“A machine learning based credit card fraud detection using the GA algorithm for feature selection”

Authors: Emmanuel Ileberi1, Yanxia Sun1 and Zenghui Wang2*
Paper: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00573-8

Dataset: Credit Card (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud)

This paper proposes a machine learning (ML) based credit card fraud detection engine using the genetic algorithm (GA) for **feature selection**. After the optimized features are chosen, the proposed detection engine uses the following ML classifiers: Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Artificial Neural Network (ANN), and Naive Bayes (NB).

To solve the issue of a high feature dimension space, we implement a feature selection algorithm that is
based on the Genetic Algorithm (GA) [25] using the **RF method in its fitness function**. The RF method is used in the GA fitness function because it can handle a large number of input variables, it can **automatically handle missing values, and because it is not affected by noisy data.**

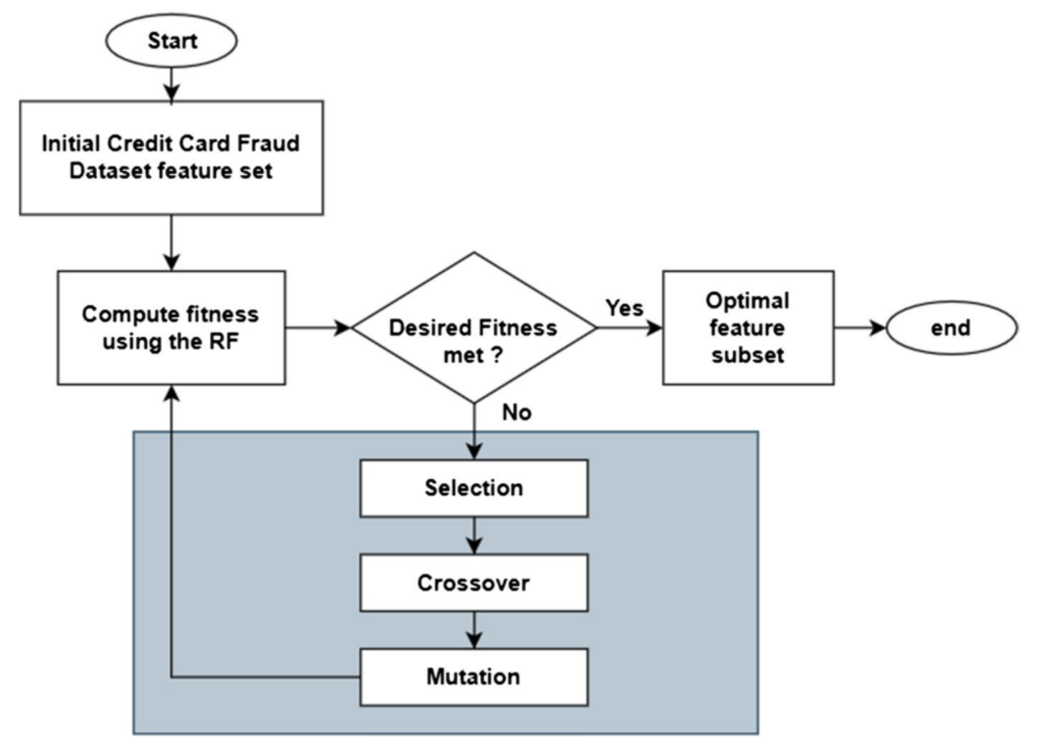

Fig. 2 GA flowchart

Table 1 GA Selected features

Attribute vector	Vector length	Attribute list
v_1	18	V1, V5, V7, V8, V11,V13, V14, V15, V16, V17, V18, V19, V20, V21, V22, V23, V24, Amount
v_2	9	V1, V6, V13, V16, V17, V22, V23, V28, Amount
v_3	13	V2, V11, V12, V13, V15, V16, V17, V18,V20, V21, V24, V26, Amount
v_4	9	V2, V7, V10, V13,V15, V17, V19, V28, Amount
v_5	13	Time, V1, V7, V8, V9, V11, V12, V14, V15, V22, V27, V28, Amount

After the implementation of the GA (Algorithm 1 and Algorithm 2) on the credit card fraud dataset, we obtained the 5 optimal feature vectors (v_1 to v_5) that are shown in Table 1. These vectors contain the feature names that represents the most optimal attributes that will be used to assess the effectiveness of our proposed method.

Fraud detection framework(Proposed):

- Data Normalize by **MinMax Scalling**
- Data Balance **using SMOTE**

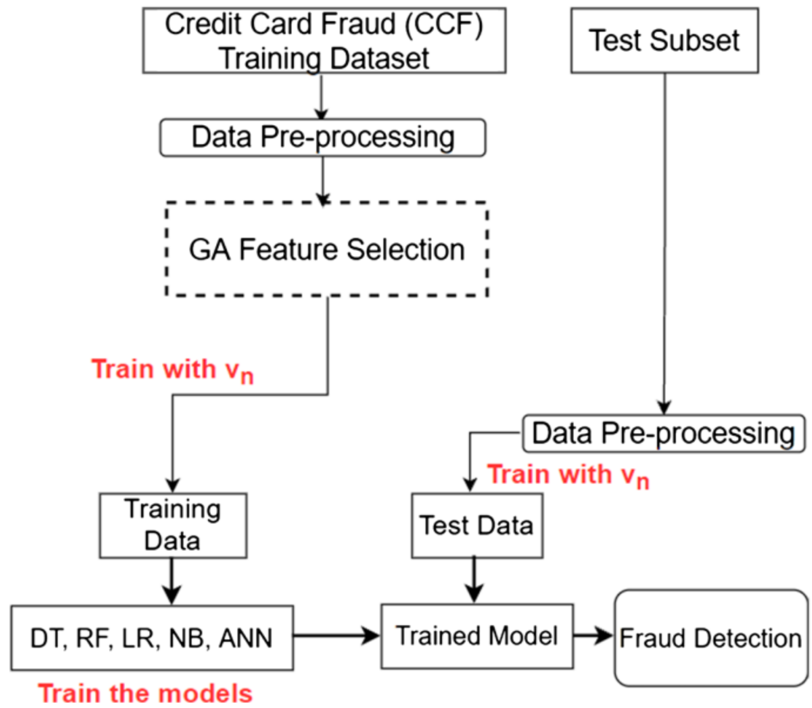

Fig. 3 Architecture of the proposed framework

Experiment on Google colab

Table 8 Classification results a random approach

Model	Accuracy	Recall	Precision	F1-Score
RF	83.78 %	79.64 %	92.78 %	85.71%
DT	89.91 %	79.64 %	68.70 %	73.77%
ANN	88.93 %	78.76 %	82.40 %	80.54%
NB	78.14 %	83.18 %	6.73 %	12.46%
LR	79.91 %	59.29%	81.70 %	68.71 %

Table 9 Comparison with existing methods

Model	Accuracy
LR [13]	97.70 %
DT [13]	95.50 %
SVM [13]	97.50 %
NB [14]	99.23 %
KNN [16]	97.69 %
LR [16]	54.86 %
DT [4]	97.08 %
LR [17]	97.18 %
IF [16]	58.83 %
GA-ANN [17]	81.82 %
GA-DT [17]	81.97 %
GA-RF [17]	77.95 %
GA-RF (Proposed v_5)	99.98 %
GA-DT (Proposed v_1)	99.92 %
GA-LR (Proposed v_1)	99.91 %
GA-NB (Proposed v_5)	99.44 %

To validate the efficiency of our proposed method, we conducted more experiments using a publicly available synthetic dataset that contains the following features: V = **{ User, Card, Year, Month, Day, Time, Amount, Use Chip, Merchant Name, Merchant City, Merchant State, Zip, MCC, Errors, Is Fraud}**.

IBM Synthetic Credit Card Fraud Dataset:
https://ibm.ent.box.com/v/tabformer-data/folder/130747715605

Table 10 GA Selected features—synthetic dataset

Attribute vector	Vector length	Attribute list
GA selected feature space, v_0	7	Card, Year, Month, Day, Amount, Zip, MCC

Table 11 Classification results for v_0 in Table 8

Model	Accuracy	Recall	Precision	F1-Score
RF	99.95 %	99.82 %	99.92 %	99.82 %
DT	100 %	99.71 %	99.51 %	99.61 %
ANN	100 %	72.09 %	84.31 %	77.72 %
NB	99.10 %	96.29 %	84.47 %	41.52 %
LR	99.96 %	99.12 %	80.68 %	88.95 %

In this research, a GA-based feature selection method in conjunction with the RF, DT, ANN, NB, and LR was proposed.

"A Feature Extraction Method for Credit Card Fraud Detection"

Authors: Yu Xie, Guanjun Liu, Ruihao Cao, Zhenchuan Li, Chungang Yan, and Changjun Jiang
Paper: https://ieeexplore.ieee.org/document/8782457

Dataset:
Their dataset comes from a **financial company in China.** It includes 5 million of B2C transactions from November
2016 to June 2017. All transactions contain labels. Label "0" means that a transaction is legal (negative sample) and "1" means fraud (positive sample). The dataset contains **64 original feature**s, including transaction date, transaction amount, transaction location and account number, etc.

The current feature engineering that is based on the frequency of transactions is not perfect.

Propose: A **rule-based feature engineering** that considers both **individual behavior and group behavior,** and portrays individual behavior as group features, and thus can more effectively distinguish legitimate and fraudulent transactions.

"In this paper, they mainly focus on the transaction behavior and the features of preprocessing."

Individual behaviors:

- Hidden Markov chains
- and self-organizational networks

When a new transaction of a user does not match the individual behavior model of the user, the transaction is thought of being conducted by a defrauder.

However, individual behavior methods usually have two flaws:

1. It is difficult for many users to create an accurate individual behavior model since their transaction data are relatively less.
2. An individual behavior model cannot characterize some new transaction behaviors of a user.

Group behaviors:

Classification methods

- Random forest
- Neural networks are often used to train a model.

Flaws:

- Class imbalance problem
- Training time too much

Feature creation :

- Frequency-based feature creation

  Raw Feature
  they lose a lot of important information such as the user's behavior habits. To deal with this, methods in [5][7], [22] create some new features based on some raw features.These new features utilize the time-series of transactions.
  According to the **time and the merchant of the transaction**, the frequency analysis is carried out from the sequential logic, and then the feature of the raw data is expanded. T**hese approaches have some flaws.** Generally, two or three different attributes are combined to construct a new feature.
  If there are time-related attributes, researchers often draw different time windows, get the eigenvalues of the same attribute at different times, or decompose/slice a feature

TABLE I PRIMARY ATTRIBUTES

Attributes name	**Description**
Common_phone	Customer's usual mobile phone number
Pay_bind_phone	Customer's number bound on the electronic payment platform
Pre_trade_result	Customer's verification results of the last transaction
Is_common_ip	Whether this transaction is a common IP
Trade_amount	Amount of a transaction
Pay_single_limit	Limit on the amount of a single transaction
Pay_accumulate_limit	Total daily transaction amount limit
Account_number	Credit card number
Client_mac	MAC address of a transaction
Trade_date	Date of transaction
Trade_time	Exact time of transaction
White_list_mark	Whether the account is in the trusted list
Card_balance	Account balance before payment
Transaction_object	Is the receiver a person or a business
Receiver_number	Receiver number of account's last transaction
Last_trade_time	Account's last transaction time

Derived Feature

TABLE II
DERIVED ATTRIBUTES BASED ON FREQUENCY

Attributes name	Description
Amount_over_month	Average amount spent per transaction over a month
Average daily over month	Average amount spent per day over the past 30 days
Average over 2 months	Average amount spent over the course of 1 week during past 2 months.
Amount Transaction_object over month	Average amount per day over a 30 day period on all transactions up to this one on the same transaction_object as this transaction
Number Transaction_object over month	Total number of transactions with same transaction_object during 30 days
Amount Transaction_object over 2 months	Average amount spent over the course of 1 week during past 2 months on the same transaction_object as this transaction
Amount same day	Total amount spent on the same day up to this transaction
Number same day	Total number of transactions on the same day up to this transaction
Number client_mac over month	Total number of transactions with same client_mac_ during 30 days

The **Trade time** in the data are constructed according to the quarter and cycle. However, this method **cannot reflect the relationship between legitimate transactions and fraud.** To deal with this, we propose a rule-based method to generate features.

- **Rule-based feature creation**

Rule 1: Consistency feature matching rule.
For a lot of fraudulent transactions, fraudsters use a new electronic payment phone number to ensure the safety of his/her common phone number and avoid being detecte.

Therefore, they construct a rule called the consistency feature matching rule.

The **matching function M_atch(a, b)** is constructed to detect whether common **phone and pay bind phone** match in a transaction record. The new feature **phone matching** is assigned by 0 if they match, and otherwise, it is 1.

Example

TABLE III
DERIVED ATTRIBUTES BASED ON RULES

Attributes name	**Description**
Phone_matching	Whether common_phone and pay_bind_phone is match
Uncertain_validation	Bank staff are unable to give timely and accurate judgments about suspected transactions
Sensitive_single_amount	Whether the client's single transaction amount is abnormal
Sensitive_daily_total_amount	Whether the client's total daily transaction amount is abnormal
Sensitive_test_amount	Whether the account has conducted exploratory trading
Large_amount	Whether the account has made transactions of large amount in a short time
Untrusted_frequent_trade	Untrusted accounts are traded frequently
Big_one-time_deal	A large one-time transaction transfers all the balance in the account
Untrusted_time	Accounts are traded at irregular times to untrusted accounts
Untrusted_place	The locations of the two transactions in the account are too wide

How Rule num 1 calculated

$$Match(NUM_{com}, NUM_{e-pay}) = \begin{cases} True & NUM_{com} = NUM_{e-pay} \\ False & otherwise \end{cases} \tag{1}$$

$$phone\_matching = \begin{cases} 1 & Match(NUM_{com}, NUM_{e-pay}) = False \\ 0 & otherwise \end{cases} \tag{2}$$

TABLE IV
EXAMPLE CALCULATION OF RULE 1

PRIMARY			DERIVED
Account Number	*Common Phone*	*Pay Bind Phone*	*Phone Matching*
1	10000	10000	0
2	10000	10101	1

Rule 2: Uncertain validation rule.

The feedback of a bank staff's. They used a system verification V f = {V0 , V1 , N U LL},

- V0 represents the verification result as genuine transaction,
- V1 represents the verification result as fraudulent transaction
- NULLindicates that a suspicious transaction cannot be determined by the bank staff.

These NULL transactions will be considered as legitimate transactions after a period of time, which is called the validation delay [4]. This will lead to a lot of misreporting.Therefore, they combine pre trade result with is common ip for further verification. is common ip
∈ {T rue, F alse}.

TABLE V
EXAMPLE CALCULATION OF RULE 2

PRIMARY			DERIVED
Account Number	*Vf*	*Is_Common_Ip*	*Uncertain_Validation*
1	NULL	FALSE	1
2	NULL	TRUE	0
3	V0	FALSE	0

Rule 3: Sensitive amount rule.
The Amount features which are related to amount are trade amount, pay single limit and
pay accumulate limit.

- **Trade amount** of each transaction is close to the **pay single limi**t;
- The **total trade amount** per day is close to the **pay accumulate limit**;
- A fraudster generally makes a **small trial** deal before making a **large transactio**n in order to avoid being found as a fraudster by a fraud detection system.

$$
sensitive\_single\_amount = \begin{cases} 1 & a_c \in [A_1 - \varepsilon_1, A_1] \\ 0 & otherwise \end{cases} \tag{4}
$$

$$
sensitive\_daily\_amount = \begin{cases} 1 & \sum_{i=1}^{c} a_i \in [A_2 - \varepsilon_2, A_2] \\ 0 & otherwise \end{cases} \tag{5}
$$

$$
sensitive\_test\_amount = \begin{cases} 1 & a_{c-1} \in A_{small} \wedge a_c \in A_{large} \\ 0 & otherwise \end{cases} \tag{6}
$$

A1 = Single transaction limit
A2 = Daily Transaction limit
A_small = Small amount of transaction
A_large = Large amount of transaction

TABLE VI
EXAMPLE CALCULATION OF RULE 3

PRIMARY					DERIVED		
Account Number	*Trade Time*	*Trade Amount*	*Pay Single Limit*	*Pay Accumulate Limit*	*Sensitive Single Amount*	*Sensitive Daily Amount*	*Sensitive Test Amount*
1	01/01 18:00	1000	5000	12000	0	0	0
1	01/01 18:30	1	5000	12000	0	1	1
1	01/01 18:33	4998	5000	12000	1	1	1
1	01/01 18:37	4999	5000	12000	1	1	1
1	01/01 18:40	2000	5000	12000	0	1	1

Rule 4: Frequent large amount transaction rule
When a credit card is stolen in a short period of time lots of large transaction is done by fraudster. New feature create large amount. Function:

$$
large\_amount = \begin{cases} 1 & Gap(a, b) \leq 3min \wedge a_c \in A_{large} \\ 0 & otherwise \end{cases} \tag{7}
$$

Rule 5: Elderly rule : Target old person
According to the **original feature trade amount, card balance**, the rule adds a new feature **big onetime deal** to raw data. Function:

$$
big\_onetime\_deal = \begin{cases} 1 & trade\_amount \rightarrow card\_balance \\ 0 & otherwise \end{cases} \tag{8}
$$

Rule 6: Non-trusted account rule.
New accounts usually have a low level of trust. Some untrusted accounts are traded frequently on the same day, and the possibility of fraud is greatly increased. In the **original feature *white list mar*k** ={V0 , V1 } ,

- V0 indicates that the account is in the trusted list and
- V1 indicates that the account is suspected. Therefore,we add a **new feature *untrusted frequent trade.***

Table IX.

$$untrusted\_frequent\_trade = \begin{cases} 1 & white\_list\_mark = V_1 \wedge Num \geq 4 \\ 0 & otherwise \end{cases} \tag{9}$$

Num: means number of daily transaction.

Rule 7: Non-trusted location rule.
In a short time, 2 transaction are done but location is very far.

According to the original feature **trade time** and **client mac,** the rule adds new feature **untrusted place** to raw data. **D** represents the distance between two client mac.

$$untrusted\_place = \begin{cases} 1 & D \geq 60km \wedge Gap(a, b) \leq 1h \\ 0 & otherwise \end{cases} \tag{10}$$

Rule 8: Non-trusted time rule.
Fraudsters usually do transactions during **nonworking hours**(20:00 to 6:00). According to the original feature **white list mark and trade time**, the rule adds **a new feature *untrusted time*** to raw data.

$$untrusted\_time = \begin{cases} 1 & white\_list\_mark = V_1 \wedge nonworking \\ 0 & otherwise \end{cases} \tag{11}$$

Experiment: Our experiments are conducted to compare three different feature engineering methods:

- The original features of the data,
- Frequency-based feature engineering
- Rule-based feature engineering.

Dataset split: train-test : 75%-25%.
Algorithm used: Random Forest.

	Precision	Recall	F1 Score	AUC
Raw	.61	.71	.66	.84
Raw+F1	.63	.74	.68	.86
Raw +F2	.61	.87	.72	.93
Raw + F1 +F2	.61	.89	.72	.93