A Genetic Programming Approach for Fraud Detection in Electronic Transactions

Carlos A. S. Assis
PPGMMC
Centro Federal de Educação Tecnológica de Minas Gerais
Belo Horizonte – MG, Brazil
Email: carlosassis@dppg.cefetmg.br

Adriano C. M. Pereira
DCC
Universidade Federal de Minas Gerais
Belo Horizonte – MG, Brazil
Email: adrianoc@dcc.ufmg.br

Marconi A. Pereira
DETCH
Universidade Federal de São João Del Rei
Ouro Branco – MG, Brazil
Email: marconi@ufsj.edu.br

Eduardo G. Carrano
DEE
Universidade Federal de Minas Gerais
Belo Horizonte – MG, Brazil
Email: egcarrano@ufmg.br

***Abstract*—The volume of online transactions has increased considerably in the recent years. Consequently, the number of fraud cases has also increased, causing billion dollar losses each year worldwide. Therefore, it is mandatory to employ mechanisms that are able to assist in fraud detection. In this work, it is proposed the use of Genetic Programming (GP) to identify frauds (charge back) in electronic transactions, more specifically in online credit card operations. A case study, using a real dataset from one of the largest Latin America electronic payment systems, has been conducted in order to evaluate the proposed algorithm. The presented algorithm achieves good performance in fraud detection, obtaining gains up to 17% with regard to the actual company baseline. Moreover, several classification problems, with considerably different datasets and domains, have been used to evaluate the performance of the algorithm. The effectiveness of the algorithm has been compared with other methods, widely employed for classification. The results show that the proposed algorithm achieved good classification effectiveness in all tested instances.**

I. Introduction

A recent report of CyberSource [1] has estimated that the fraud cost for online retailers is around 3.5 billion dollars or 0.9% percent of the online revenue per year. This is a significant amount that should be part of the retailer profits, instead of becoming part of the business cost. Unfortunately, the fraud rate increases when the retailer supports mobile commerce or provides international shipping.

Considering that the fraud in electronic commerce has been increasing consistently and it represents significant losses for business, the prevention and detection of frauds became a key factor to the success of the electronic markets. There is a number of challenges in handling frauds, such as the different characteristics of fraudsters and the huge amount of data to be analyzed.

This work proposes the use of Genetic Programming (GP) for evaluating online electronic transactions, especially credit card payments, and to identify fraudulent transactions. The choice for GP is justified by its efficiency in problems with large search spaces and its flexibility [2].

The proposed solution has been designed with focus on online payment and, more specifically, in companies that work as proxy for transactions between the sellers and the buyers. These companies usually receive a small transaction percentage to intermediate the whole payment process for the store. However, if the transaction is a fraud, the company is the entire responsible by the loss and, consequently, the refund. This kind of service is provided by companies such as PayPal[1] and UOL PagSeguro[2].

The datasets used for validation contain information of electronic transactions for the period of April 2011. These datasets have 12 attributes, which can be numerical, categorical and boolean. The number of records is bigger than 100,000 per dataset. The datasets were provided from a Brazilian electronic payment company, UOL PagSeguro, with cooperation of the fraud analysis department of the company.

The proposed algorithm was also tested on three datasets from the UCI Machine Learning Repository [3]: Iris Dataset, Cancer Dataset, Hepatitis Dataset. These datasets have been chosen because they are extensively studied in the literature and it was possible to compare the proposed algorithm. However, it should be noticed that such a tool has been developed with focus on fraud detection application and, therefore, it is not necessarily tuned to handle with other applications.

The main contributions of this work are:

(a) Proposal of a methodology based on Genetic Programming (GP), design and implementation of this method as a decision support tool to detect fraud in electronic transactions.
(b) Validation and analysis of the method in real scenarios, such as in the UOL PagSeguro datasets.
(c) Implementation of this tool as an open-source framework[3], available under the GNU General Public License [4].

[1] http://www.paypal.com/

[2] http://www.pagseguro.uol.com.br/

[3] https://code.google.com/p/data-mining-genetic-programming/

The remainder of this article is organized as follows. Some recent related works are described in section II. The proposed methodology is presented in section III. Such an approach is applied to real fraud datasets and UCI datasets in section IV. Finally, conclusions and directions for future improvement are given in section V.

II. Related Work

Some works that are related with fraud detection in credit card operations are described in this section.

Bent et al. [5] presented a Fuzzy Darwinian Detection system that used genetic programming for evolving fuzzy logic rules. It classified the transactions into suspicious and non-suspicious. The approach presented high accuracy and produced 5.79% of false negatives, which is relatively small. A total of 4000 transactions from January 1995 to December 1995 of a domestic credit card company were used on tests. This dataset was composed of 96 attributes, being 62 used on tests.

Bolton and Hand [6] proposed an unsupervised detection technique based on breakpoint analysis to identify changes in spending pattern. A breakpoint is an observation or time in which anomalous behavior is detected. An example of a possible fraudulent behavior is a sudden increase in the number of transactions. An advantage of breakpoint analysis is that it does not require balanced data. The approach was in early development stage and the results presented were restricted to simple examples.

In [7] the authors employed Support Vector Machines (SVM) and Artificial Neural Networks (ANN) to investigate the time-varying fraud problem. The results showed that ANN outperforms SVM in terms of training accuracy but it is not so efficient for predicting future data, what is probably caused by data overfitting.

Guo [8] modeled a sequence of operations in credit card transaction using a confidence-based neural network. The receiver operating characteristic (ROC) analysis was introduced to ensure the accuracy of the fraud detection mechanism. Initially, the confidence-based neural network was trained with synthetic data. If an incoming credit card transaction was not accepted by the trained neural network model (NNM) with sufficiently low confidence, it was considered to be fraudulent. In this study 7,000 synthetic data were used for training and 3,000 were used for testing. The system obtained 91.2% of true positives and 13.4% of false negatives. The author claimed that the proposed classifier is well suited for credit card fraud detection.

Brabazon et al. [9] investigated the effectiveness of Artificial Immune Systems (AIS) for credit card fraud detection using a large dataset obtained from an online retailer. Three AIS algorithms were implemented and their performances were compared against a logistic regression model. The results of the Unmodified Negative Selection Algorithm were promising only for Accuracy. The Modified Negative Selection Algorithm offered good trade-off between classifying self and non-self correctly. The results of the Clonal Selection Algorithm were very unpromising in relation to the accuracy metric used. The results suggested that the AIS can be applied in this problem, but the proposed algorithms require improvements.

Duman and Ozcelik [10] proposed a method that combines genetic algorithms and scatter search. The method was applied to real data. Comparing with current practices of a large Turkish bank, the obtained results seemed to be impressive: the authors were able to improve the bank existing fraud detection strategy in 200%.

Soltani et al. [11] presented a new model based on Artificial Immune Systems for credit card fraud detection based on user behavior. The model brought together two methodologies of fraud detection, namely tracking account behavior and general threshold. The system generates normal memory cells based on normal user transactions and fraud memory cells based on known fraudulent records. Although AIS requires long time for being trained, the authors report good results. They suggested that the implementation of the model in cloud computing systems could reduce the training time.

Li and Wong [12] compared the performance of six different data mining techniques for solving the China Corporate Securities Fraud (CCSF) problem. The authors identified that the Grammar Based Genetic Programming (GBGP) outperformed Logistic Regression Model (LRM), ANN, Sequential Minimal Optimization (SMO), Radial Basis Function Networks (RBF) and Bayesian Networks (BN) in terms of accuracy.

Finally, it is possible to find some works that present comprehensive reviews about this subject [13], [14], [15]. Among all reviews, it was not possible to find works that combine genetic programming and fraud detection in electronic transactions.

When compared to the above cited papers, this work is complementary. The technique implemented, although widely known, was not exploited to detect credit card frauds. Furthermore, this work is applied to real unbalanced data[4]. The unbalance ratio of the set considered is three to one thousand.

The developed algorithm is based on GP and it presents the following features: (a) a fitness function that is suitable for unbalanced problems; (b) the framework provides access to SQL databases using Java Database Connectivity; (c) it performs economical evaluation of the classifier performance (based on the concept of Economic Efficiency – EE); (d) it is an open-source framework.

It should be noticed that GP has the advantage of encoding rules directly as computer programs, avoiding time-costly rule interpretation procedures. In the specific case of this work, the rules are built using SQL-based [17] WHERE-clause directly.

The authors were not able to find a fraud detection algorithm that addresses and combines all the features presented in this work.

III. Methodology

The approach proposed in this paper is described along this section. It is composed of three main steps, as shown in Figure 1:

- **Data Preparation**: in this step the data is prepared for being used. Such a preparation consists on extracting

[4] Datasets are unbalanced when at least one class is represented by only a small number of training examples while the other class(es) make up the majority [16]

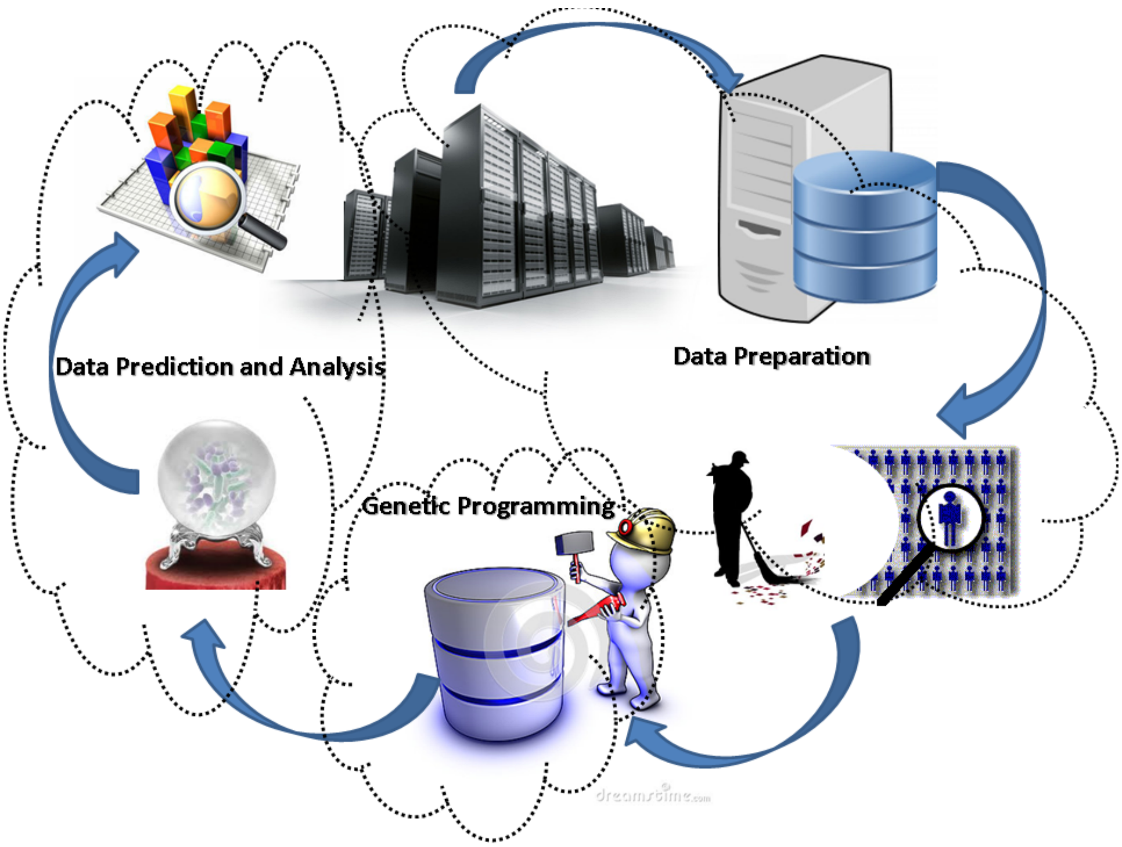

Fig. 1: Proposed methodology

parts from the database, loading them to a local database in order to clean, integrate, transform and create partitions. This is an important step, since it makes possible to focus only on relevant data. Real-world data may be incomplete, noisy, and inconsistent, what makes knowledge discovery more difficult. Therefore, data preparation is highly recommend to generate good classification rules [18].

- **Genetic Programming algorithm**: a population of random classification rules (GP individual) is generated at first. Each rule is evaluated in order to receive a fitness value, in such a way that better rules have more chances of survival. At each iteration, the rules are subjected to reproduction, crossover and mutation, in order to obtain better classification models. The process stops when the maximum number of generations is reached.

- **Data Prediction and Analysis**: in this step the obtained models (classification rules) are used to classify l to the new data tuples. It is necessary to estimate the generality capacity of the classification rules achieved. Such a process is usually referred as validation.

The main parts of the proposed approach are described in the next subsections.

A. Genetic Programming

A framework based on the Genetic Programming evolutionary algorithm [19], [20] has been proposed to build the classification rules. A big advantage of GP is its flexibility, since it is possible to change the individual representation in order to deal with any kind of problem [21]. In this work the individuals are represented as WHERE clauses of the SQL query.

At first, it is important to ensure that the data set is composed only by non-repeated data. Otherwise, the classification tool can present a false improvement in their performance, based on a memorizing a subset of repeated data.

Before applying the algorithm, it is necessary to separate the data into training and validation blocks. The training set is used to provide information about the class behaviors in order to build the classification rules. The validation set is necessary to evaluate the generalization ability of the obtained models. There are different approaches that could be used for data splitting. The holdout method is one of the simplest ways of performing cross validation [22]. In this approach the dataset is separated in two sets, so-called training set and testing set. K-fold cross validation is one way to improve the traditional holdout method. The dataset is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other $k-1$ subsets are used for training. The average error across all k trials is used to evaluate the model.

In this work, the K-fold cross validation method was used for the UCI datasets, since it is the most common choice in the literature. For the fraud datasets, holdout validation method was used. This is the method that simulates the actual procedure of the companies and it is suitable for handling with time dependent data.

The GP algorithm is composed of four main steps: (i) initialization; (ii) evaluation; (iii) selection; (iv) crossover and mutation. The algorithm flow is illustrated in Algorithm 1. In this algorithm, the parameters r_c, r_m, max_ger and p_{size} are crossover rate, mutation rate, maximum number of generations (stop criterion) and population size, respectively. The variables X and S are respectively the population and the selected individuals from the selection step.

Algorithm 1 Pseudocode for Genetic Programming.

```
procedure PG($r_c, r_m, max_{ger}, p_{size}$)
    $\mathbf{X} \leftarrow$ initializePopulation($p_{size}$)
    $ger \leftarrow 1$
    while $ger < max_{ger}$ do
        $\mathbf{X} \leftarrow$ evaluatePopulation($\mathbf{X}$)
        while $newPopulation < p_{size}$ do
            $\mathbf{S} \leftarrow selectOperator(\mathbf{X})$
            $\mathbf{S} \leftarrow crossoverOperator(\mathbf{S}, r_c)$
            $\mathbf{S} \leftarrow mutationOperator(\mathbf{S}, r_m)$
            $newPopulation \leftarrow add(\mathbf{S})$
        end while
        $\mathbf{X} \leftarrow newPopulation$
        $ger \leftarrow ger + 1$
    end while
end procedure
```

1) Structure of individuals: GP individuals are usually coded as parse trees, in which leaves correspond to terminal symbols (variables and constants) and internal nodes correspond to nonterminals (operators and functions). The set of all the nonterminal symbols allowed is called the function set, whereas the terminal symbols allowed constitute the terminal set.

Two conditions must be satisfied to ensure that GP can be successfully applied to a specific problem: sufficiency and closure. Sufficiency states that the terminals and nonterminals (in combination) must be capable of representing a solution to the problem. Closure requires that each function of the nonterminal set should be able to handle all values it might receive as input.

The population is a set of individuals that, in this work, are SQL predicates. The individual is modeled to represent a rule that should be applied to pattern selection in a database, as a WHERE clause of a SQL query. The terminal set includes the dataset attributes and values inside their corresponding domains. Nonterminals can be boolean operators (AND, OR, NOT) and attribute comparison operators ($>$, $>=$, $<$, $<=$, $=$, $! =$). Moreover, other operators supported by SQL could be eventually considered for nonterminals as well.

An example is shown in Figure 2. It is assumed that the GP follows the Michigan approach i.e, an individual represents a single rule [23].

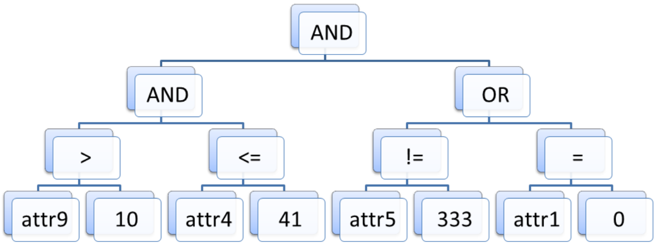

Fig. 2: Representation of the individual

2) Initialization of GP: At first, it is necessary to build random trees that are syntactically valid. The main parameters of initialization method are the maximum number of nodes and the maximum depth, which are used to control the complexity of the solutions of the initial population. The grow method, proposed in [19], has been used in this work to generate the solutions. Initially, a function node is chosen at random to be the root. At each iteration of the grow method, either a function or a terminal node is selected and included in the tree. The process stops when the maximum depth or the maximum number of nodes is reached.

3) Fitness function: The fitness of an individual is assigned based on its performance using one or more metrics, such as accuracy, sensitivity and specificity. The values for such metrics are evaluated based on the confusion matrix. In the confusion matrix [24], the main diagonal contains the correct classifications performed by the classification model, while the sum of the elements outside this diagonal is the number of errors. The number of true positives (TP) is the number of tuples that are classified by the individual and that belong to its covered class. The number of false positive (FP) is calculated by counting the total number of tuples obtained by the rule codified by the individual that do not belong to the class predicted by the rule. The number of true negatives (TN) is calculated by counting the number of tuples that do not belong to the class in question, subtracted by the number of FP. The number of false negatives (FN) is calculated by counting the number of tuples belonging to the class in question, less the number of TP. The dimension of the confusion matrix is defined by the number of problem classes (2×2 for two classes, 3×3 for three classes, and so fourth).

In this work, the fitness value is assigned to each individual based on Equation 1.

$$f_1(I, X) = Acur(I, X) \times Sens(I, X) \times Spec(I, X) \quad (1)$$

in which:

- $f_1(I, X)$ is the fitness value and I is an individual that is employed to identify patterns belonging to class X;
- $Acur(I, X) = \frac{(TP(I,X)+TN(I,X))}{(TP(I,X)+TN(I,X)+FP(I,X)+FN(I,X))}$ measures the overall accuracy of the model; $Sens(I, X) = \frac{(TP(I,X))}{(TP(I,X)+FN(I,X))}$ is conditional

probability of correctly identifying the true-positives subjects;

- $Spec(I,X) = \frac{(TN(I,X))}{(TN(I,X)+FP(I,X))}$ is conditional probability of correctly identifying the true-negatives subjects.

4) Genetic operators: The selection of individuals is performed through binary tournament based on fitness value (f_1). The individuals that remain after selection are subjected to crossover and mutation, in order to build the offspring population.

Each pair of individuals of the population can suffer crossover based on a given crossover probability. If the crossover is to be performed, then a cut point is chosen at random in each one of the parent individuals and the subtrees after these points are swapped. The Figure 3 illustrates the crossover procedure.

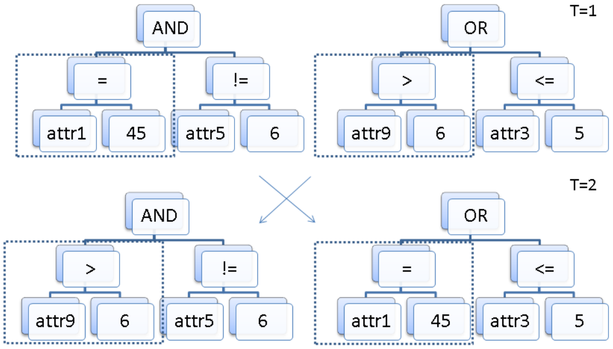

Fig. 3: Crossover example.

After crossover, each individual can be mutated, based on a given mutation probability. The mutation operator randomly chooses a function node (except the root) and replaces it with a new randomly generated sub-tree.

There are four possible outcomes for the mutation operator [23]:

- **Point Mutation**: a terminal node is replaced by another terminal node.
- **Collapse Mutation**: a terminal node replaces a function node (subtree);
- **Expansion Mutation**: a function node replaces a terminal node.
- **Subtree Mutation**: a function node replaces another function node.

It is important to take some care during the mutation process, since it can easily lead to unfeasible rules. In this sense, the mutation operator cannot replace a node (or a subtree) by a node of a different type [25]. Figure 4 illustrates a example of Subtree Mutation. Note that the new node returns the same data type (Boolean) of the old one.

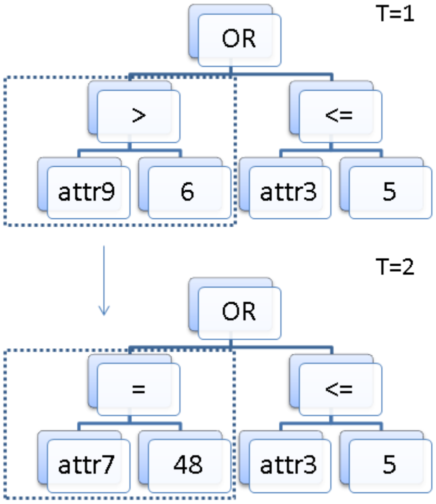

Fig. 4: Mutation example.

B. *Prediction*

After executing the GP algorithm, it is necessary to employ some strategy to use the classification rules achieved (GP final population). These rules should be combined to build some kind of classification committee.

The rule set is used to predict the class of each record. The best rules for each class (e.g., fraud or non-fraud) are used as a voting system to classify each dataset pattern. Such a voting system generates a score, which is calculated based on Algorithm 2.

In this approach, a sample “a” is assigned to the class A when the number of rules of class A that select “a” is higher than the number of rules of other classes that also select “a”. For instance, if one record is selected by three rules of class “fraud”, one rule of class “non-fraud”, then the record is classified as class “fraud”.

It is expected that the transactions with higher score for fraud have higher probability of being fraud.

Algorithm 2 Pseudocode for calculate score.

```
procedure SCORE($bestRulesF, bestRulesNF$)
    $numRulesF \leftarrow$ matchingRules($bestRulesF$)
    $numRulesNF \leftarrow$ matchingRules($bestRulesNF$)
    $scoreF \leftarrow numRulesF/bestRulesF.size()$
    $scoreNF \leftarrow numRulesNF/bestRulesNF.size()$
    $score \leftarrow scoreF - scoreNF$
end procedure
```

Then the records or transactions are sorted in decreasing score order, generating a ranking. Using the Equation (2), called Economic Efficiency (EE) [26], it is possible to evaluate the economical result for each transaction, considering its class (fraud or non-fraud).

This economical analysis allows to evaluate the ranking position that should maximize the profit. In other words, it identifies the ranking position that identifies the highest amount of frauds, besides minimizing the rate of good transactions that would be blocked.

$$EE = \sum_{j=1}^{n} [G_j \cdot r - L_j \cdot (1 - r)] \qquad (2)$$

The Gain (G) represents the financial value of the true positive transactions; the Loss (L) is the financial value of false negative transactions; rate (r) is the commission of the company for intermediating the transaction; n is the number of transactions that are being analyzed. Rate r has been set as 3%, which is an average value used by proxy corporations.

This indicator is important because Precision (fraction of correctly retrieved instances [27]) and Recall (fraction of correctly retrieved instances out of all matching instances [27]) are not the best metrics to evaluate the performance of the rules. This concept of Economic Efficiency has been proposed and validated jointly with fraud specialists of UOL PagSeguro.

Table I exemplifies the proposed strategy, considering the ranking of transactions and the economic evaluation, for a hypothetical set of 10 transactions. In Table Ia the transactions are disposed in the same order in which they were provided by the database (the fraud transactions are colored with red background for improving readability). In Table Ib the the transactions sorted by the Fraud Score. In this example, the transaction with Id 3 (highlight with blue color) would be the cutting point for maximizing the EE. Note that four transactions would be correctly classified as fraud and only one non-fraud would be incorrectly labeled.

TABLE I: Example of fraud score ranking for 10 transactions. “Id” is a transaction identifier, “F” is a label that indicates if the transaction is a fraud or not, “FP” is the fraud score and “NFP” is the non-fraud score [28].

(a)

Ordered Transactions			
Id	**F.**	**FP**	**NPF**
1	0	0.20	0.80
2	0	0.40	0.60
3	1	0.45	0.55
4	1	0.47	0.53
5	0	0.60	0.40
6	0	0.40	0.60
7	0	0.30	0.70
8	1	0,99	0.01
9	0	0.25	0.75
10	1	0.95	0.05

(b)

Ranked Transactions			
Id	**F.**	**FP**	**NPF**
8	1	0.99	0.01
10	1	0.95	0.05
5	0	0.60	0.40
4	1	0.47	0.53
3	1	0.45	0.55
2	0	0.40	0.60
6	0	0.40	0.60
7	0	0.30	0.70
9	0	0.25	0.75
1	0	0.20	0.80

Equation 3 provides a relative gain of the given method. This value can vary from negative values to 100%, which means that all transactions are correctly classified. In this scale, $EE_{\%} = 0\%$ is exactly the current situation of the company. This equation is used to assess the performance of the proposed algorithm when compared to the real situation.

$$EE_{\%} = \frac{EE - EE_{Real}}{EE_{Max} - EE_{Real}} \cdot 100 \qquad (3)$$

The EE_{Max} (Equation 3) is the maximum gain that the company could have if all frauds are detected and all valid transactions are accepted.

$$EE_{Max} = \sum_{j=1}^{n} (G_j \cdot r) \qquad (4)$$

The EE_{Real} is the current EE of the company.

IV. Experiments - Case Study

Two groups of problems have been used to validate the proposed methodology:

- Three datasets from UCI Machine Learning repository [3]: Breast Cancer Wisconsin, Hepatitis and Iris. The Breast Cancer Wisconsin dataset consists of 699 instances taken from Fine Needle Aspirates (FNA) of human breast tissue. Each record in the database has nine attributes, which are integer values between 1 and 10. Each sample is associated to one of two classes (label): benign or malignant. The class has a distribution of 65% benign samples and 35% malignant samples. Hepatitis dataset contains 155 samples belonging to two different target classes: to die or to live. There are 19 attributes, 13 binary and 6 attributes with 6-8 discrete values. Iris dataset consists on 150 instances from three species of the Iris flower: setosa, virginica and versicolor. The length and width of sepal and petals are measured for each sample. The objective behind using UCI Machine Learning datasets is to show that the algorithm can be adapted to different applications, although it has been designed to work with online credit card transactions. It is not the intention of the authors to present a new state of art method for such datasets.
- Two datasets from a Brazilian electronic payment company. Some attributes available for each transaction have been selected to build the classification rules. Due to a confidentiality agreement, it is not possible to provide many details about the dataset, but the cases considered involve up to hundred thousand transactions.

Preliminary tests have been performed to set up the algorithm. These tests have leaded to the parameter set of Table II.

TABLE II: GP parameters

Mutation rate	Crossover rate	Generations	Population size
2.5%	75%	100	500

A. UCI Machine Learning Datasets

In the UCI instances, each database has been split into 10 partitions, for 10-fold cross-validation. The results achieved for accuracy in the UCI datasets are presented in Table III. This table shows the mean (±) standard error of the accuracy.

Some literature results can be used as benchmark on the UCI datasets (Tables IV, V and VI). It is possible to note

TABLE III: Results for UCI datasets

Cancer	Hepatitis	Iris
96.0 ± 0.006	86.0 ± 0.0137	94.0 ± 0.008

that the performance of the proposed algorithm (bold face accuracy) is comparable to the best results shown in the literature.

TABLE IV: Results - Comparisons for Cancer Dataset. Adapted from [29].

Technique	Accuracy
SMO	97.7
Linear discreet analysis	96.8
GP	**96.0**
Hybrid Approach	95.9
Supervised fuzzy clustering	95.5
Neuron-fuzzy	95.0
C4.5	94.7
CART with feature selection (Chisquare)	92.6

TABLE V: Results - Comparisons for Hepatitis Dataset. Adapted from [30].

Technique	Accuracy
PCA-ANN	89.6
15NN, stand. Euclidean	89.0
FSM without rotations	88.5
LDA	86.4
Nave Bayes and semiNB	86.3
GP	**86.0**
IncNet	86.0
QDA, quadratic discriminant analysis	85.8
1NN	85.3

TABLE VI: Comparison for Iris dataset. Adapted from [31].

Technique	Accuracy
NNGE Classifier	96.0
GP	**94.0**
JRIP Classifier	94.0
PART Classifier	94.0
RIDOR Classifier	94.0
OneR Classifier	94.0
Decision Table Classifier	92.6
DTNB Classifier	92.0
Conjunctive Rule Classifier	66.6
ZeroR Classifier	33.3

B. Fraud Datasets

The fraud datasets have been split into training and test sets, as shown in Table VII. In some datasets, the undersampling technique has been applied [32]. Undersampling is a technique commonly employed to adjust the distribution of the classes in a dataset. The Economic Efficiency of a rule is estimated by its classification performance on the test set.

TABLE VII: Division training/test fraud datasets.

ID	Attributes	Training (%)	Test (%)	Undersampling
1	12	75.814	24.186	
2	12	3.643	96.357	1/1

The whole experiment has been executed 20 times for each dataset. The performances achieved are presented in Table VIII. This table shows the mean (±) standard error of the Economic Efficiency. Note that the technique obtains gains in both datasets ($EE > 0\%$, means a gain with regard to the company baseline). In the worst case, the GP reached a gain close to 15%. It means that it would be possible to save $150 for every $1,000 in refunds, what is very significant.

TABLE VIII: Results - Genetic Programming Approach

ID	GP
1	17.6% ± 0,0391
2	15.69% ± 0,0459

V. Conclusion

The number of fraud cases in electronic transactions has increased considerably in the recent years. This paper proposes a framework based on genetic programming to perform fraud detection in electronic transactions. The candidate solutions (individuals) are represented as SQL WHERE-clause, which identify samples of one of the two possible classes (fraud or non-fraud). The suitability of the designed algorithm has been estimated based on popular datasets from the literature and in two real scenarios of one of the largest Latin American electronic payment companies.

The main contribution of the proposed work is the design and implementation of this technique of genetic programming as a framework. Despite there are many articles related to the topic, none of them provides all features implemented in this framework, which will be available online as an open-source tool. Moreover, the work has presented good results for the fraud detection problems. Additionally, the algorithm has shown good results when it was applied to the UCI Machine Learning dataset. This fact indicates the generality and robustness of the approach, which can be easily adapted to handle with different domains and applications.

As a future work, it is suggested the extension of the algorithm to a multiobjective scenario, in order to consider rule complexity as one of the objectives jointly with the Economic Efficiency. It is also planned to consider new datasets with additional information that is not yet available. It is expected to reach even better results with these datasets, provided that an adequate feature selection method is employed.

Acknowledgements

The authors would like to thank the CEFET-MG and the Brazilian agencies CAPES, CNPq and FAPEMIG by the financial support.

References

[1] CyberSource. (2014, January) Cybersource corporation. CyberSource. Accessed: January 2014. [Online]. Available: http://www.cybersource.com/

[2] C. C. Bojarczuk, H. S. Lopes, A. A. Freitas, and E. L. Michalkiewicz, "A constrained-syntax genetic programming system for discovering classification rules: Application to medical data sets," *Artif. Intell. Med.*, vol. 30, no. 1, pp. 27–48, Jan. 2004. [Online]. Available: http://dx.doi.org/10.1016/j.artmed.2003.06.001

[3] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[4] GNU. (2014, January) Gnu project. GNU. Accessed: January 2014. [Online]. Available: https://www.gnu.org/licenses/

[5] P. J. Bentley, J. Kim, G. H. Jung, and J. U. Choi, "Fuzzy Darwinian Detection of Credit Card Fraud," in *14th Annual Falll Symposium of the Korean Information Processing Society*, 2000. [Online]. Available: http://www.cs.ucl.ac.uk/staff/P.Bentley/BEKIJUCHC1.pdf

[6] R. J. Bolton and D. J. Hand, "Unsupervised profiling methods for fraud detection," *Conference on Credit Scoring and Credit Control*, September 2001.

[7] R.-C. Chen, S.-T. Luo, X. Liang, and V. C. S. Lee, "Personalized approach based on svm and ann for detecting credit card fraud," in *Neural Networks and Brain, 2005. ICNN B '05. International Conference on*, vol. 2, 2005, pp. 810–815.

[8] T. Guo, "Neural data mining for credit card fraud detection," in *International Conf. on Machine Learning and Cybernetics*, vol. 7, July 2008.

[9] A. Brabazon, J. Cahill, P. Keenan, and D. Walsh, "Identifying online credit card fraud using artificial immune systems," in *IEEE Congress on Evolutionary Computation*, 2010, pp. 1–7.

[10] E. Duman and M. H. Ozcelik, "Detecting credit card fraud by genetic algorithm and scatter search," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13 057–13 063, Sep. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2011.04.110

[11] N. Soltani, M. Akbari, and M. Javan, "A new user-based model for credit card fraud detection based on artificial immune system," in *6th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, 2012, pp. 029–033.

[12] H.-B. Li and M.-L. Wong, "Knowledge discovering in corporate securities fraud by using grammar based genetic programming," *Journal of Computer and Communications*, vol. 2, pp. 148–156, March 2014.

[13] E. Ngai, Y. Hu, Y. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decision Support Systems*, vol. 50, no. 3, pp. 559 – 569, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167923610001302

[14] A. Sharma and P. K. Panigrahi, "Article: A review of financial accounting fraud detection based on data mining techniques," *International Journal of Computer Applications*, vol. 39, no. 1, pp. 37–47, February 2012, published by Foundation of Computer Science, New York, USA.

[15] P. Espejo, S. Ventura, and F. Herrera, "A survey on the application of genetic programming to classification," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 40, no. 2, pp. 121–144, 2010.

[16] U. Bhowan, M. Johnston, and M. Zhang, "Evolving ensembles in multi-objective genetic programming for classification with unbalanced data," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '11. New York, NY, USA: ACM, 2011, pp. 1331–1338. [Online]. Available: http://doi.acm.org/10.1145/2001576.2001756

[17] J. Sayles, *SQL for DB2 and SQL/DS application developers*. QED Information Services, 1989.

[18] S. Zhang, C. Zhang, and Q. Yang, "Data preparation for data mining," *Applied Artificial Intelligence*, vol. 17, no. 5-6, pp. 375–381, 2003. [Online]. Available: http://dx.doi.org/10.1080/713827180

[19] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[20] W. Banzhaf, F. D. Francone, R. E. Keller, and P. Nordin, *Genetic Programming: An Introduction: on the Automatic Evolution of Computer Programs and Its Applications*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998.

[21] J. P. Rosca and D. H. Ballard, "Genetic programming with adaptive representations," University of Rochester, Rochester, NY, USA, Tech. Rep., 1994.

[22] P. R. Cohen, *Empirical Methods for Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1995.

[23] A. A. Freitas, *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2002.

[24] F. Provost and R. Kohavi, "Guest editors&lsquo; introduction: On applied research in machinelearning," *Mach. Learn.*, vol. 30, no. 2-3, pp. 127–132, Feb. 1998. [Online]. Available: http://dx.doi.org/10.1023/A:1007442505281

[25] M. A. Pereira, C. Davis Jnior, and J. Vasconcelos, "A niched genetic programming algorithm for classification rules discovery in geographic databases," in *Simulated Evolution and Learning*, ser. LNCS. Springer Berlin Heidelberg, 2010, vol. 6457, pp. 260–269. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-17298-4_27

[26] R. A. F. Lima and A. C. M. Pereira, "Fraud detection in web transactions," in *Proceedings of the 18th Brazilian symposium on Multimedia and the web*, ser. WebMedia '12. New York, NY, USA: ACM, 2012, pp. 273–280. [Online]. Available: http://doi.acm.org/10.1145/2382636.2382695

[27] A. A. Freitas, "Evolutionary computation," in *Handbook of Data Mining and Knowledge Discovery*, W. Klosgen and J. Zytkow, Eds. Oxford University Press, 2002, ch. 32, pp. 698–706. [Online]. Available: http://citeseer.ist.psu.edu/460298.html

[28] E. Caldeira, G. Brandao, and A. Pereira, "Characterizing and preventing chargebacks in next generation web payments services," in *Computational Aspects of Social Networks (CASoN), 2012 Fourth International Conference on*, Nov 2012, pp. 333–338.

[29] G. I. Salama, M. Abdelhalim, and M. A.-e. Zeid, "Breast cancer diagnosis on three different datasets using multi-classifiers," *International Journal of Computer and Information Technology*, vol. 32, no. 569, p. 2, 2012. [Online]. Available: http://ijcit.com/archives/volume1/issue1/Paper010105.pdf

[30] T. A. Jilani, H. Yasin, and M. M. Yasin, "Article: Pca-ann for classification of hepatitis-c patients," *International Journal of Computer Applications*, vol. 14, no. 7, pp. 1–6, February 2011, published by Foundation of Computer Science.

[31] C. L. Devasena, T. Sumathi, V. Gomathi, and M. Hemalatha, "Effectiveness evaluation of rule based classifiers for the classification of iris data set," *Bonfring International Journal of Man Machine Interface*, vol. 1, no. Special Issue, pp. 05–09, 2011. [Online]. Available: http://www.journal.bonfring.org/papers/mmi/volume1/BIJMMI-01-1002.pdf

[32] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 2, pp. 539–550, April 2009.