

RESEARCH

Open Access



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

Emmanuel Ileberi^{1*}, Yanxia Sun¹ and Zenghui Wang²

*Correspondence:
emmanuelileberi@gmail.com

¹ Department of Electrical & Electronic Engineering Science, University of Johannesburg, Kingsway Ave, 2006 Johannesburg, South Africa
Full list of author information is available at the end of the article

Abstract

The recent advances of e-commerce and e-payment systems have sparked an increase in financial fraud cases such as credit card fraud. It is therefore crucial to implement mechanisms that can detect the credit card fraud. Features of credit card frauds play important role when machine learning is used for credit card fraud detection, and they must be chosen properly. 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 validate the performance, the proposed credit card fraud detection engine is evaluated using a dataset generated from European cardholders. The result demonstrated that our proposed approach outperforms existing systems.

Keywords: Machine learning, Genetic algorithm, Fraud detection, Cybersecurity

Introduction

In the last decade, there has been an exponential growth of the Internet. This has sparked the proliferation and increase in the use of services such as e-commerce, tap and pay systems, online bills payment systems etc. As a consequence, fraudsters have also increased activities to attack transactions that are made using credit cards. There exists a number of mechanisms used to protect credit cards transactions including credit card data encryption and tokenization [1]. Although such methods are effective in most of the cases, they do not fully protect credit card transactions against fraud.

Machine Learning (ML) is a sub-field of Artificial Intelligence (AI) that allows computers to learn from previous experience (data) and to improve on their predictive abilities without explicitly being programmed to do so [2]. In this work we implement Machine Learning (ML) methods for credit card fraud detection. Credit card fraud is defined as a fraudulent transaction (payment) that is made using a credit or debit card by an unauthorised user [3]. According to the Federal Trade Commission (FTC), there were about 1579 data breaches amounting to 179 million data points whereby credit

card fraud activities were the most prevalent [4]. Therefore, it is crucial to implement an effective credit card fraud detection method that is able to protect users from financial loss. One of the key issues with applying ML approaches to the credit card fraud detection problem is that most of the published work are impossible to reproduce. This is because credit card transactions are highly confidential. Therefore, the datasets that are used to develop ML models for credit card fraud detection contain anonymized attributes. Furthermore, credit card fraud detection is a challenging task because of the constantly changing nature and patterns of the fraudulent transactions [5]. Additionally, existing ML models for credit card fraud detection suffer from a low detection accuracy and are not able to solve the highly skewed nature of credit card fraud datasets. Therefore, it is essential to develop ML models that can perform optimally and that can detect credit card fraud with a high accuracy score.

This research focuses on the application of the following supervised ML algorithms for credit card fraud detection: Decision Tree (DT) [7], Random Forest (RF) [8], Artificial Neural Network (ANN) [12], Naive Bayes (NB) [11] and Logistic Regression (LR) [6]. ML systems are trained and tested using large datasets. In this work, a credit card fraud dataset generated from European credit cardholders is utilized. Often-times, these datasets may have many attributes that could have a negative impact on the performance of the classifiers during the training process. 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 [9].

The reminder of this paper is structured as follows. The second section provides an overview of the classifiers that are used in this research. Section III provides a literature review of similar work. Section IV provides the details of the dataset used in this research. Section V outlines the GA algorithm. Section VI. explains the architecture of the proposed system. We conduct the experiments in Section VII. The conclusion is presented in Section VIII.

Classifiers

Logistic regression

The Logistic Regression (LR) classifier, sometimes referred to as the Logit classifier, is a supervised ML method that is generally used for binary classification tasks [6]. LR is a special type of linear regression whereby a linear function is fed to the logit function.

$$y = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \cdots + \alpha_n X_n \quad (1)$$

$$q = \frac{1}{1 + e^{-y}} \quad (2)$$

where the value of q will be between 0 and 1. q is the probability that determines the prediction of a given class. The closer q is to 1, the more accurately it predicts a particular class.

Decision trees and random forest

Decision Tree (DT) is a supervised ML based approach that is utilized to solve regression and classification tasks. A DT contains the following types of nodes: root node, decision node and leaf node. The root node is the starting point of the algorithm. The decision node is a point whereby a choice is made in order to split the tree. A leaf node represents a final decision [7]. The RF method conducts its predictions by using an ensemble of DTs [8]. In the RF, a decision is reached by majority vote. The following is a mathematical definition of the RF [10]:

Given a number of trees k , a RF is defined as, $RF = \{g(X, \theta_k)\}$, where $\{\theta_k\}$ represents independent identically distributed trees that cast a vote on input vector X . The label with the most votes is the prediction.

Naive Bayes

The Naive Bayes (NB) is a supervised ML technique that is based on Bayes' theorem. The NB method assumes the independence of each pair of attributes when provided with the dependant variable (the class). In this research, the Gaussian NB (GNB) classifier was used. With the GNB, we assume that the probability of the attributes is Gaussian as explained in Equation (3).

$$P(x_n|y) = \frac{1}{\sqrt{2\pi\alpha_y^2}} \exp\left(-\frac{(x_n - \beta_y)^2}{2\alpha_y^2}\right) \quad (3)$$

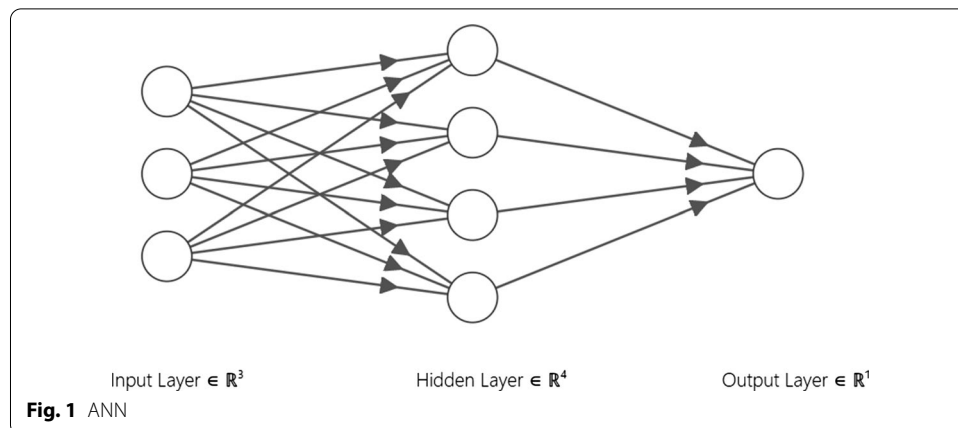
where β_y and α_y are computed using the maximum probability.

Artificial Neural Network

Artificial Neural Network (ANN) is a supervised ML method that is inspired from the inner workings of the human brain. The simplest ANN have the following basic structure: an input layer, one hidden layer and an output layer. The input layer size is based on the number of features in a given dataset. The hidden layer size can be varied based on the complexity of a task and the output layer size depends on the type of problems to be solved. The most basic component of an ANN is a node or neuron. In this research, we consider feed forward ANNs. Therefore, the information flows in one direction (from its input to its output) through a neuron [12]. Figure 1 depicts a graphical representation of a simple ANN with 3 nodes in the input layer, a hidden layer with 4 nodes and an output layer with 1 node.

Related work

In ref. [13], the authors implemented a credit card fraud detection system using several ML algorithms including logistic regression (LR), decision tree (DT), support vector machine (SVM) and random forest (RF). These classifiers were evaluated using a credit card fraud detection dataset generated from European cardholders in 2013. In this dataset, the ratio between non-fraudulent and fraudulent transactions is highly skewed; therefore, this is a highly imbalanced dataset. The researcher used the classification accuracy to assess the performance of each ML approach. The experimental



Better precision
and not the
accuracy as
metric

outcomes showed that the LR, DT, SVM and RF obtained the following accuracy scores: 97.70%, 95.50%, 97.50% and 98.60%, respectively. Although these outcomes are good, the authors suggested that the implementation of advanced pre-processing techniques could have a positive impact on the performance of the classifiers.

Varmedja et al. [14] proposed a credit card fraud detection method using ML. The authors used a credit card fraud dataset sourced from Kaggle [19]. This dataset contains transactions made within 2 days by European credit card holders. To deal with the class imbalance problem present in the dataset, the researcher implemented the Synthetic Minority Oversampling Technique (SMOTE) oversampling technique. The following ML methods were implemented to assess the efficacy of the proposed method: RF, NB, and multilayer perceptron (MLP). The experimental results demonstrated that the RF algorithm performed optimally with a fraud detection accuracy of 99.96%. The NB and the MLP methods obtained accuracy scores of 99.23% and 99.93%, respectively. The authors concede that more research should be conducted to implement a feature selection method that could improve on the accuracy of other ML methods.

Khatri et al. [15] conducted a performance analysis of ML techniques for credit card fraud detection. In this research, the authors considered the following ML approaches: DT, k-Nearest Neighbor (KNN), LR, RF and NB. To assess the performance of each ML method, the authors used a highly imbalanced dataset that was generated from European cardholders. One of the main performance metric that was used in the experiments is the precision which was obtained by each classifier. The experimental outcomes showed that the DT, KNN, LR, and RF obtained precisions of 85.11%, 91.11%, 87.5%, 89.77%, 6.52%, respectively.

Good, they used
precision

Awoyemi et al. [16] presented a comparison analysis of different ML methods on the European cardholders credit card fraud dataset. In this research, the authors used an hybrid sampling technique to deal with the imbalanced nature of the dataset. The following ML were considered: NB, KNN, and LR. The experiments were carried out using a Python based ML framework. The accuracy was the main performance metric that was utilized to assess the effectiveness of each ML approach. The experimental results demonstrated that the NB, LR, and KNN achieved the following accuracies, respectively: 97.92%, 54.86%, and 97.69%. Although the NB and KNN performed relatively well, the authors did not explore the possibility to implement a feature selection method.

Data imbalance usually reflects an unequal distribution of classes within a dataset

In ref. [4] the authors utilized several ML learning based methods to solve the issue of credit card fraud. In this work, the researchers used the European credit cardholder fraud dataset. To deal with the highly imbalanced nature of this dataset, the authors employed the SMOTE sampling technique. The following ML methods were considered: DT, LR, and Isolation Forest (IF). The accuracy was one of the main performance metrics that was considered. The results showed that the DT, LR, and IF obtained the accuracy scores of 97.08%, 97.18%, and 58.83%, respectively.

Manjeevan et al. [17] implemented an intelligent payment card fraud detection system using the GA for feature selection and aggregation. The authors implemented several machine learning algorithms to validate the effectiveness of their proposed method. The results demonstrated that the GA-RF obtained an accuracy of 77.95%, the GA-ANN achieved an accuracy of 81.82%, and the GA-DT attained an accuracy of 81.97%.

Research methodology

Dataset

In this research, we use a dataset that includes credit card transactions that were made by European cardholders for 2 days in September 2013. This dataset contains 284807 transactions in total in which 0.172% of the transactions are fraudulent. The dataset has the following 30 features ($V1, \dots, V28$), *Time* and *Amount*. All the attributes within the dataset are numerical. The last column represents the class (type of transaction) whereby the value of 1 denotes a fraudulent transaction and the value of 0 otherwise. The features $V1$ to $V28$ are not named for data security and integrity reasons [19]. This dataset has been used in ref. [4, 13, 14, 16] and one of the key issues that we discovered is the low detection accuracy score that was obtained by those models because of the highly imbalanced nature of the dataset. In order to solve the issue of class imbalance, we applied the Synthetic Minority Oversampling Technique (SMOTE) method in the Data-Preprocessing phase of the proposed framework in Fig. 5 [18]. The SMOTE method works by picking samples that are close to each other within the feature space, drawing a line between the data points in the feature space and creating a new instance of the minority class at a point along the line.

Feature selection

Feature selection (FS) is a crucial step when implementing machine learning methods. This is partly because the dataset used during the training and testing processes may have a large feature space that may negatively impact the overall performance of the models. The choice of which FS method to use depends on the kind of problem a researcher is trying to solve. The following paragraph provides an overview of instances where using a FS method improved on the performance of ML models.

Kasongo [20] implemented a GA-based FS in order to increase the performance of ML based models applied to the domain of intrusion detection systems. The results demonstrated that the application of GA improved the performance of the RF classifier with an Area Under the Curve (AUC) of 0.98. Mienye [21] et al. implemented a particle swarm optimization (PSO) technique to increase the performance of stacked sparse autoencoder network (SSAE) coupled with the softmax unit for heart disease prediction. The PSO technique was used to improve the feature learning capability

of SSAE by optimally tuning its parameters. The results demonstrated that the PSO-SSAE achieved an accuracy of 97.3% on the Framingham heart disease dataset. Hemavathi et al. [22] implemented an effective FS method in an integrated environment using enhanced principal component analysis (EPCA). The results demonstrated that using the EPCA yields optimal results in supervised and unsupervised environments. Pouramirarsalani et al. [23] implemented a FS method using hybrid FS and GA for fraud detection in an e-banking environment. The experimental results demonstrated that using a FS method on a financial fraud datasets has a positive impact on the overall performance of the models that were used. In ref. [24], the authors implemented the GA-based FS method in conjunction with NB, SVM and RF algorithms for credit card fraud detection. The experimental output demonstrated that the RF yielded a better performance in comparison to the NB and SVM.

Genetic algorithm feature selection

The Genetic Algorithm (GA) is a type of Evolutionary inspired Algorithm (EA) that is often used to solve a number of optimization tasks with a reduced computational overhead. EAs generally possess the following attributes [25, 26]:

- **Population** EAs approaches maintain a sample of possible solutions called *population*.
- **Fitness** A solution within the population is called an *individual*. Each individual is characterized by a gene representation and a fitness measure.
- **Variation** The individual evolves through *mutations* that are inspired from the biological gene evolution.

In this study, the RF approach is used as the fitness method inside the GA. Further, the RF method is employed because it resolves the problem of over-fitting that is generally encountered when using regular Decision Trees (DTs). Moreover, RF performs well with both continuous and categorical attributes and RF are known to perform optimally on datasets that have a class imbalance problem. Additionally, the RF is a rule-based approach; therefore, the normalising of data is not required [27]. The alternative to the RF include tree-based ML algorithms such as Extra-Trees and Extreme Gradient Boosting [28, 29]. The fitness method is defined a function that receives a candidate solution (a feature vector) and determines whether it is fit or not. The measure of fitness is determined by the accuracy that is yielded by a particular attribute vector in the testing process of the RF method within the GA. Algorithm 1 provides more details about the implementation of RF in the GA.

Algorithm 1 denotes the pseudo code implementation of the fitness function that was used in the GA. This algorithm consists of 6 main steps. In step 1, the data (20% of the full Credit Card Fraud dataset) is divided into a training (F_{train} and y_{train}) and testing (F_{test} and y_{test}) subsets. In Step 2, an instance of the RF classifier is instantiated. In Step 3, the RF instance is trained using the training set. In Step 4, the resulting model is then evaluated using the testing data y_{test} . In Step 5, the predictions are stored in y_{pred} . In the last step, the evaluation process is conducted using y_{pred} .

During the evaluation procedure, the accuracy is used as the main performance metric. The most optimal model is one that yields the highest accuracy score.

Algorithm 2 is a pseudo code that represents the computation process of a candidate feature vector. In the initialization phase, the clean Credit Card Fraud dataset is loaded. In the second phase, we define all the variables that will be used in the computation procedure of a candidate feature vector. This includes the following: a list, A , that will store the names of all the features that are present in the Credit Card Fraud dataset; y represents the target variable; B denotes an empty array that will store the most optimal feature names. k represents the total number of iterations required to compute a candidate feature vector. Once the definition phase is completed; in Step 1, we generate the initial population (feature names) and store them in A . In Step 2 and Step 3, Algorithm 2 is computed. The fitness value, q is generated in Step 4. q determines whether a candidate feature vector is optimal or not. If a candidate feature vector is not optimal; we compute the crossover (k -point crossover, where $k = 1$), the mutation, the fitness (from Step 6 to Step 10). This process is conducted iteratively till the algorithm converges. The convergence point is decided once the maximum accuracy has been reached over k iterations.

Algorithm 1 Fitness function computation

Input: F , y ; the input vector and the dependant variable.

Output: Acc ; the Accuracy achieved by the RF classifier

Step 1: Divide F and y in F_{train} , F_{test} , y_{train} , y_{test}

Step 2: Instantiate clf , the RF classifier.

Step 3: Fit clf with F_{train} and y_{train}

Step 4: Evaluate clf using F_{test}

Step 5: Get the predictions y_{pred}

Step 6: Get the Acc with the y_{pred} and y_{train}

Algorithm 2 Compute the candidate feature vectors using the GA

Require: C , the Credit Card Fraud Dataset

Require: A , a list that contains all attributes names.

Require: y , the dependant variable

Require: B , an empty array to save the selected attributes

Require: k , the total number of iterations

BEGIN

Step 1: Compute the initial the population PI with A .

Step 2: Compute the fitness method

Step 3: Compute the fitness using C , A , y and PI

Step 4: Calculate optimal fitness value, q

Step 5: Update the list B

for i in $\text{range}(k)$

Step 6: conduct the crossover

Step 7: compute the mutations

Step 8: calculate the fitness

Step 9: Generate the optimal fitness score, q

Step 10: Update the list B

end for

11. The convergence is achieved \rightarrow (update) B and q

STOP

The main steps of the GA that was adapted to our case study are depicted in Fig. 2. This flowchart represents the compact version of the implementation of the pseudo code in Algorithm 1 and Algorithm 2 [30].

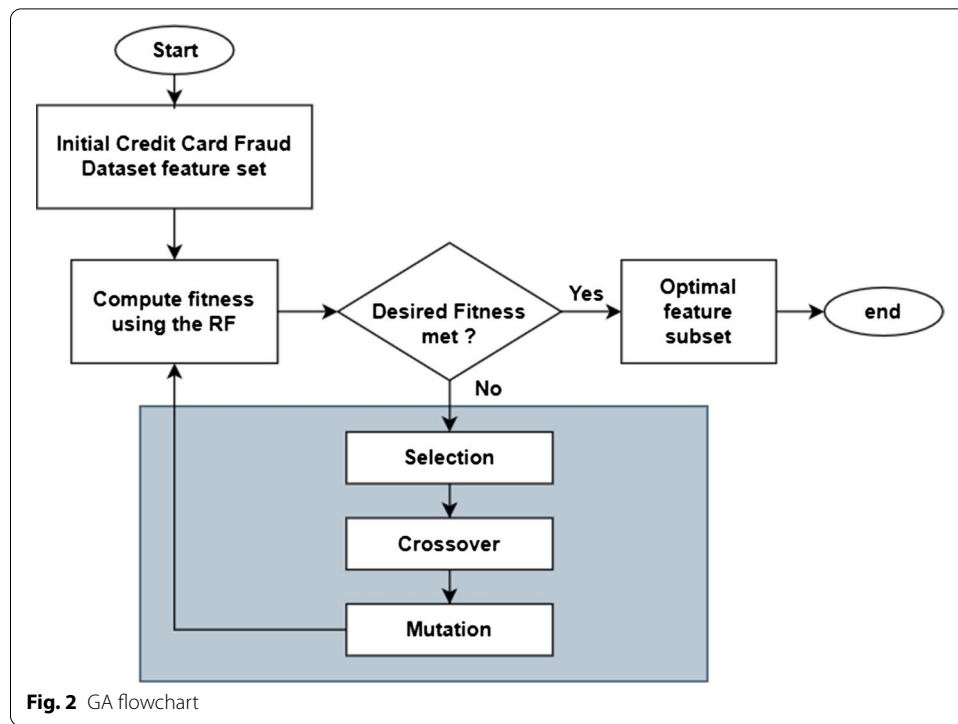


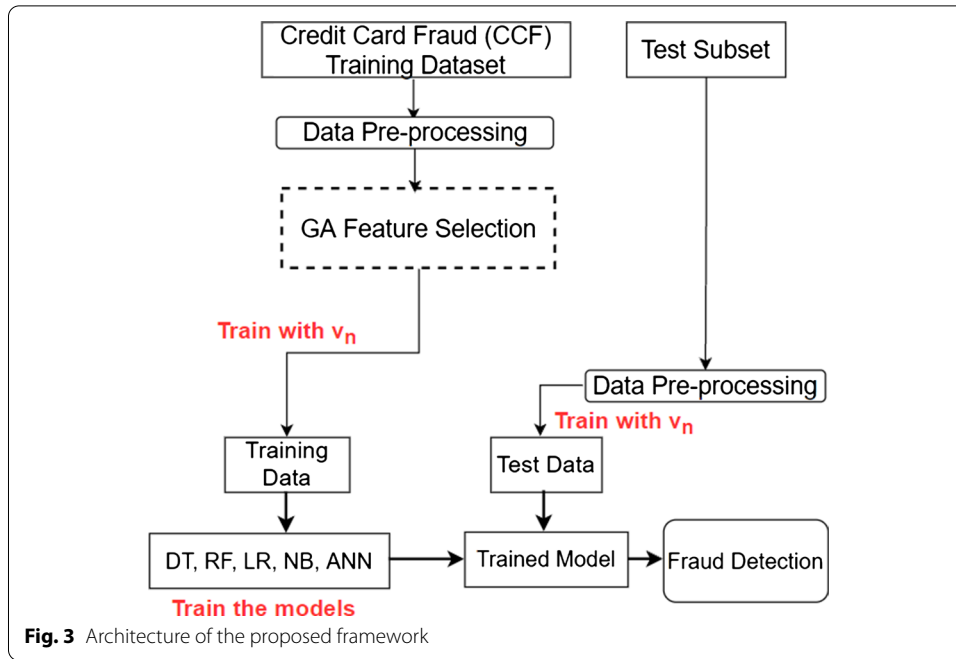
Table 1 GA Selected features

| 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

The architecture of the proposed methodology is depicted in Fig. 3. The initial step is computed in the *Normalize Inputs* block whereby the training dataset is normalized using the min-max scaling method in Equation (4) [31]. The scaling process is done to ensure that all the input values are within a predefined range. The GA algorithm is implemented in the *GA Feature Selection block* using the normalized data from the *Normalize Inputs* block. At each iteration of the *GA Feature Selection block*, the GA generates a candidate attribute vector v_n that is used to train the models in the *Training* block represented by the *Training data* and *Train the models* blocks. The same



vector is also used to test the trained models using the test data. The testing process is conducted using the *Trained Model* block using the *Test Data*. For a given model, the testing process is conducted for each v_n until the desired results are obtained.

$$f_s = \frac{f - \min(f)}{\max(f) - \min(f)} \quad (4)$$

where f is a feature in the dataset.

Performance metrics

The research presented in this paper is modeled as a ML binary classification task. Therefore, we use the accuracy (AC) that was obtained on the test data as the main performance metric. Additionally, for each model, we compute the recall (RC), the precision (PR) and the F1-Score (F-Measure) [32]. To assess the classification quality of each model, we further plot the Area Under the Curve (AUC). The AUC is a metric that reveals how effective a classifier is for a given classification task. The value of the AUC varies between 0 and 1 whereby an efficient classifier would have an AUC value close to 1 [33].

- True positive (TP): attacks/intrusions that are accurately flagged as attacks.
- True Negative (TN): normal traffic patterns/traces that are successfully categorized as normal.
- False positive (FP): legitimate network traces that are incorrectly labeled as intrusive.
- False Negative (FN): attacks/intrusions that are incorrectly classified as non-intrusive.

Table 2 Classification results for v_1

| Model | Accuracy | Recall | Precision | F1-Score |
|-------|----------|---------|-----------|----------|
| RF | 99.94 % | 76.99 % | 89.69 % | 82.85% |
| DT | 99.92 % | 75.22 % | 75.22 % | 75.22% |
| ANN | 99.94 % | 77.87 % | 84.61 % | 81.10% |
| NB | 98.13 % | 84.95 % | 6.83 % | 12.65% |
| LR | 99.91 % | 57.52 % | 82.27 % | 67.70 % |

Table 3 Classification results for v_2

| Model | Accuracy | Recall | Precision | F1-Score |
|-------|----------|---------|-----------|----------|
| RF | 99.93 % | 76.10 % | 82.69 % | 79.26 % |
| DT | 99.87 % | 68.14 % | 60.62 % | 64.16 % |
| ANN | 99.91 % | 66.37 % | 76.53 % | 71.09 % |
| NB | 98.65 % | 77.87 % | 8.59 % | 15.47 % |
| LR | 99.89 % | 47.78 % | 79.41 % | 59.66 % |

$$AC = \frac{TN + TP}{TP + TN + FP + FN} \quad (5)$$

$$RC = \frac{TP}{FN + TP} \quad (6)$$

$$PR = \frac{TP}{FP + TP} \quad (7)$$

$$F1_{score} = 2 \frac{PR \cdot RC}{PR + RC} \quad (8)$$

Experiments

Experimental configuration

The experimental processes were conducted on Google Colab [34]. The compute specifications are as follows: Intel(R) Xeon(R), 2.30GHz, 2 Cores. The ML framework used in this research is the Scikit-Learn [35].

Results and discussions

The experiments were carried out in two folds. In the first step, a classification process was conducted using $F = \{v_1, v_2, v_3, v_4, v_5\}$. For each feature vector in F , the following methods were trained and tested: RF, DT, ANN, NB and LR. The results are depicted in Tables 2, 3, 4, 5, 6. As shown in Table 2, both the ANN and the RF algorithms obtained the highest test accuracy (TAC) of 99.94% using v_1 . However, the RF method obtained the best results in terms of precision. In Table 3, the results that were obtained using v_2 demonstrate that the best model is the RF approach with an accuracy of 99.93%. In Table 4, the RF method also obtained the best fraud detection accuracy of 99.94% using

Table 4 Classification results for v_3

| Model | Accuracy | Recall | Precision | F1-Score |
|-------|----------|---------|-----------|----------|
| RF | 99.94 % | 75.22 % | 85.85 % | 80.18 % |
| DT | 99.90 % | 76.10 % | 68.80 % | 72.26 % |
| ANN | 99.91 % | 67.25 % | 77.55 % | 72.03 % |
| NB | 98.81 % | 81.41 % | 10.07 % | 17.93 % |
| LR | 99.90 % | 53.09 % | 80.00 % | 63.82 % |

Table 5 Classification results for v_4

| Model | Accuracy | Recall | Precision | F1-Score |
|-------|----------|---------|-----------|----------|
| RF | 99.94 % | 77.87 % | 83.80 % | 80.73 % |
| DT | 99.91 % | 76.10 % | 72.26 % | 74.13 % |
| ANN | 99.91 % | 61.06 % | 81.17 % | 69.69 % |
| NB | 98.48 % | 81.41 % | 7.97 % | 14.53 % |
| LR | 99.89 % | 46.90 % | 77.94 % | 58.56 % |

Table 6 Classification results for v_5

| Model | Accuracy | Recall | Precision | F1-Score |
|-------|----------|---------|-----------|----------|
| RF | 99.98 % | 72.56 % | 95.34 % | 82.41 % |
| DT | 99.89 % | 72.56 % | 65.07 % | 68.61 % |
| ANN | 99.08 % | 77.87 % | 12.27 % | 21.20 % |
| NB | 99.44 % | 57.52 % | 15.85 % | 24.85 % |
| LR | 99.77 % | 46.90 % | 34.64 % | 39.84 % |

Table 7 Classification results for full feature vector

| Model | Accuracy | Recall | Precision | F1-Score |
|-------|----------|---------|-----------|----------|
| RF | 87.95 % | 77.87 % | 92.63 % | 84.61 % |
| DT | 96.91 % | 76.10 % | 71.07 % | 73.50 % |
| ANN | 97.80 % | 74.33 % | 42.85 % | 54.36 % |
| NB | 80.31 % | 64.60 % | 13.95 % | 22.95 % |
| LR | 93.88 % | 60.17 % | 62.96 % | 61.53 % |

v_3 . Table 5 presents the results that were achieved by v_4 whereby the DT obtained an accuracy of 99.1% and a precision of 81.17%. Table 6 depicts the outcomes that were obtained when using v_5 . In this case, the RF attained a fraud detection accuracy of 99.98% and precision of 95.34%. In comparison to the results obtained by v_1 , v_2 , v_3 and v_4 ; v_5 obtained the best results. Moreover, looking at the outcomes presented in Tables 2, 3, 4, 5, 6, the NB method under performed in terms of Recall, Precision and F1-Score.

As an initial validation of the proposed method, we ran further experiments using the full feature vector and a feature vector that was generated using a random approach $\text{random_vec} = \{V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{11}, V_{12}, V_{13}, V_{16}, V_{17}, V_{18}, V_{19}, V_{20}, V_{21}, V_{22}, V_{23}, V_{25}, V_{26}, V_{28}, \text{Amount}\}$. The result are listed in Tables 7 and 8. In both

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 % |

instances, we observed serve drop in the performance our the models in comparison to the models that were coupled with the GA (Tables 2, 3, 4, 5, 6).

Furthermore, we computed the AUC of each vector in F . These results are depicted in Figs. 4, 5, 6, 7, 8. In Fig. 4 (v_1), the best performing models in terms of the quality of classification are the RF, NB, and LR with the AUCs of 0.96, 0.97, and 0.97, respectively. In the instance of v_5 (Fig 8), the RF and NB obtained the highest AUCs of 0.95 and 0.96. Moreover, a comparison analysis is presented in Table 7. This comparison reveals that the GA feature selection approach presented in this paper as well as most of the proposed ML methods that were implemented outperformed the existing techniques that are proposed in [4, 13, 14, 16]. For instance, the GA-RF proposed in this research obtained an accuracy that is 2.28% higher than the LR in [13]. The GA-DT proposed in this work yielded a fraud detection accuracy that is 4.42% higher than the DT model presented in [14]. The GA-LR obtained an accuracy that is 2.41% higher than the SVM model presented in [13]. The GA-NB proposed in this research achieved an accuracy that is 1.75% higher than the KNN model proposed in [16]. Additionally, the GA-DT presented in this research achieved an accuracy that is 17.23% greater than the accuracy obtained in [17]. In terms of classification accuracy, the most optimal classifier is the RF (implemented with v_5). This model achieved a noteworthy credit card fraud detection accuracy of 99.98%.

Experiments on synthetic dataset

To validate the efficiency of our proposed method, we conducted more experiments using a publicly available synthetic dataset that contains the following features: $V = \{ \text{User, Card, Year, Month, Day, Time, Amount, Use Chip, Merchant Name, Merchant City, Merchant State, Zip, MCC, Errors, Is Fraud} \}$, where *Is Fraud* denotes the target variable. This dataset contained 24357143 legitimate credit card transactions and 29757 fraudulent ones [36]. In the experiments, we considered the following methods: RF, DT, ANN, NB, and LR. We first processed the dataset through the framework in Fig. 5. The GA module selected the features represented by v_0 in Table 8. These were the features that were used during the training and testing processes of the ML models. Table 9 provides the details of the results that were obtained after the experiments converged. The GA-ANN and the GA-DT achieved accuracies of 100%. These results are backed by AUCs of 0.94 and 1, respectively. The other models that performed remarkably well are the GA-RF and the GA-LR with accuracies of 99.95% and 99.96%. However, the GA-LR yielded a low AUC of 0.63 (Table 10).

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 % |

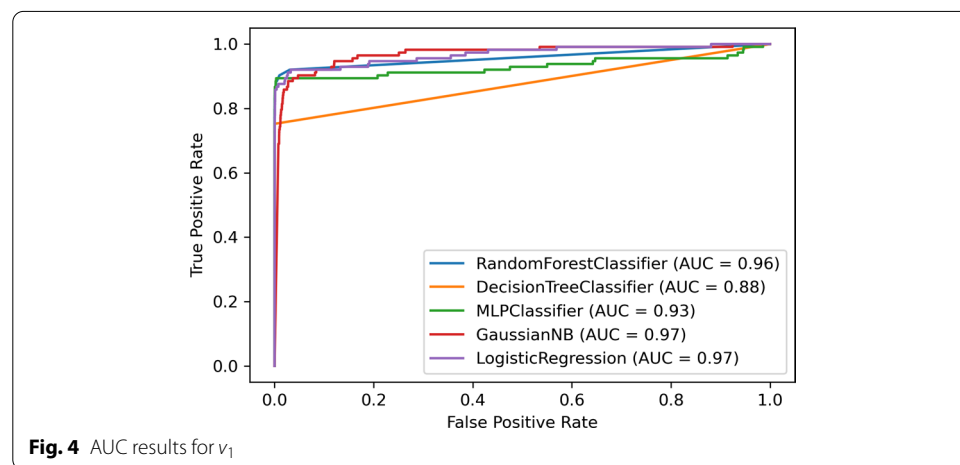


Fig. 4 AUC results for v_1

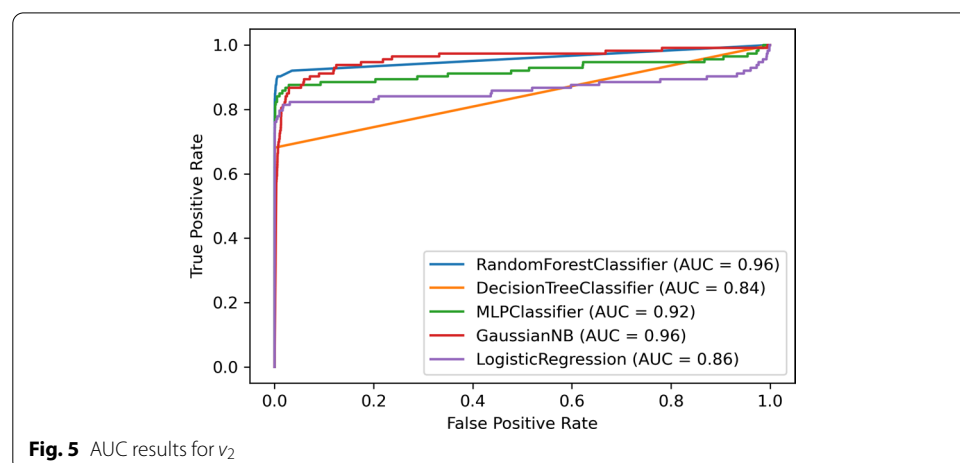
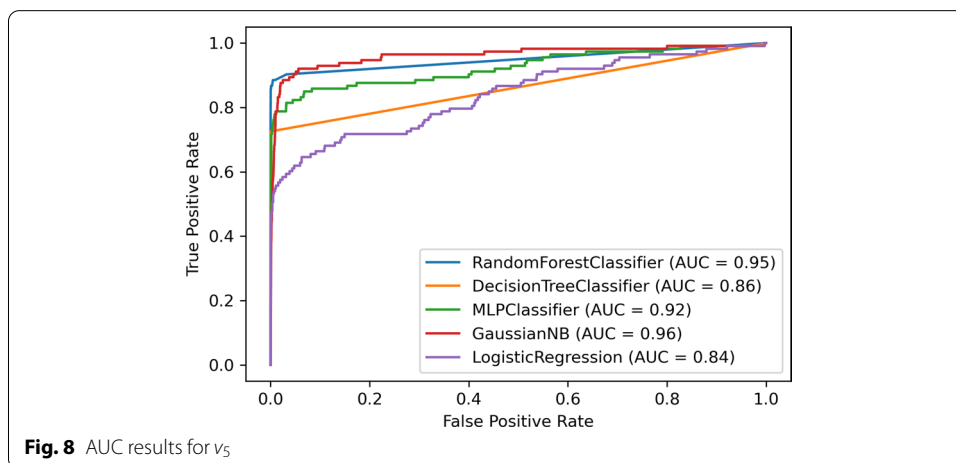
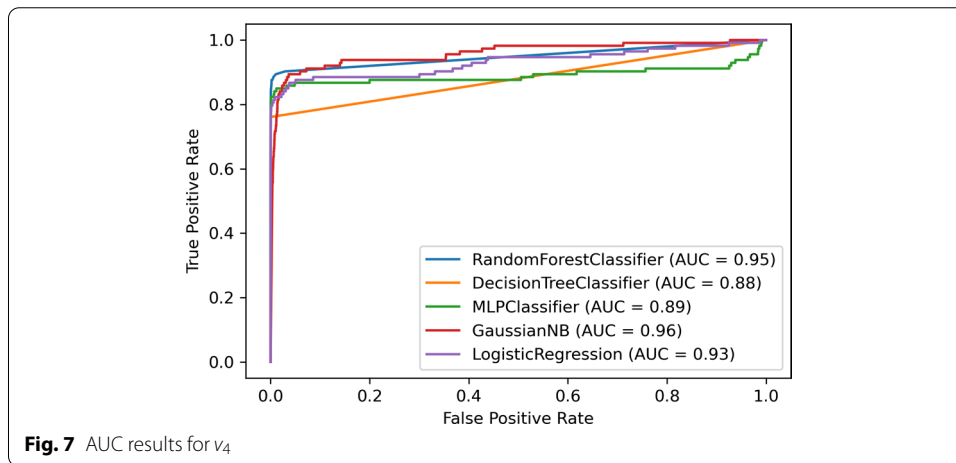
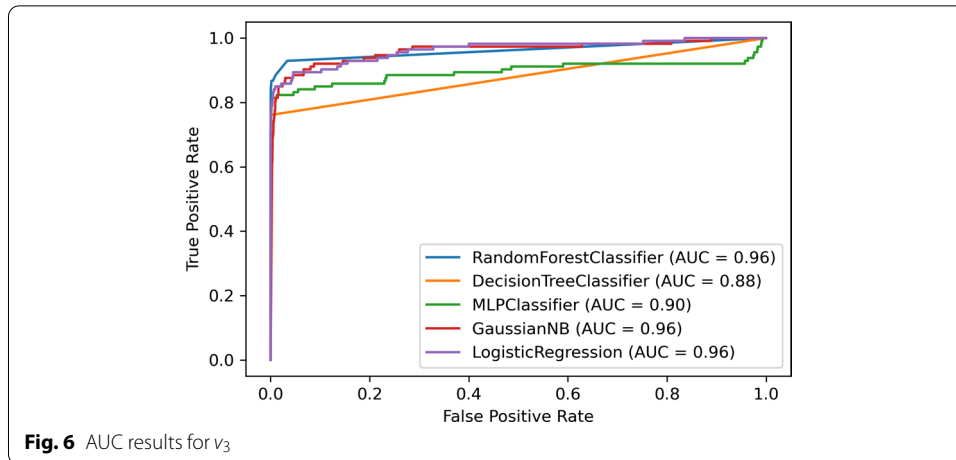


Fig. 5 AUC results for v_2



Moreover, Fig. 7 depicts the ROC curves of the ML models that were considered in the experiments. The result demonstrated that the RF and the DT models achieved an AUC of 1. This indicates that models were perfect at detecting fraudulent activities (Table 11).

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 % |

Conclusion

In this research, a GA based feature selection method in conjunction with the RF, DT, ANN, NB, and LR was proposed. The GA was implemented with the RF in its fitness function. The GA was further applied to the European cardholders credit card transactions dataset and 5 optimal feature vectors were generated. The experimental results that were achieved using the GA selected attributes demonstrated that the GA-RF (using v_5) achieved an overall optimal accuracy of 99.98%. Furthermore, other classifiers such as the GA-DT achieved a remarkable accuracy of 99.92% using v_1 . The results obtained in this research were superior to those achieved by existing methods. Moreover, we implemented our proposed framework on a synthetic credit card fraud dataset to validate the results that were obtained on the European credit card fraud dataset. The experimental outcomes showed that the GA-DT obtained an AUC of 1 and an accuracy of 100%. Seconded by the GA-ANN with an AUC of 0.94 and an accuracy of 100%. In future works, we intend to use more datasets to validate our framework.

Authors' contributions

Ileberi Emmanuel wrote the algorithms and methods related to this research and he interpreted the results. Y. Sun and Z. Wang provided guidance in terms of validating the obtained results. All authors read and approved the final manuscript.

Authors' information

Yanxia Sun got her joint qualification: D-Tech in Electrical Engineering, Tshwane University of Technology, South Africa and PhD in Computer Science, University Paris-EST, France in 2012. Yanxia Sun is currently working as Professor is the Department of Electrical and Electronic Engineering Science, University of Johannesburg, South Africa. She has 15 years teaching and research experience. She has lectured five courses in the universities. She has supervised or co-supervised five postgraduate projects to completion. Currently she is supervising six PhD students and four master students. She published 42 papers including 14 ISI master indexed journal papers. She is the investigator or co-investigator for six research projects. She is the member of the South African Young Academy of Science (SAYAS). Her research interests include Renewable Energy, Evolutionary Optimization, Neural Network, Nonlinear Dynamics and Control Systems. Zenghui Wang, a Professor in Department of Electrical Engineering, University of South Africa.

Funding

This research is funded by the University of Johannesburg, South Africa.

Availability of data and materials

The datasets used during the current study are available a Kaggle, <https://www.kaggle.com/mlg-ulb/creditcardfraud>. Synthetic Credit Card Fraud Dataset, <https://ibm.ent.box.com/v/tabformer-data/folder/130747715605>.

Declarations

Competing interests

The authors declare that they have no competing interests

Author details

¹Department of Electrical & Electronic Engineering Science, University of Johannesburg, Kingsway Ave, 2006 Johannesburg, South Africa. ²Department of Electrical Engineering, University of South Africa, Florida, 1709 Johannesburg, South Africa.

Received: 30 July 2021 Accepted: 6 February 2022

Published online: 25 February 2022

References

- Iwasokun GB, Omomule TG, Akinyede RO. Encryption and tokenization-based system for credit card information security. *Int J Cyber Sec Digital Forensics*. 2018;7(3):283–93.
- Burkov A. The hundred-page machine learning book. 2019;1:3–5.
- Maniraj SP, Saini A, Ahmed S, Sarkar D. Credit card fraud detection using machine learning and data science. *Int J Eng Res* 2019; 8(09).
- Dornadula VN, Geetha S. Credit card fraud detection using machine learning algorithms. *Proc Comput Sci*. 2019;165:631–41.
- Thennakoon, Anuruddha, et al. Real-time credit card fraud detection using machine learning. In: 2019 9th international conference on cloud computing, data science & engineering (Confluence). IEEE; 2019.
- Robles-Velasco A, Cortés P, Muñuzuri J, Onieva L. Prediction of pipe failures in water supply networks using logistic regression and support vector classification. *Reliab Eng Syst Saf*. 2020;196:106754.
- Liang J, Qin Z, Xiao S, Ou L, Lin X. Efficient and secure decision tree classification for cloud-assisted online diagnosis services. *IEEE Trans Dependable Secure Comput*. 2019;18(4):1632–44.
- Ghiasi MM, Zendehboudi S. Application of decision tree-based ensemble learning in the classification of breast cancer. *Comput in Biology and Medicine*. 2021;128:104089.
- Lingjun H, Levine RA, Fan J, Beemer J, Stronach J. Random forest as a predictive analytics alternative to regression in institutional research. *Pract Assess Res Eval*. 2020;23(1):1.
- Breiman L. Random forests. *Mach Learn*. 2001;45(1):5–32.
- Ning B, Junwei W, Feng H. Spam message classification based on the Naive Bayes classification algorithm. *IAENG Int J Comput Sci*. 2019;46(1):46–53.
- Katare D, El-Sharkawy M. Embedded system enabled vehicle collision detection: an ANN classifier. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC); 2019. p. 0284–0289.
- Campus K. Credit card fraud detection using machine learning models and collating machine learning models. *Int J Pure Appl Math*. 2018;118(20):825–38.
- Varmedja D, Karanovic M, Sladojevic S, Arsenovic M, Anderla A. Credit card fraud detection-machine learning methods. In: 18th international symposium INFOTEH-JAHORINA (INFOTEH); 2019. p. 1–5.
- Khatri S, Arora A, Agrawal AP. Supervised machine learning algorithms for credit card fraud detection: a comparison. In: 10th international conference on cloud computing, data science & engineering (Confluence); 2020. p. 680–683.
- Awoyemi JO, Adetunmbi AO, Oluwadare SA. Credit card fraud detection using machine learning techniques: a comparative analysis. In: International conference on computer networks and Information (ICCN); 2017. p. 1–9.
- Seera M, Lim CP, Kumar A, Dhamotharan L, Tan KH. An intelligent payment card fraud detection system. *Ann Oper Res* 2021;1–23.
- Guo S, Liu Y, Chen R, Sun X, Wang X. X. Improved SMOTE algorithm to deal with imbalanced activity classes in smart homes. *Neural Process Lett*. 2019;50(2):1503–26.
- The Credit card fraud [Online]. <https://www.kaggle.com/mlg-ulb/creditcardfraud>
- Kasongo SM. An advanced intrusion detection system for IIoT based on GA and tree based algorithms. *IEEE Access*. 2021;9:113199–212.
- Mienye ID, Sun Y. Improved heart disease prediction using particle swarm optimization based stacked sparse autoencoder. *Electronics*. 2021;10(19):2347.
- Hemavathi D, Srimathi H. Effective feature selection technique in an integrated environment using enhanced principal component analysis. *J Ambient Intell Hum Comput*. 2021;12(3):3679–88.
- Pouramirsalani A, Khalilian M, Nikravanshalmani A. Fraud detection in E-banking by using the hybrid feature selection and evolutionary algorithms. *Int J Comput Sci Netw Secur*. 2017;17(8):271–9.
- Saheed YK, Hambali MA, Arowolo MO, Olasupo YA. Application of GA feature selection on Naive Bayes, random forest and SVM for credit card fraud detection. In: 2020 international conference on decision aid sciences and application (DASA); 2020. p. 1091–1097.
- Davis L. Handbook of genetic algorithms; 1991.
- Li Y, Jia M, Han X, Bai XS. Towards a comprehensive optimization of engine efficiency and emissions by coupling artificial neural network (ANN) with genetic algorithm (GA). *Energy*. 2021;225:120331.
- Khalilia M, Chakraborty S, Popescu M. Predicting disease risks from highly imbalanced data using random forest. *BMC Med Inf Decis Mak*. 2011;11(1):1–13.
- Abhishek L. Optical character recognition using ensemble of SVM, MLP and extra trees classifier. In: International conference for emerging technology (INCET) IEEE; 2020. p. 1–4.
- Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H. Xgboost: extreme gradient boosting. R package version 04-2. 2015;1(4):1–4.

30. Harik GR, Lobo FG, Goldberg DE. The compact genetic algorithm. *IEEE Trans Evol Comput.* 1999;3(4):287–97.
31. Jain A, Nandakumar K, Ross A. Score normalization in multimodal biometric systems. *Pattern Recognit.* 2005;38(12):2270–85.
32. Kasongo SM, Sun Y. A deep long short-term memory based classifier for wireless intrusion detection system. *ICT Express.* 2020;6(2):98–103.
33. Norton M, Uryasev S. Maximization of auc and buffered auc in binary classification. *Math Program.* 2019;174(1):575–612.
34. Google Colab [Online]. Available: <https://colab.research.google.com/>
35. Scikit-learn : machine learning in Python [Online]. <https://scikit-learn.org/stable/>
36. Altman ER. Synthesizing credit card transactions. 2019. arXiv preprint [arXiv:1910.03033](https://arxiv.org/abs/1910.03033)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
