of ARO algorithm is presented in Figure 1. In ARO algorithm, an individual is shown by a vector of variables $X=(x_1, x_2, \dots, x_n)$, $X \in \mathcal{R}_n$. Each variable is considered as a chromosome. A binary string represents a chromosome consisted of genes. The length of the string is $L=l_1 + l_2 + 1$. It is supposed that every generated answer exists in the environment, and because of limited resources, only the best solution can remain alive. The algorithm starts with a random individual in the answer scope. This parent reproduces the offspring named bud. Just the parent or the offspring can survive. In this competition, the one which outperforms in fitness function remains alive. If the offspring has suitable performance, it will be the next parent, and the current parent becomes obsolete. Otherwise, the offspring perishes, and the present parent survives. The algorithm recurs until the stop condition occurs.

8 No Author Given

In the reproduction stage, a substring with λ bits is picked out in all chromosomes, which is named larva. λ is a random number between 1 and L . In the exploration phase, the substring is mutated, in each gene in the substring, 1 is swapped by 0 and 0 by 1. In the exploitation phase, the parent and larva merge as shown in Figure 3. Process of bud reproduction. If P which is calculated from $P = \frac{1}{1 + L_n(\lambda)}$ is higher than 0.5, the bud gene is picked out from the larva, otherwise the bud gene will be picked out from the parent chromosome.



Figure 2: A model for a chromosome in ARO.

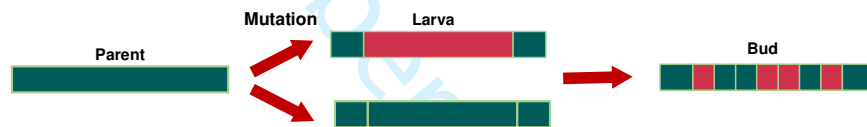


Figure 3: Process of bud reproduction.

Equation 6 relates the exploitation and exploration. If λ is a big number, less exploitation is needed and vice versa. In fact, exploration and exploitation are inversely related.

$$P = \frac{1}{1 + L_n(\lambda)} \quad (6)$$

The fitness of both bud and parent is calculated to choose the best one for the algorithm's next run after reproduction [124, 125]. Note that we do this procedure for all records and all features. Each record has a fraud or normal label. There are the following hints to mention:

1. According to Figure 2, a chromosome has three parts. Here, just the integer part is considered because we do not have the sign or decimal part.
2. Genes are not binary and they contain integer numbers.
3. Only the normal (or legal) records are sampled because the dataset is skewed toward normal transactions. The number of normal transactions is significantly more than the fraudulent transactions. Thus, normal records society is suitable for sampling versus the fraudulent society.

4. For generating the first parent, one should determine the range for each bit (gene). Then, for each gene in the first parent chromosome, a random number between the maximum and the minimum of that gene is chosen.
5. The fitness function will be used, which is described in the next section.
6. First, a larva should be generated when reproducing a bud. For generating a larva, a random length should be created. Each gene in this length assumes a random number between the maximum and the minimum of that gene. This length would be a larva. The next step for reproducing a bud is choosing the gene between larva and parent, like in Figure 3 (the process of bud reproduction). In this step, for choosing each gene between larva and parent, a random number is generated. If the random number is less than p , which is obtained from (1), the gene is selected from the larva, otherwise the gene is selected from a parent.

Parameter setting causes plenty of problems in methods such as PSO and GA. ARO does not need parameter setting. ARO is an individual-based technique which saves time, unlike population-based techniques that waste time. ARO can be used in different kinds of optimization issues despite many algorithms, which can only be used in one sort of optimization problems. Adjustment with a diverse genetic environment is one of the problems faced in ARO. However, it can be solved by special reproduction operators [124].

Fitness function In case one decides to evaluate fitness for a specific record, at first, the distance between the record and all fraud transactions is calculated by Equation 4, and then the distance between the record and all normal transactions is calculated by Equation 1. The difference between these two numbers, as shown in Equation 3, would be the fitness. Due to only sampling the normal records, the higher is the fitness number, the better it is because it shows that the record is closer to the normal transactions than the fraud ones. Thus, it can be a suitable normal sample. In Equations 4 and 1, each record is considered to have k fields. Here, k is 17. The value of the i 'th field of the record is r , the value of the i 'th field of the j 'th normal transaction is nt_{ji} , and the value of the i 'th field of the j 'th fraud transaction is ft_{ji} . The maximum and the minimum of i 'th field in all records of dataset are represented by max_i and min_i . The number of all normal transactions in the considered dataset is N , and the number of all fraudulent transactions is F .

AIS algorithm AIS is inspired by the immune system of the human body. It creates the detectors called lymphocytes for identifying non-self-cells like viruses. Negative selection and clonal selection are two stages of the AIS. Through negative selection, lymphocytes are created by a random combination of protein patterns. Lymphocytes should not detect self-cells. Thus, the immune system eliminates the lymphocytes that react to self-cells. In fact, all of the lymphocytes generated randomly that react to self-cells are eliminated immediately after creation, and other lymphocytes survive. This procedure is named negative selection. After negative selection, a short life starts for the remaining lymphocytes. They meet any non-self-cells. If any lymphocyte reacts to a non-self-cell, it can survive to protect the body against those non-self-cells. This procedure is named clonal selection. The lymphocyte which detects a non-self-cell is cloned by mutation. The colony cells, which are closer to the non-self-cell, are chosen

10 No Author Given

to survive. These colony cells are considered as memory cells and will react to non-self-cells like viruses [1].

Both non-self-/self-cells are considered as vectors. At first, the training-set should be normalized. Initializing the parameters is the next step. Then, N_{pop} of normal records is selected randomly as primary detectors (Just the normal records like ARO were sampled). The affinity of these records is calculated using the distance function. N_c of the records with higher affinity is selected. A colony is expanded from them. It means the records with more affinity will be replicated more. The colony is mutated. N_m of the best-mutated population is chosen to replicate N_m of the worst memory cells. This algorithm continues until the stop condition occurs [1]. Here, the loop repeats are considered 150 times. N_{pop} , N_c , and N_m are 25, 7, and 5, which have been driven from Gadi et al. and Halvaiee [1, 15].

Algorithm 1 PSO

```

1: Determine  $N_{pop}$  % the number of all detectors
2: Determine  $N_c$  % the number of detectors best match with non - self cell
3: Determine  $N_m$  % the number of best mutated detectors
4: while stop conditions do not occur do
5:   Choose  $N_{pop}$  of population randomly, call it first-pop
6:   Choose  $N_c$  of best first-pop based on their fitness, call it best-first-pop
7:   Expand a colony from best-first-pop, call it colony-pop
8:   Mutate colony-pop, call it mutated-pop
9:   Choose  $N_m$  of best mutated-pop based on their fitness, call it best-mutated-pop;
10:  Replace  $N_m$  of worst detectors in memory cell by best - mutated - pop
11: end while

```

4 Experiments

In this section, first, the experimental dataset is described. Afterward, we explain some details of the experimental-setting, and next, we will present the results based on the metrics discussed above.

4.1 Dataset

In our experiments, we used a Brazilian bank's dataset, according to which 3.74% of all transactions are fraudulent. Nine splits are generated from all transactions in the dataset. Each split has two parts. The first part, which contains 70% of transactions, is for the training phase. The second part, which contains 30% of transactions, is for the testing phase. The number of fraudulent and legitimate transactions in each split is shown in Table 4 [28]. We used MATLAB 2016 software for AIS and ARO implementation. Thus, we changed the format of datasets to CSV. We trained the fraud detection system by two methods, i.e., ARO and AIS, as explained in the previous sections.

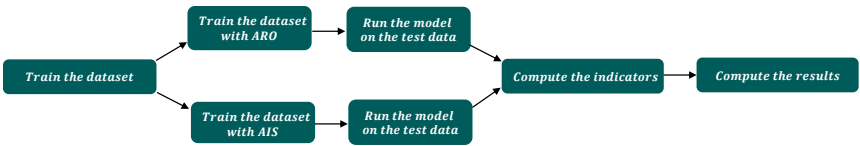


Figure 4: Fraud detection system.

4.2 Experimental Settings

After training the model, in the second step, we ran our model on the validation data with labeled transactions as fraud or normal. Then, in the next step, a comparison is made between the predicted labels and real labels by calculating four parameters:

- *False positive (FP)*: The number of normal transactions that mistakenly predicted as frauds by our method.
- *False negative (FN)*: The number of fraud transactions that mistakenly predicted as normal by our method.
- *True positive (TP)*: The number of fraud transactions that correctly detected by our method.
- *True negative (TN)*: The number of normal transactions that correctly detected by our method.

Using the above parameters, we are able to compute some common metrics to evaluate the performance. We used four metrics in our testing phase:

- *Sensitivity* ($\frac{TP}{TP+FN}$): It is the ability to recognize a fraudulent transaction as a fraudulent one.
- *Precision* $\frac{TP}{TP+FP}$: It is the accuracy on cases predicted as fraud.
- *Specificity* $\frac{TN}{FP+TN}$: It is the ability to recognize a legitimate transaction as a legitimate one.
- *Accuracy* $\frac{TP+TN}{TP+FP+TN+FN}$: It presents the proportion of correct predictions.

In addition, we measured training and testing time, which are critical issues in fraud detection. We used Equation 7 for cost calculating on this dataset. Gadi et al. used this formula because the dataset has 100% of fraudulent records and only 10% of legitimate records [15].

$$Cost = 100 \times FN + 10 \times FP + TP \quad (7)$$

In the next step, we compared the performance of the two algorithms. The whole process of the fraud detection system is described in Figure 4.

As mentioned before, each dataset has a specific cut point. By trial and error method, we found the cut points presented in Table 3. In each training dataset, there are about 28,000 records with 17 features. Finally, test (or validation) datasets are used in the testing phase. For testing the samples obtained by ARO or AIS method, these steps are followed:

12 No Author Given

Table 3: Cut points in the datasets.

dataset	1	2	3	4	5	6	7	8	9
Cut point	0.1754	0.1841	0.1739	0.1762	0.175	0.1777	0.176	0.1916	0.1749

Table 4: Number of fraudulent and legitimate transactions in datasets.

Split type	Transaction type	1	2	3	4	5	6	7	8	9
train	Legitimate	27,904	28,012	28,061	28,145	28,045	27,973	28,113	27,884	28,188
	Fraudulent	1,084	1,092	1,088	1,075	1,081	1,116	1,099	1,106	1,100
test	Legitimate	12,184	12,076	12,027	11,943	12,043	12,115	11,975	12,204	11,960
	Fraudulent	475	467	471	484	478	443	460	453	459

1. The distance of the record from all normal samples is measured.
2. The distance is divided by the number of normal samples. One can call it final distance.
3. If the final distance is below the best cut-off value, one can categorize the distance as normal, otherwise it would be fraudulent.

The performance is measured by the metrics discussed above. For the AIS method, we have provided the results presented in [1], also the results of our implemented version of this algorithm to have a more fair comparison. All the codes are available in <https://gitlab.com/Anahita-Farhang/ARO-AIS>.

4.3 Experimental results

This subsection presents the computational results of running AIS and ARO algorithms¹. In sensitivity, precision, specificity, and accuracy, ARO achieved a higher average than AIS, as shown in Table 5. For training time, test time, and cost, ARO shows better performance. Results are shown in Table 6. As shown in Figure 5, ROC curve of testing results with ARO and AIS algorithm for all datasets, by implementing ARO, AUC, which is a suitable criterion for imbalanced datasets, is increased by 13% more than the AIS method. It shows that for each cut-off value, ARO outperforms AIS.

Finally, two non-parametric statistical tests (i.e., wilcoxon and Kruskal-Wallis) were conducted to ensure the statistical significance in terms of accuracy for the ARO model. The Wilcoxon test results are illustrated in Table 8 that the ARO model almost reaches the significance level compared to AIS. The Kruskal-Wallis test was used to show the equality of results in all nine sections of the dataset. Results are presented in Table 9.

5 Discussion

We trained the system with ARO and AIS methods. As mentioned in Section 2, given that ARO is a single-solution evolutionary algorithm, it responds faster than the AIS that

¹ All the experiments were performed in a PC with an Intel® Core™ i5-3210M CPU @ 2.5GHz with 4GB RAM in Windows 8(x64).

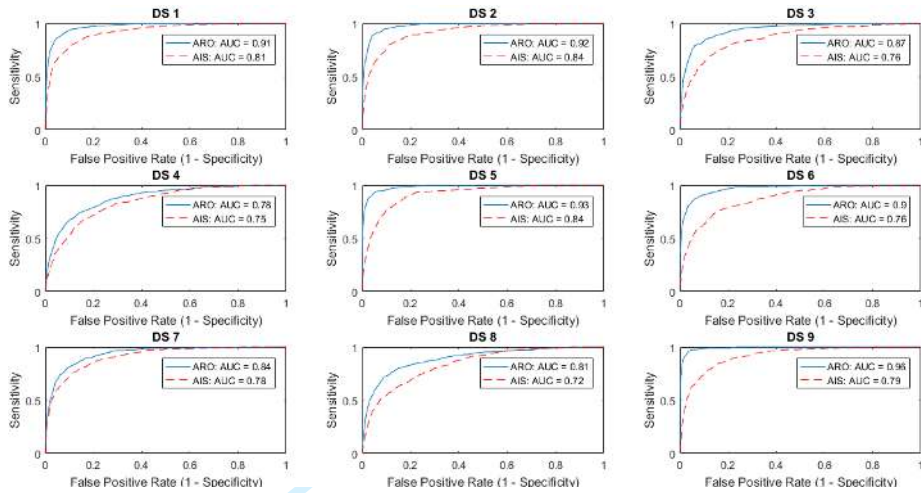


Figure 5: ROC curve of testing results with ARO and AIS algorithm for all datasets.

generates a community of data [124–126]. Therefore, the good speed with no parameter setting and good convergence rate have made ARO a good candidate versus AIS, and in our experiment, this claim was confirmed in four indicators. In classification problems, there are some common metrics to evaluate the performance: sensitivity (recall), precision, specificity, and accuracy. These four metrics have been measured in our testing phase. AUC was measured, which is the area under the ROC curve. ROC curve plots sensitivity versus false-positive rate. In fact, the cut-off value in the test phase is located at the top left corner in ROC curve. It is the point where sensitivity and specificity are equal. Gadi et al. found that if they use a cost function shown in Equation 7 in which they adopted an average cost of 1 dollar for every verification and an average loss of 100 dollars for every undetected fraud, they will obtain more applicable results. They

Table 5: The results of implementing ARO and AIS on datasets.

Metric	Sensitivity		Precision		Specificity		Accuracy	
	ARO	AIS	ARO	AIS	ARO	AIS	ARO	AIS
DS 1	0.86	0.68	0.42	0.33	0.95	0.95	0.95	0.94
DS 2	0.88	0.8	0.46	0.22	0.96	0.89	0.96	0.89
DS 3	0.79	0.61	0.34	0.22	0.94	0.92	0.93	0.9
DS 4	0.65	0.63	0.23	0.16	0.91	0.87	0.9	0.86
DS 5	0.88	0.78	0.58	0.23	0.97	0.9	0.97	0.89
DS 6	0.86	0.58	0.38	0.24	0.95	0.94	0.95	0.92
DS 7	0.74	0.6	0.32	0.34	0.94	0.96	0.93	0.94
DS 8	0.72	0.51	0.23	0.2	0.91	0.93	0.91	0.91
DS 9	0.95	0.63	0.54	0.33	0.97	0.95	0.97	0.94
Average	0.81	0.65	0.39	0.25	0.95	0.92	0.94	0.91

14 No Author Given

Table 6: The results of implementing ARO and AIS on datasets.

Metric	Train time (s)		Test time (s)		Cost		AUC	
	ARO	AIS	ARO	AIS	ARO	AIS	ARO	AIS
DS 1	8.08	24.25	1.87	1.85	12,570	22,072	0.91	0.81
DS 2	8.46	24.90	1.32	1.19	10,781	23,213	0.92	0.84
DS 3	7.33	24.65	2.22	1.75	17,473	28,966	0.87	0.76
DS 4	4.68	24.44	1.23	1.52	27,923	33,864	0.78	0.75
DS 5	4.63	24.66	1.13	1.62	9,132	23,115	0.93	0.84
DS 6	7.78	24.57	1.34	1.75	12,889	26,925	0.9	0.76
DS 7	3.8	24.25	1.27	1.68	19,522	24,224	0.84	0.78
DS 8	7.06	24.32	1.79	1.72	23,944	31,599	0.81	0.72
DS 9	4.43	25.30	1.54	1.80	6,407	23,071	0.96	0.79
Average	6.25	24.59	1.52	1.65	15,627	26,339	0.88	0.78

used this formula because the dataset has 100% of fraudulent records and only 10% of legitimate records [1, 15, 132, 133]. This was considered to be more similar to the practice used for a fraud score compared to a ROC curve that compares multiple references simultaneously [15].

One of the main problems of AIS is the extreme need for a hyper-parameter setting, which is not present in ARO. ARO is a bio-inspired algorithm, and we aimed to test this algorithm against one of the algorithms that works best in detecting fraud on a Brazilian bank's dataset. Compared to other studies on this dataset, the detecting speed and the computational cost were important. We trained each dataset by each algorithm thirty times and registered the results of the best cost. As shown in Tables 5 and 6, ARO has better performance than AIS in all the metrics. The ARO method's best performance appears on the ninth dataset with the sensitivity of 0.95 and the cost of 6,407, which is better than the AIS method (sensitivity=0.63 & cost=23071). The ARO method's worst performance appears on the fourth dataset with the sensitivity of 0.65 and the cost of 27,923, which is still better than the AIS method (sensitivity=0.63 & cost=33864). The average sensitivity for ARO is 0.81 with the average precision 0.39. For the AIS technique, the average sensitivity and precision are 0.65 and 0.25. Training time, which is a vital issue in fraud detection, has been remarkably reduced. The average training time for ARO is 6.25s, so ARO fraud detection system can be considered as a real-time one. ARO improved sensitivity up to 25%, and precision up to 56%, decreased cost up to 41%, and training time up to 75%. The first fraud detection on our dataset was implemented by Gadi et al. He proved that by optimizing the parameters, AIS is the best method in comparison with BN, NB, and DT [15]. One of the best fraud detection systems on this dataset was performed in [1]. They employed AFDM which is a kind of improved AIS method. They achieved 17,389 for cost and 79 seconds for training time. By implementing ARO, we achieved 15,627 for cost and 6.25 for training time which are considerably better than the previous results. The obtained results and the results of the previous researches are shown in Table 7.

In the last part, we have used two non-parametric statistical tests. We have applied Wilcoxon to show the significant difference between AIS and ARO algorithms. This

Table 7: History of the previous and obtained results.

Method	AIS	AFDM	AIS	ARO
Cost	23,303	17,389	26,339	15,627
Reference	[15]	[1]	Proposed AIS	Proposed ARO

test ranks all differences and applies a negative sign to all the ranks where the difference between the two observations is negative. The hypothesis H_0 in this test is the equality of the two algorithms and, as shown in Table 8, in accuracy, sensitivity, precision, train time, and cost, because of the p-values which are less than alpha ($\alpha = 0.05$), this hypothesis was rejected that means two algorithms are not equal. Also, in the Wilcoxon test, negative rank for train time, test time, cost, and positive rank for other indices showed that ARO performs better than AIS in all indices.

Table 8: Results of Wilcoxon signed-rank test with $\alpha=0.05$.

Compared model	Sensitivity	Precision	Specificity	Accuracy	Train time (s)	Test time (s)	Cost	AUC
Asymp. Sig.	0.008	0.011	0.118	0.020	0.008	0.314	0.008	0.008

We have done Kruskal-wallis test because our dataset was divided into nine sections and it is important to check whether all the samples are originated from the same distribution. We performed this test to check whether there is a significant difference between the nine samples in each index. The results are shown in Table 9. The hypothesis H_0 in this test is the equality between all the nine samples. Due to the p-values which are greater than alpha ($\alpha = 0.05$), and also because of the values of chi-square that are 8, which is less than 15.5073 ($\chi^2_{0.05} = 15.5073$ with $df = 8$), the H_0 hypothesis cannot be rejected which means in all indexes, our nine sections are the same.

Table 9: Results of Kruskal-Wallis test.

Compared model	Sensitivity	Precision	Specificity	Accuracy	Train time (s)	Test time (s)	Cost	AUC
Chi-square	8.000	8.000	8.000	8.000	8.000	8.000	8.000	8.000
df	8	8	8	8	8	8	8	8
Asymp. Sig.	0.433	0.433	0.433	0.433	0.433	0.433	0.433	0.433

As it was discussed, we have achieved promising results by using ARO algorithm. However, there is room for improvement. For example, an algorithm can be employed to choose the optimized cut-point values in Section 4.2. Moreover, to increase the performance of the algorithm, we suggest using cloud computing, i.e. implementing ARO algorithm on a cloud-based file system (e.g. Hadoop) which makes data parallelization possible. Furthermore, new methods in the deep learning area show progress in terms of time in comparison with metaheuristic algorithms. Therefore, employing deep learning methods may reduce the training time and have positive impacts on the final results.

16 No Author Given

6 Conclusion

Fraud is a critical concern for financial services (e.g., commercial banks, investment banks, insurance companies, etc.) and individuals. Different types of fraud cost millions of dollars every year. Among different types of fraud, credit card fraud is the most common one and several solutions have been proposed to detect fraudulent transactions. In this paper, we have implemented the ARO (Asexual Reproduction Optimization) in credit card fraud detection. This effective approach has achieved better results than the best techniques implemented on our dataset so far. We have compared the results with those of the AIS, which was one of the best methods ever implemented on the benchmark dataset.

The chief focus of the fraud detection studies is finding the algorithms that can detect legal transactions from the fraudulent ones with high detection accuracy in the shortest time and at a low cost. ARO meets all these demands. ARO is an Evolutionary Single-Objective Optimization algorithm with lots of advantages that make it suitable for implementing in fraud detection problems. First of all, being an individually based technique, it converges faster to the global optimal point. Secondly, it has good exploration and exploitation rates. Thirdly, it has no parameter settings, which is a common issue in meta-cognitive problems such as Genetic Algorithms (GA), Annealing Simulation (SA), Taboo Search (TS), and Particle Swarm Optimization (PSO). Results show that ARO has increased the AUC, sensitivity, precision, specificity, and accuracy by 13%, 25%, 56%, 3%, and 3%, in comparison with AIS, respectively. We have achieved a high precision value indicating that if ARO detects a record as a fraud, with a high probability, it is a fraud one. Supporting a real-time fraud detection system is another vital issue. ARO outperforms AIS not only in the mentioned criteria, but also decreases the training time by 75% in comparison with the AIS, which is significant. Furthermore, two non-parametric statistical tests (i.e., Wilcoxon and Kruskal-Wallis) were conducted to ensure the statistical significance in terms of accuracy for the ARO model. The Wilcoxon test results show that the ARO model almost reaches the significance level compared to AIS. The Kruskal-Wallis test was used to show the equality of results in all nine sections of the dataset. The results of applying these two statistical tests ensure the statistical significance in our study.

7 Future Work

Our framework has addressed the problems such as high costs and training time in credit card fraud detection. Although, there is still room for further improvement. To increase the performance of the proposed method, it is possible to test the proposed model in a cloud environment, i.e., Hadoop. Moreover, ARO can be compared to PSO and QPSO, which have fewer parameter settings than AIS.

In addition, the writers believe ARO has the potential to obtain much better results. One improvement can be done by weighting the fields that compose a transaction. In fact, there are plenty of fields in a transaction and some fields are more important than other fields. Therefore, we can increase or decrease the effect of the field on the final results through weighting fields in the distance function. Furthermore, the distance

function can be different for each property in the dataset. As we discussed, each transaction has several fields with different meanings. Then the concept of distance is not the same for all the fields. As an example, suppose the person goes shopping once per month. So the distance of 30 is usual and it equals zero for the time concept. However, the distance of 30 for the amount column is important and it is not equal to zero. Therefore, considering application-based distance functions for each field is an interesting point to address in future work.

References

1. Halvaiee, N.S., Akbari, M.K.: A novel model for credit card fraud detection using artificial immune systems. *Applied Soft Computing* **24** (2014) 40–49
2. Pozzolo, A.D., Boracchi, G., Caelen, O., Alippi, C., Bontempi, G.: Credit card fraud detection and concept-drift adaptation with delayed supervised information. In: 2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12–17, 2015, IEEE (2015) 1–8
3. Šumak, B., Heričko, M., Budimac, Z., Pušnik, M.: Investigation of moderator factors in e-business adoption: A quantitative meta-analysis of moderating effects on the drivers of intention and behavior. *Computer Science and Information Systems* **14**(1) (2017) 75–102
4. Bhattacharyya, S., Jha, S., Tharakunnel, K., Westland, J.C.: Data mining for credit card fraud: A comparative study. *Decision Support Systems* **50**(3) (2011) 602–613
5. Aguilar, M., Gill, J., Pino, L.: Preventing fraud and corruption in world bank projects. A Guide for Staff. Washington, DC: The World Bank (2000)
6. Phua, C., Lee, V., Smith, K., Gayler, R.: A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119* (2010)
7. Chen, S., Gangopadhyay, A.: A novel approach to uncover health care frauds through spectral analysis. In: 2013 IEEE International Conference on Healthcare Informatics, IEEE (2013) 499–504
8. Bolton, R.J., Hand, D.J., et al.: Unsupervised profiling methods for fraud detection. *Credit scoring and credit control VII* (2001) 235–255
9. Kou, Y., Lu, C.T., Sirwongwattana, S., Huang, Y.P.: Survey of fraud detection techniques. In: IEEE International Conference on Networking, Sensing and Control, 2004. Volume 2., IEEE (2004) 749–754
10. Behdad, M., Barone, L., Bennamoun, M., French, T.: Nature-inspired techniques in the context of fraud detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **42**(6) (2012) 1273–1290
11. Fiore, U., De Santis, A., Perla, F., Zanetti, P., Palmieri, F.: Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences* **479** (2019) 448–455
12. Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.E., He-Guelton, L., Caelen, O.: Sequence classification for credit-card fraud detection. *Expert Systems with Applications* **100** (2018) 234–245
13. Tripathi, K.K., Ragha, L.: Hybrid approach for credit card fraud detection. *Int. J. Soft Comput. Eng.(IJSCE)* **3**(4) (2013)
14. Zareapoor, M., Seeja, K., Alam, M.A.: Analysis on credit card fraud detection techniques: based on certain design criteria. *International journal of computer applications* **52**(3) (2012)
15. Gadi, M.F.A., Wang, X., do Lago, A.P.: Credit card fraud detection with artificial immune system, Springer (2008) 119–131

18 No Author Given

16. Axelsson, S.: The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security (TISSEC)* **3**(3) (2000) 186–205
17. Carneiro, N., Figueira, G., Costa, M.: A data mining based system for credit-card fraud detection in e-tail. *Decision Support Systems* **95** (2017) 91–101
18. Awoyemi, J.O., Adetunmbi, A.O., Oluwadare, S.A.: Credit card fraud detection using machine learning techniques: A comparative analysis, *IEEE* (2017) 1–9
19. Dhankhad, S., Mohammed, E., Far, B.: Supervised machine learning algorithms for credit card fraudulent transaction detection: A comparative study, *IEEE* (2018) 122–125
20. Kumari, P., Mishra, S.P.: Analysis of credit card fraud detection using fusion classifiers. In: *Computational Intelligence in Data Mining*. Springer (2019) 111–122
21. Yu, W.F., Wang, N.: Research on credit card fraud detection model based on distance sum, *IEEE* (2009) 353–356
22. Ito, F., Singh, S., et al.: Comparison and analysis of logistic regression, naïve bayes and knn machine learning algorithms for credit card fraud detection. *International Journal of Information Technology* (2020) 1–9
23. Prusti, D., Padmanabhuni, S.H., Rath, S.K.: Credit card fraud detection by implementing machine learning techniques. (2019)
24. Bagga, S., Goyal, A., Gupta, N., Goyal, A.: Credit card fraud detection using pipeling and ensemble learning. *Procedia Computer Science* **173** (2020) 104–112
25. Zareapoor, M., Shamsolmoali, P.: Application of credit card fraud detection: Based on bagging ensemble classifier. *Procedia Computer Science* **48** (2015) 679–685
26. de Sá, A.G.C., Pereira, A.C.M., Pappa, G.L.: A customized classification algorithm for credit card fraud detection. *Engineering Applications of Artificial Intelligence* **72** (2018) 21–29
27. Filippov, V., Mukhanov, L., Shchukin, B.: Credit card fraud detection system, *IEEE* (2008) 1–6
28. Gadi, M.F.A., Wang, X., do Lago, A.P.: Comparison with parametric optimization in credit card fraud detection, *IEEE* (2008) 279–285
29. Maes, S., Tuyts, K., Vanschoenwinkel, B., Manderick, B.: Credit card fraud detection using bayesian and neural networks. (2002) 261–270
30. Panigrahi, S., Kundu, A., Sural, S., Majumdar, A.K.: Credit card fraud detection: A fusion approach using dempster–shafer theory and bayesian learning. *Information Fusion* **10**(4) (2009) 354–363
31. Taha, A.A., Malebary, S.J.: An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. *IEEE Access* **8** (2020) 25579–25587
32. Yee, O.S., Sagadevan, S., Malim, N.H.A.H.: Credit card fraud detection using machine learning as data mining technique. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* **10**(1-4) (2018) 23–27
33. Devi, J.V., Kavitha, K.S.: Fraud detection in credit card transactions by using classification algorithms, *IEEE* (2017) 125–131
34. Kavitha, M., Suriakala, M.: Hybrid multi-level credit card fraud detection system by bagging multiple boosted trees (bmbt), *IEEE* (2017) 1–5
35. Kokkinaki, A.I.: On atypical database transactions: identification of probable frauds using machine learning for user profiling, *IEEE* (1997) 107–113
36. Minegishi, T., Niimi, A.: Proposal of credit card fraudulent use detection by online-type decision tree construction and verification of generality. *International Journal for Information Security Research (IJISR)* **1**(4) (2011) 229–235
37. Patil, D.D., Karad, S.M., Wadhai, V.M., Gokhale, J.A., Halgaonkar, P.S.: Efficient scalable multi-level classification scheme for credit card fraud detection. *IJCSNS International Journal of Computer Science and Network Security* **10**(8) (2010) 123–130

38. Randhawa, K., Loo, C.K., Seera, M., Lim, C.P., Nandi, A.K.: Credit card fraud detection using adaboost and majority voting. *IEEE ACCESS* **6** (2018) 14277–14284

39. Sahin, Y., Duman, E.: Detecting credit card fraud by ann and logistic regression, *IEEE* (2011) 315–319

40. Sisodia, D.S., Reddy, N.K., Bhandari, S.: Performance evaluation of class balancing techniques for credit card fraud detection, *IEEE* (2017) 2747–2752

41. Husejinovic, A.: Credit card fraud detection using naive bayesian and c4. 5 decision tree classifiers. Husejinovic, A.(2020). Credit card fraud detection using naive Bayesian and C **4** (2020) 1–5

42. Hammed, M., Soyemi, J.: An implementation of decision tree algorithm augmented with regression analysis for fraud detection in credit card. *International Journal of Computer Science and Information Security (IJCSIS)* **18**(2) (2020) 79–88

43. Singh, A., Jain, A.: Adaptive credit card fraud detection techniques based on feature selection method. In: *Advances in computer communication and computational sciences*. Springer (2019) 167–178

44. Lenka, S., Pant, M., Barik, R., Patra, S., Dubey, H.: Investigation into the efficacy of various machine learning techniques for mitigation in credit card fraud detection. In: *Evolution in Computational Intelligence*. Springer (2020) 255–264

45. Brabazon, A., Cahill, J., Keenan, P., Walsh, D.: Identifying online credit card fraud using artificial immune systems, *IEEE* (2010) 1–7

46. Tuo, J., Ren, S., Liu, W., Li, X., Li, B., Lei, L.: Artificial immune system for fraud detection. Volume 2., *IEEE* (2004) 1407–1411

47. Wong, N., Ray, P., Stephens, G., Lewis, L.: Artificial immune systems for the detection of credit card fraud: an architecture, prototype and preliminary results. *Information Systems Journal* **22**(1) (2012) 53–76

48. Akila, S., Reddy, U.S.: Credit card fraud detection using non-overlapped risk based bagging ensemble (nrbe), *IEEE* (2017) 1–4

49. Akila, S., Reddy, U.S.: Risk based bagged ensemble (rbe) for credit card fraud detection, *IEEE* (2017) 670–674

50. Alowais, M.I., Soon, L.K.: Credit card fraud detection: Personalized or aggregated model, *IEEE* (2012) 114–119

51. Mohammed, R.A., Wong, K.W., Shiratuddin, M.F., Wang, X.: Scalable machine learning techniques for highly imbalanced credit card fraud detection: A comparative study, Springer (2018) 237–246

52. Monika, E., Amarpreet Kaur, E.: Fraud prediction for credit card using classification method. Volume 7. (6 2018)

53. Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., Anderla, A.: Credit card fraud detection-machine learning methods. In: *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, *IEEE* (2019) 1–5

54. Chen, R.C., Chen, T.S., Lin, C.C.: A new binary support vector system for increasing detection rate of credit card fraud. *International Journal of Pattern Recognition and Artificial Intelligence* **20**(02) (2006) 227–239

55. Chen, R.C., Chiu, M.L., Huang, Y.L., Chen, L.T.: Detecting credit card fraud by using questionnaire-responded transaction model based on support vector machines, Springer (2004) 800–806

56. Chen, R.C., Luo, S.T., Liang, X., Lee, V.C.S.: Personalized approach based on svm and ann for detecting credit card fraud. Volume 2., *IEEE* (2005) 810–815

57. Chen, R., Chen, T., Chien, Y., Yang, Y.: Novel questionnaire-responded transaction approach with svm for credit card fraud detection, Springer (2005) 916–921

58. Lu, Q., Ju, C.: Research on credit card fraud detection model based on class weighted support vector machine. *Journal of Convergence Information Technology* **6**(1) (2011)

20 No Author Given

59. Tran, P.H., Tran, K.P., Huong, T.T., Heuchenne, C., HienTran, P., Le, T.M.H.: Real time data-driven approaches for credit card fraud detection, *ACM* (2018) 6–9
60. Whitrow, C., Hand, D.J., Juszczak, P., Weston, D., Adams, N.M.: Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery* **18**(1) (2009) 30–55
61. Makki, S., Assaghir, Z., Taher, Y., Haque, R., Hacid, M.S., Zeineddine, H.: An experimental study with imbalanced classification approaches for credit card fraud detection. *IEEE Access* **7** (2019) 93010–93022
62. Wiese, B., Omlin, C.: Credit card transactions, fraud detection, and machine learning: Modelling time with lstm recurrent neural networks. In: *Innovations in neural information paradigms and applications*. Springer (2009) 231–268
63. Van Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., Snoeck, M., Baesens, B.: Apate: A novel approach for automated credit card transaction fraud detection using network-based extensions. *Decision Support Systems* **75** (2015) 38–48
64. Sohony, I., Pratap, R., Nambiar, U.: Ensemble learning for credit card fraud detection, *ACM* (2018) 289–294
65. Sailusha, R., Gnaneswar, V., Ramesh, R., Rao, G.R.: Credit card fraud detection using machine learning. In: *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE (2020) 1264–1270
66. Devi, D., Biswas, S.K., Purkayastha, B.: A cost-sensitive weighted random forest technique for credit card fraud detection. In: *2019 10th international conference on computing, communication and networking technologies (ICCCNT)*, IEEE (2019) 1–6
67. Kumar, M.S., Soundarya, V., Kavitha, S., Keerthika, E., Aswini, E.: Credit card fraud detection using random forest algorithm. In: *2019 3rd International Conference on Computing and Communications Technologies (ICCCT)*, IEEE (2019) 149–153
68. Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S., Jiang, C.: Random forest for credit card fraud detection. (2018) 1–6
69. Lucas, Y., Portier, P.E., Laporte, L., He-Guelton, L., Caelen, O., Granitzer, M., Calabretto, S.: Towards automated feature engineering for credit card fraud detection using multi-perspective hmms. *Future Generation Computer Systems* **102** (2020) 393–402
70. Duman, E., Ozcelik, M.H.: Detecting credit card fraud by genetic algorithm and scatter search. *Expert Systems with Applications* **38**(10) (2011) 13057–13063
71. Ma, H., Li, X.: Application of data mining in preventing credit card fraud, *IEEE* (2009) 1–6
72. Monirzadeh, Z., Habibzadeh, M., Farajian, N.: Detection of violations in credit cards of banks and financial institutions based on artificial neural network and metaheuristic optimization algorithm. *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS* **9**(1) (2018) 176–182
73. Özçelik, M.H., Duman, E., Işık, M., Çevik, T.: Improving a credit card fraud detection system using genetic algorithm, *IEEE* (2010) 436–440
74. Patel, R.D., Singh, D.K.: Credit card fraud detection and prevention of fraud using genetic algorithm. *International Journal of Soft Computing and Engineering (IJSCE) ISSN* (2013) 2231–2307
75. RamaKalyani, K., UmaDevi, D.: Fraud detection of credit card payment system by genetic algorithm. *International Journal of Scientific and Engineering Research* **3**(7) (2012) 1–6
76. Wu, C.H., Tzeng, G.H., Goo, Y.J., Fang, W.C.: A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. *Expert Systems with Applications* **32**(2) (2007) 397–408
77. Aleskerov, E., Freisleben, B., Rao, B.: Cardwatch: A neural network based database mining system for credit card fraud detection, *IEEE* (1997) 220–226

78. Behera, T.K., Panigrahi, S.: Credit card fraud detection using a neuro-fuzzy expert system. In: Computational Intelligence in Data Mining. Springer (2017) 835–843

79. Brause, R., Langsdorf, T., Hepp, M.: Neural data mining for credit card fraud detection, IEEE (1999) 103–106

80. Carneiro, E.M., Dias, L.A.V., da Cunha, A.M., Mialaret, L.F.S.: Cluster analysis and artificial neural networks: A case study in credit card fraud detection, IEEE (2015) 122–126

81. Dorronsoro, J.R., Ginel, F., Sgnchez, C., Cruz, C.S.: Neural fraud detection in credit card operations. IEEE transactions on neural networks **8**(4) (1997) 827–834

82. Fu, K., Cheng, D., Tu, Y., Zhang, L.: Credit card fraud detection using convolutional neural networks, Springer (2016) 483–490

83. Ghosh, S., Reilly, D.L.: Credit card fraud detection with a neural-network. Volume 3., IEEE (1994) 621–630

84. Gómez, J.A., Arévalo, J., Paredes, R., Nin, J.: End-to-end neural network architecture for fraud scoring in card payments. Pattern Recognition Letters **105** (2018) 175–181

85. Arya, M., Sastry G, H.: Deal-‘deep ensemble algorithm’framework for credit card fraud detection in real-time data stream with google tensorflow. Smart Science **8**(2) (2020) 71–83

86. Cheng, D., Xiang, S., Shang, C., Zhang, Y., Yang, F., Zhang, L.: Spatio-temporal attention-based neural network for credit card fraud detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 34. (2020) 362–369

87. Dubey, S.C., Mundhe, K.S., Kadam, A.A.: Credit card fraud detection using artificial neural network and backpropagation. In: 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE (2020) 268–273

88. Carrasco, R.S.M., Sicilia-Urbán, M.Á.: Evaluation of deep neural networks for reduction of credit card fraud alerts. IEEE Access **8** (2020) 186421–186432

89. Gangwar, A.K., Ravi, V.: Wip: Generative adversarial network for oversampling data in credit card fraud detection. In: International Conference on Information Systems Security, Springer (2019) 123–134

90. Gulati, A., Dubey, P., MdFuzail, C., Norman, J., Mangayarkarasi, R.: Credit card fraud detection using neural network and geolocation. Volume 263., IOP Publishing (2017) 42039

91. Guo, T., Li, G.Y.: Neural data mining for credit card fraud detection. Volume 7., IEEE (2008) 3630–3634

92. Kolli, C.S., Tatavarthi, U.D.: Fraud detection in bank transaction with wrapper model and harris water optimization-based deep recurrent neural network. Kybernetes (2020)

93. Jog, A., Chandavale, A.A.: Implementation of credit card fraud detection system with concept drifts adaptation. In: Intelligent Computing and Information and Communication. Springer (2018) 467–477

94. Khan, A.U.S., Akhtar, N., Qureshi, M.N.: Real-time credit-card fraud detection using artificial neural network tuned by simulated annealing algorithm, Citeseer (2014) 113–121

95. Kim, M.J., Kim, T.S.: A neural classifier with fraud density map for effective credit card fraud detection, Springer (2002) 378–383

96. Modi, H., Lakhani, S., Patel, N., Patel, V.: Fraud detection in credit card system using web mining. International Journal of Innovative Research in Computer and Communication Engineering **1**(2) (2013) 175–179

97. Patidar, R., Sharma, L.: Credit card fraud detection using neural network. International Journal of Soft Computing and Engineering (IJSCE) **1**(32-38) (2011)

98. Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., Beling, P.: Deep learning detecting fraud in credit card transactions, IEEE (2018) 129–134

99. Syeda, M., Zhang, Y.Q., Pan, Y.: Parallel granular neural networks for fast credit card fraud detection. Volume 1., IEEE (2002) 572–577

22 No Author Given

100. Lebichot, B., Le Borgne, Y.A., He-Guelton, L., Oblé, F., Bontempi, G.: Deep-learning domain adaptation techniques for credit cards fraud detection. In: INNS Big Data and Deep Learning conference, Springer (2019) 78–88
101. Zhang, X., Han, Y., Xu, W., Wang, Q.: Hoba: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture. *Information Sciences* (2019)
102. Wang, C., Wang, Y., Ye, Z., Yan, L., Cai, W., Pan, S.: Credit card fraud detection based on whale algorithm optimized bp neural network, *IEEE* (2018) 1–4
103. Mahmoudi, N., Duman, E.: Detecting credit card fraud by modified fisher discriminant analysis. *Expert Syst. Appl.* **42**(5) (2015) 2510–2516
104. Olszewski, D.: Fraud detection using self-organizing map visualizing the user profiles. *Knowledge-Based Systems* **70** (2014) 324–334
105. Quah, J.T.S., Sriganesh, M.: Real-time credit card fraud detection using computational intelligence. *Expert Systems with Applications* **35**(4) (2008) 1721–1732
106. Zaslavsky, V., Strizhak, A.: Credit card fraud detection using self-organizing maps. *Information and Security* **18** (2006) 48
107. Bentley, P.J., Kim, J., Jung, G.H., Choi, J.U.: Fuzzy darwinian detection of credit card fraud. Volume 14. (2000)
108. Sánchez, D., Vila, M.A., Cerda, L., Serrano, J.M.: Association rules applied to credit card fraud detection. *Expert Systems with Applications* **36**(2) (2009) 3630–3640
109. Sarno, R., Dewandono, R.D., Ahmad, T., Naufal, M.F., Sinaga, F.: Hybrid association rule learning and process mining for fraud detection. *IAENG International Journal of Computer Science* **42**(2) (2015)
110. Lepoivre, M.R., Avanzini, C.O., Bignon, G., Legendre, L., Piwele, A.K.: Credit card fraud detection with unsupervised algorithms. *Journal of Advances in Information Technology* **7**(1) (2016)
111. Bhusari, V., Patil, S.: Detailed discussion on hidden markov model in credit card fraudulent detection. *Research Journal of Engineering and Technology* **4**(2) (2013) II
112. Falaki, S.O., Alese, B.K., Adewale, O.S., Ayeni, J.O., Aderounmu, G.A., Ismaila, W.O.: Probabilistic credit card fraud detection system in online transactions. *Int. J. Softw. Eng. Appl* **6** (2012) 69–78
113. Khan, A.P., Mahajan, V.S., Shaikh, S.H., Koli, A.B.: Credit card fraud detection system through observation probability using hidden markov model. *International Journal of Thesis Projects and Dissertations (IJTPD)* **1**(1) (2013) 7–16
114. Kumari, N., Kannan, S., Muthukumaravel, A.: Credit card fraud detection using hidden markov model-a survey. *Middle-East Journal of Scientific Research* **19**(6) (2014) 821–825
115. Rani, J.K., Kumar, S.P., Mohan, U.R., Shankar, C.U.: Credit card fraud detection analysis. *International Journal of Computer Trends and Technology* **2**(1) (2011) 24–27
116. Srivastava, A., Kundu, A., Sural, S., Majumdar, A.: Credit card fraud detection using hidden markov model. *IEEE Transactions on dependable and secure computing* **5**(1) (2008) 37–48
117. Abdallah, A., Maarof, M.A., Zainal, A.: Fraud detection system: A survey. *Journal of Network and Computer Applications* **68** (2016) 90–113
118. Zhu, X., Goldberg, A.B.: Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning* **3**(1) (2009) 1–130
119. Campos, G.O., Moreira, E., Meira Jr, W., Zimek, A.: Outlier detection in graphs: A study on the impact of multiple graph models. *Computer Science and Information Systems* **16**(2) (2019) 565–595
120. Kültür, Y., Çağlayan, M.U.: A novel cardholder behavior model for detecting credit card fraud. *Intelligent Automation & Soft Computing* (2017) 1–11
121. West, J., Bhattacharya, M.: Intelligent financial fraud detection: a comprehensive review. *Computers & security* **57** (2016) 47–66

122. Singh, A., Jain, A.: An empirical study of aml approach for credit card fraud detection–financial transactions. *International Journal of Computers Communications & Control* **14**(6) (2020) 670–690

123. Gianini, G., Fossi, L.G., Mio, C., Caelen, O., Brunie, L., Damiani, E.: Managing a pool of rules for credit card fraud detection by a game theory based approach. *Future Generation Computer Systems* **102** (2020) 549–561

124. Mansouri, T., Farasat, A., Menhaj, M.B., Moghadam, M.R.S.: Aro: A new model free optimization algorithm for real time applications inspired by the asexual reproduction. *Expert Systems with Applications* **38**(5) (2011) 4866–4874

125. Farasat, A., Menhaj, M.B., Mansouri, T., Moghadam, M.R.S.: Aro: A new model-free optimization algorithm inspired from asexual reproduction. *Applied Soft Computing* **10**(4) (2010) 1284–1292

126. Salmeron, J.L., Mansouri, T., Moghadam, M.R.S., Mardani, A.: Learning fuzzy cognitive maps with modified asexual reproduction optimisation algorithm. *Knowledge-Based Systems* **163** (2019) 723–735

127. Faust, M.A., Gullledge, R.A.: Identifying harmful marine dinoflagellates. (2002)

128. Holmström, K., Jensen, H.J.: Who runs fastest in an adaptive landscape: sexual versus asexual reproduction. *Physica A: Statistical Mechanics and its Applications* **337**(1-2) (2004) 185–195

129. Prall, F., Ostwald, C.: High-degree tumor budding and podia-formation in sporadic colorectal carcinomas with k-ras gene mutations. *Human pathology* **38**(11) (2007) 1696–1702

130. Lee, K., Singh, P., Chung, W.C., Ash, J., Kim, T.S., Hang, L., Park, S.: Light regulation of asexual development in the rice blast fungus, *magnaporthe oryzae*. *Fungal Genetics and Biology* **43**(10) (2006) 694–706

131. Song, W.: Morphogenesis of *cylindrocapsa tetraoculata* (ciliophora, hypotrichia, oxytrichidae) during binary fission. *European Journal of Protistology* **40**(3) (2004) 245–254

132. Akila, S., Reddy, U.S.: Cost-sensitive risk induced bayesian inference bagging (ribib) for credit card fraud detection. *Journal of computational science* **27** (2018) 247–254

133. Ghobadi, F., Rohani, M.: Cost sensitive modeling of credit card fraud using neural network strategy. In: 2016 2nd international conference of signal processing and intelligent systems (ICSPIS), IEEE (2016) 1–5